UFMG UNIVERSIDADE FEDERAL
DE MINAS GERAIS

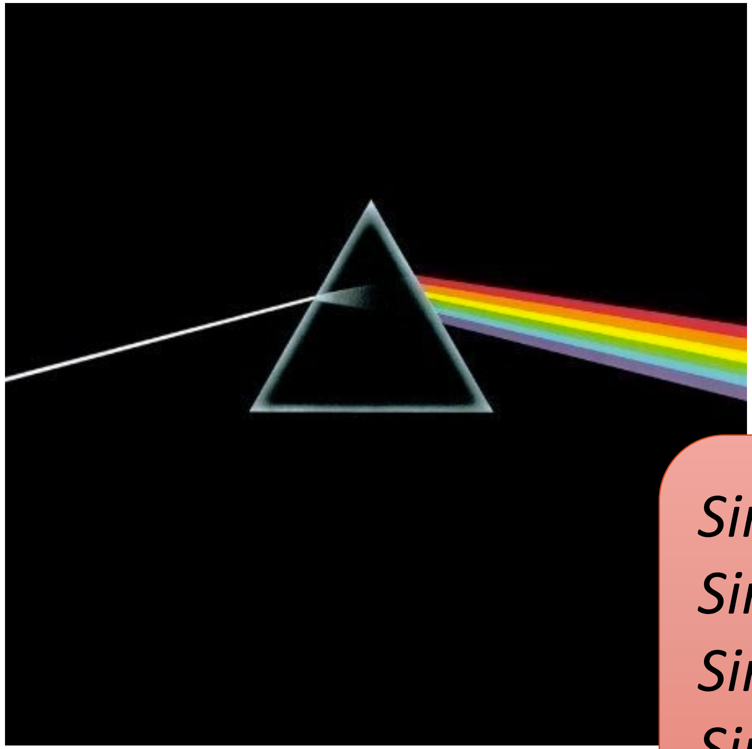
Recommender Systems

Topic Modeling

Rodrygo L. T. Santos
rodrygo@dcc.ufmg.br

Content-based recommendation

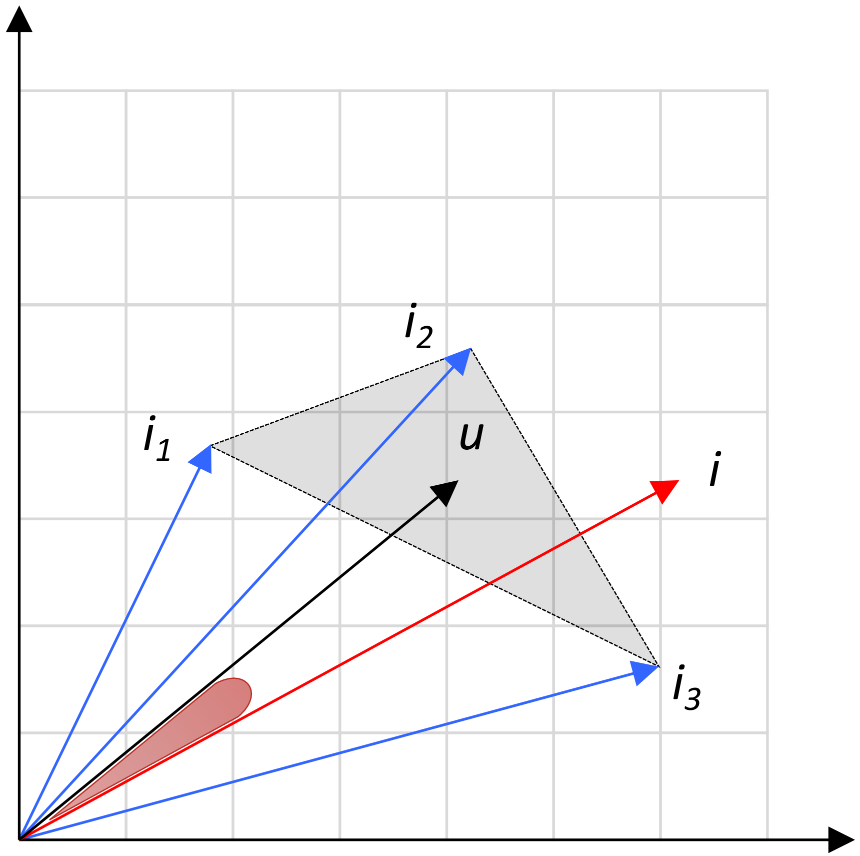
You bought



You may like

Similar artist: Pink Floyd
Similar origin: England
Similar genre: Rock
Similar period: 1970s

Vector space representation



Each item is a vector

- One component for each term in the vocabulary

Each user is a vector

- Some combination of item vectors

Prediction by similarity

- Cosine of the angle between the user and item vectors

The curse of dimensionality

The space of terms is very **high-dimensional**!

Problems

- **Efficiency:** it will take longer to compute similarities
- **Effectiveness:** it will be harder to match similar concepts

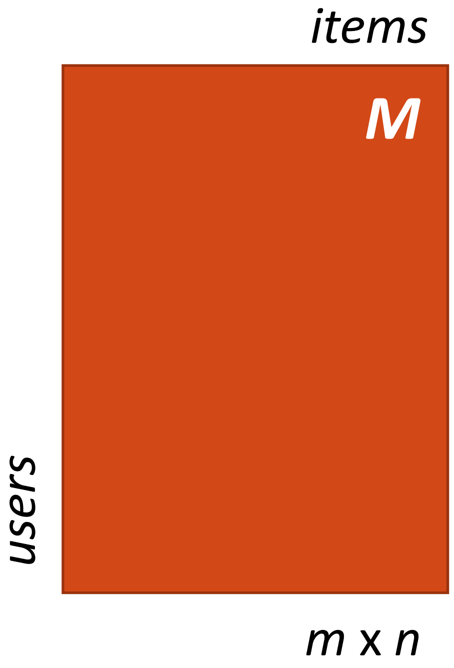
Google Web N-grams

[Franz and Brants, 2006]

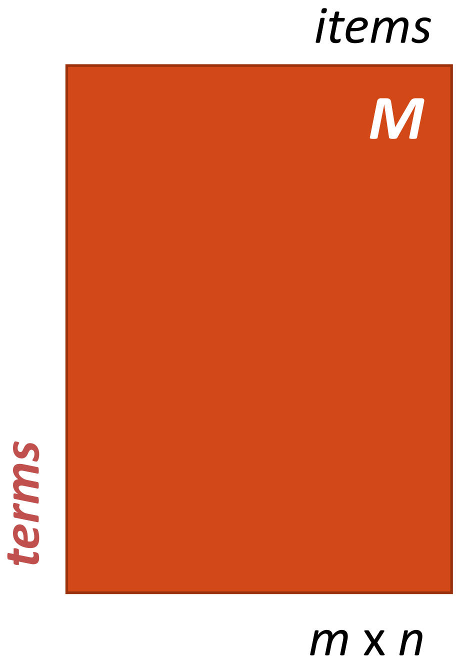
# tokens	1,024,908,267,229
# sentences	95,119,665,584
# 1-grams	**13,588,391**
# 2-grams	314,843,401
# 3-grams	977,069,902
# 4-grams	1,313,818,354
# 5-grams	1,176,470,663

The curse of dimensionality

Collaborative filtering

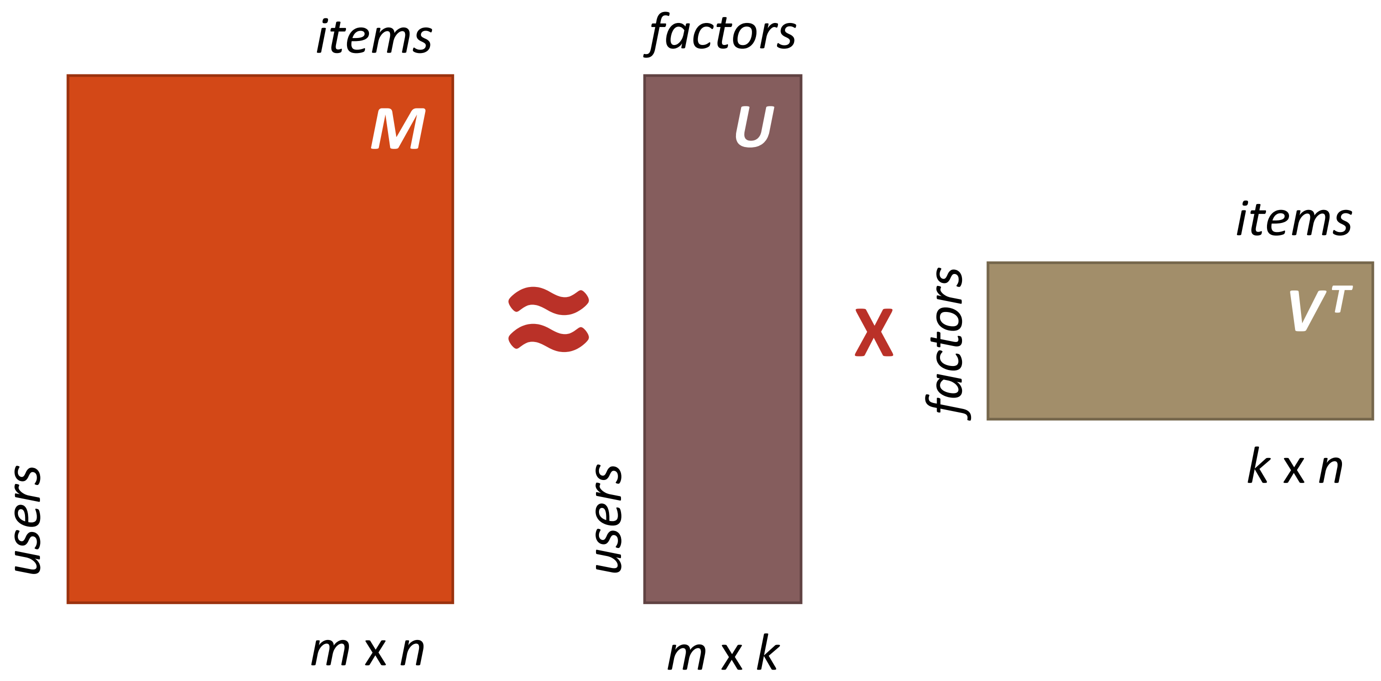


Content-based filtering



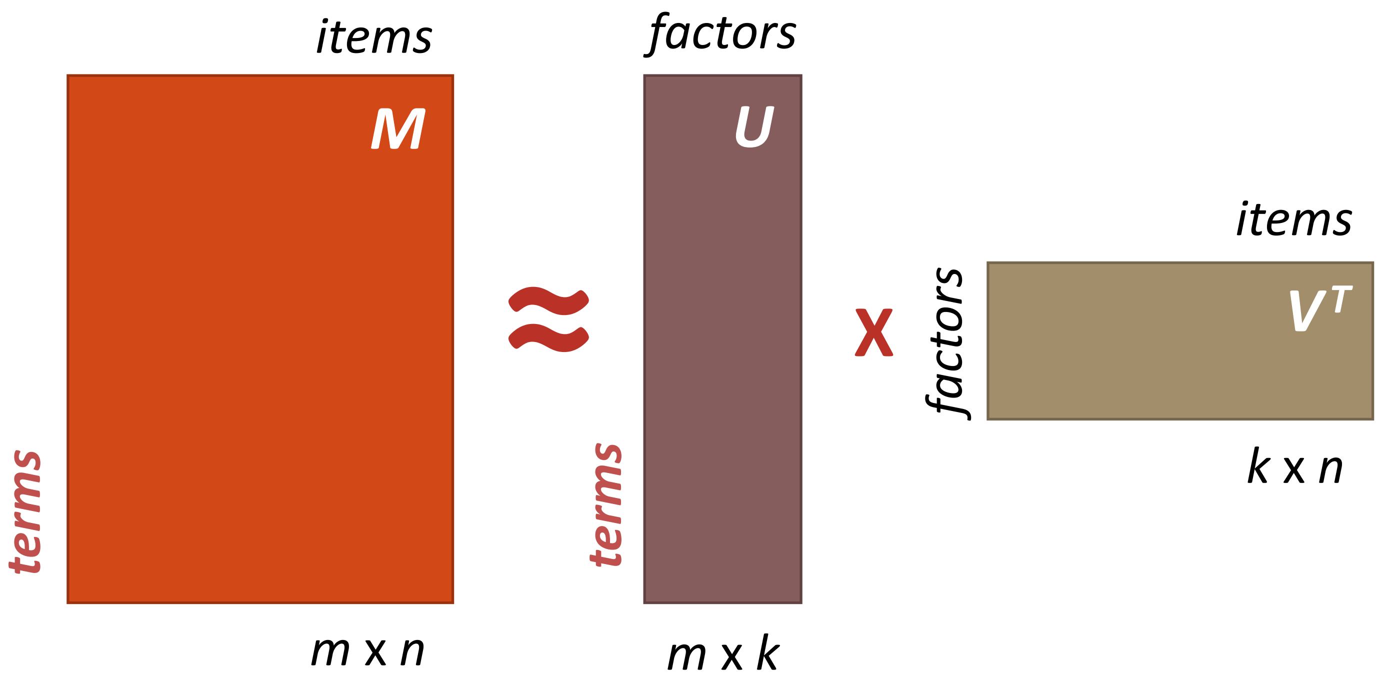
Latent semantic analysis

Collaborative filtering



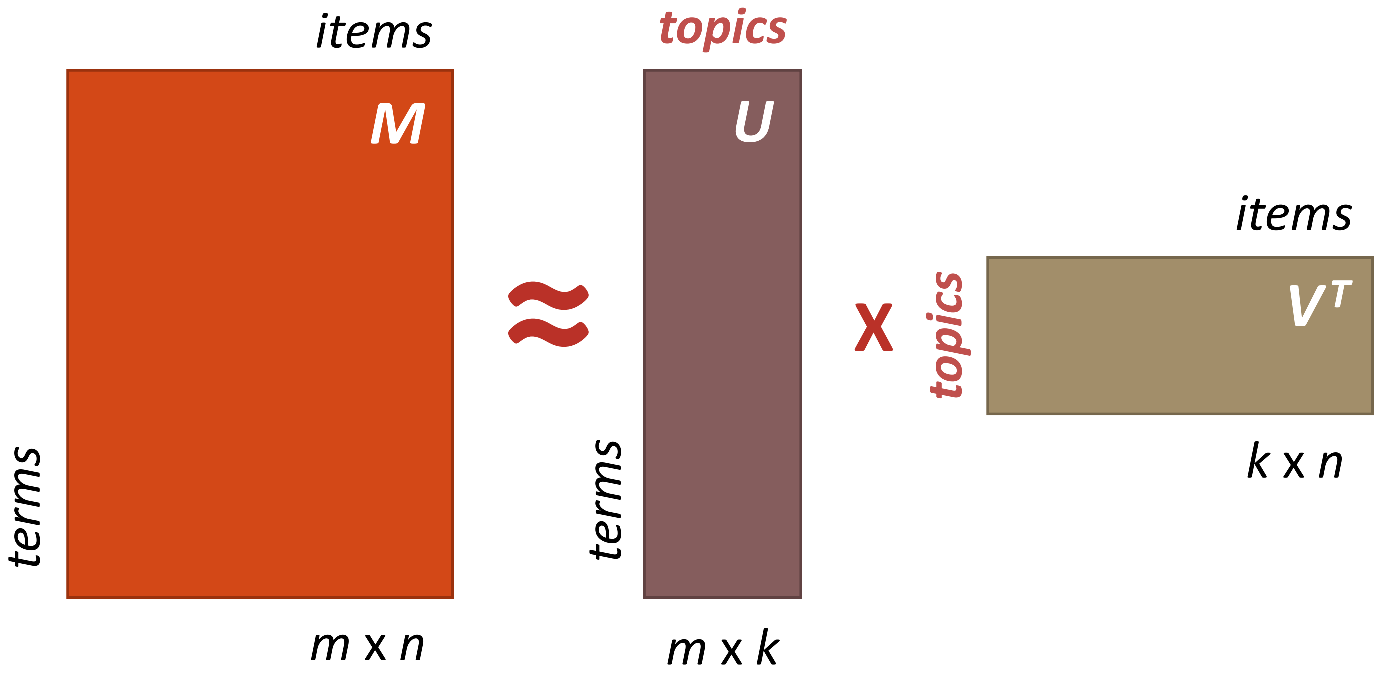
Latent semantic analysis

Content-based filtering

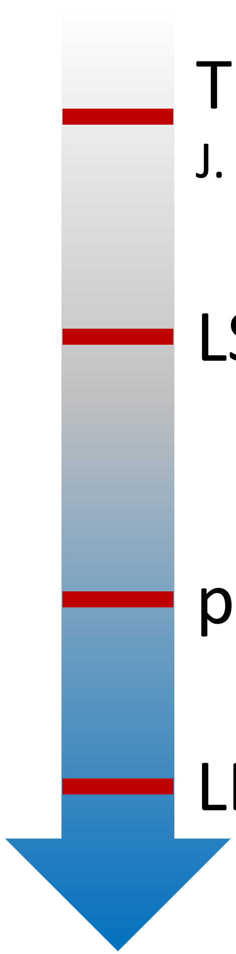


Latent topic modeling

Content-based filtering



Dimensionality reduction



TF-IDF [Luhn, IBM J. R&D 1957; Sparck-Jones, J. Doc. 1972; Salton and Buckley, IP&M 1988]

LSI [Deerwester et al., ASIS 1988]

pLSI [Hofmann, SIGIR 1999]

LDA [Blei et al., JMLR 2003]

Latent Dirichlet allocation (LDA)

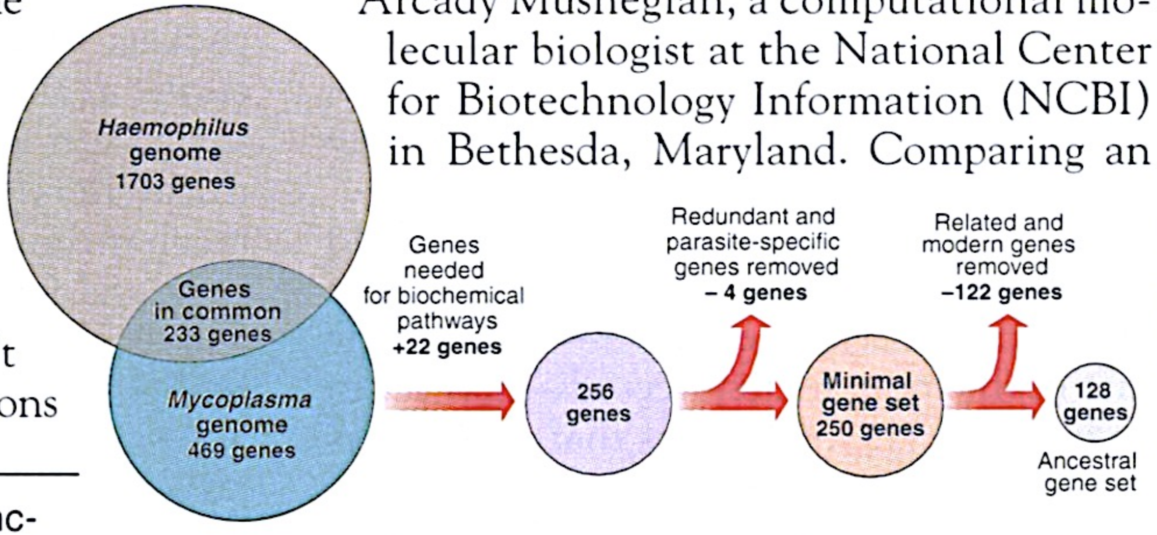
> *“Imagine searching and exploring documents based on the themes that run through them. […] we might first find the theme that we are interested in, and then examine the documents related to that theme.”*

◦ Blei, CACM 2012

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 number. But coming up with a consensus answer may be more than just a genetic numbers game, particularly as more and more genomes are completely 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



ADAPTED FROM NCBI

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

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

Latent topics

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
“genetics”	**“evolution”**	**“disease”**	**“computers”**

Generative modeling

Say you want a document with n words

- Assume there are k known topics
- Choose the document’s distribution over topics
- For each of the n words to be generated
    - Choose a topic from the document’s topic distribution
    - Choose a word from the chosen topic

Generative modeling

Generating n words

- Choose the document’s topic distribution
- For each of the n words to be generated
    - Choose a topic from the document’s distribution
    - Choose a word from the chosen document topic

Generative modeling

Generating n words

- **Choose the document's topic distribution**
- For each of the n words to be generated
  - Choose a topic from the document's distribution
  - Choose a word from the chosen document topic

50%	genetics
30%	evolution
15%	disease
5%	computers

Generative modeling

Generating n words

- Choose the document’s topic distribution
- For each of the n words to be generated
  - **Choose a topic from the document’s distribution**
  - Choose a word from the chosen document topic

50% genetics
30% evolution
15% disease
5% computers

“evolution”
evolution
evolutionary
species
organisms
life
origin

Generative modeling

Generating n words

- Choose the document's topic distribution
- For each of the n words to be generated
  - Choose a topic from the document's distribution
  - **Choose a word from the chosen document topic**

50% genetics
30% evolution
15% disease
5% computers

"evolution"

evolution
evolutionary
species
organisms
life
origin

origin

Generative modeling

Generating n words

- Choose the document's topic distribution
- For each of the n words to be generated
  - **Choose a topic from the document's distribution**
  - Choose a word from the chosen document topic

50% genetics
30% evolution
15% disease
5% computers

"genetics"
human
genome
dna
genetic
genes
sequence

origin

Generative modeling

Generating n words

- Choose the document’s topic distribution
- For each of the n words to be generated
  - Choose a topic from the document’s distribution
  - **Choose a word from the chosen document topic**

50% genetics
30% evolution
15% disease
5% computers

“genetics”

human
genome
dna
genetic
genes
sequence

origin human

Generative modeling

Generating n words

- Choose the document's topic distribution
- For each of the n words to be generated
  - **Choose a topic from the document's distribution**
  - Choose a word from the chosen document topic

50% genetics
30% evolution
15% disease
5% computers

"computers"

computer
models
information
data
computers
system

origin human

Generative modeling

Generating n words

- Choose the document’s topic distribution
- For each of the n words to be generated
  - Choose a topic from the document’s distribution
  - **Choose a word from the chosen document topic**

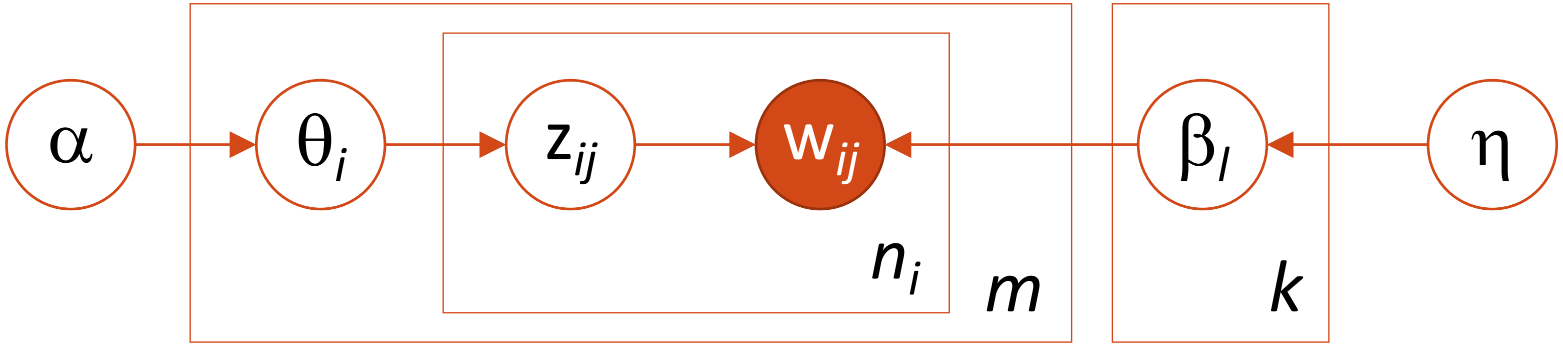
50% genetics
30% evolution
15% disease
5% computers

“computers”

computer
models
information
data
computers
system

origin human models

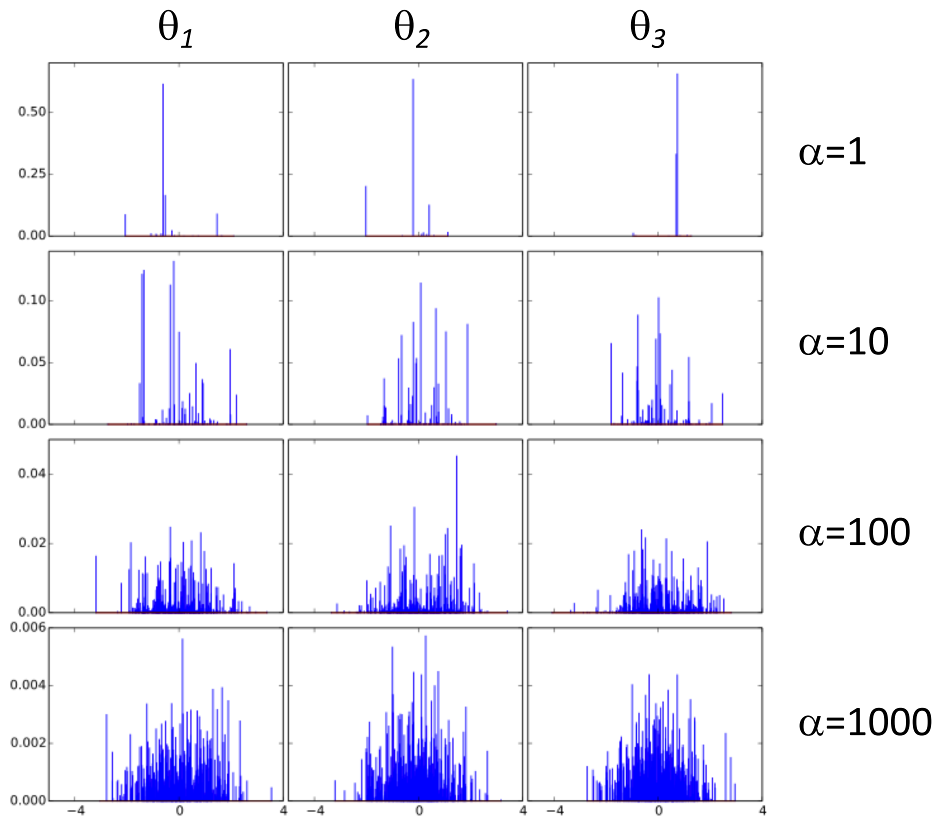
In plate notation



- θ_i: topic distribution of document *i* (of *m* documents)
  - α: parameter of the Dirichlet prior
- β_l: word distribution of topic *l* (of *k* topics)
  - η: parameter of the Dirichlet prior
- w_{ij}: *j*-th word in document *i* (with n_i words)
- z_{ij}: chosen topic of word w_{ij}

Dirichlet distribution

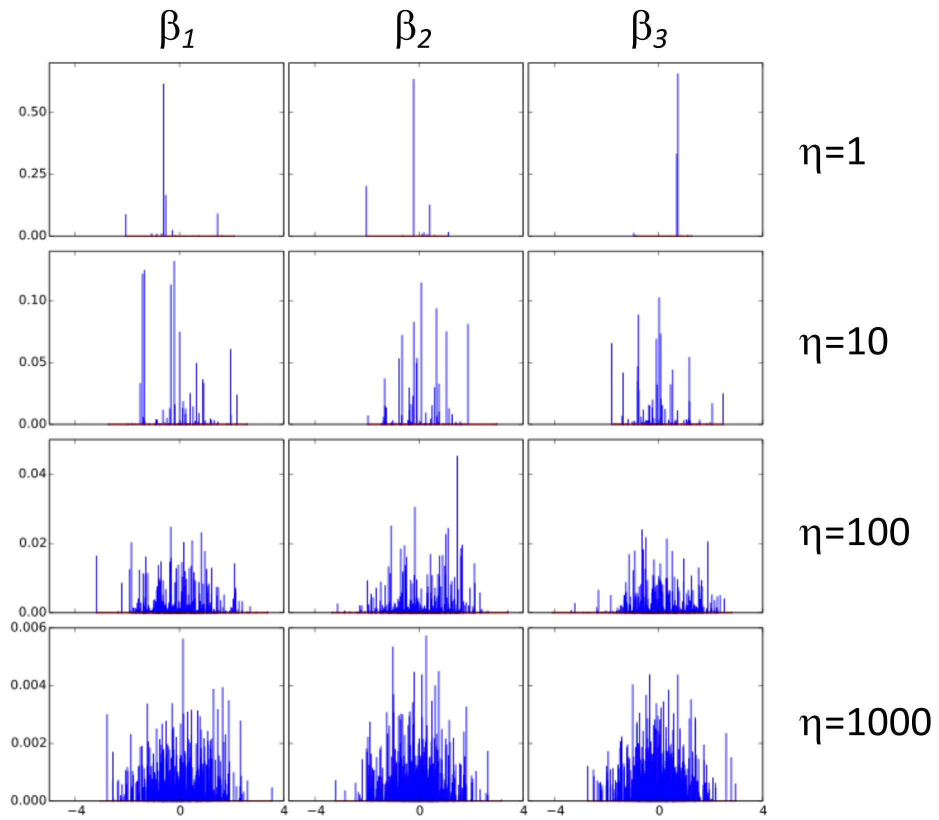
A ”distribution of distributions” with concentration parameter α



Document models as topic distros

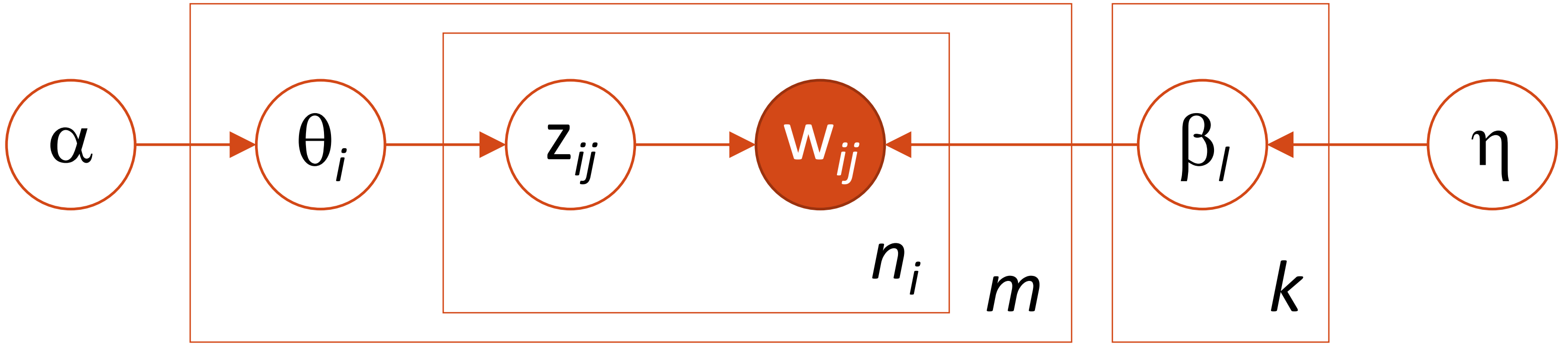
Dirichlet distribution

A ”distribution of distributions” with concentration parameter β



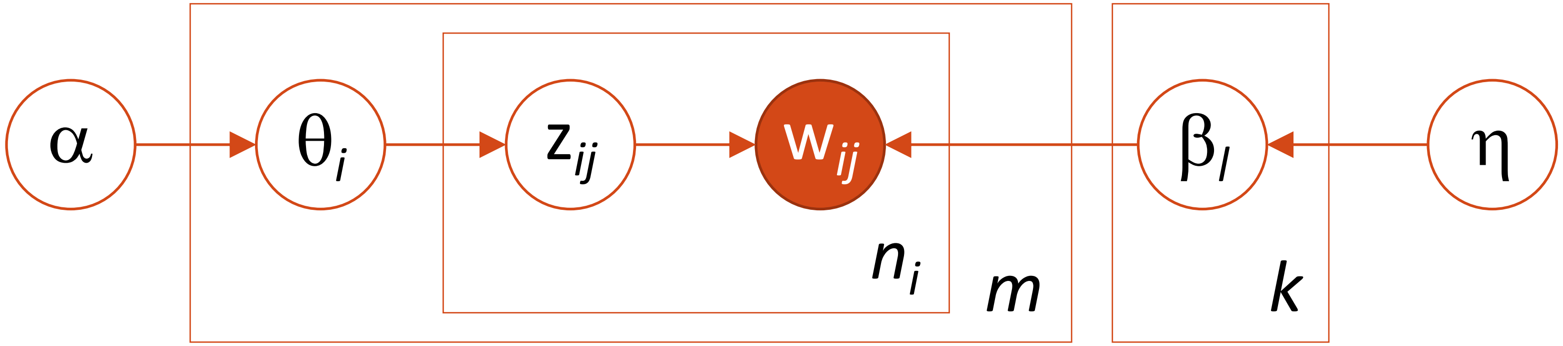
Topic models as word distros

In plate notation



- Choose $\theta_i \sim \mathrm{Dir}(\alpha)$ for $i \in \{1, \ldots, m\}$
- Choose $\beta_l \sim \mathrm{Dir}(\eta)$ for $l \in \{1, \ldots, k\}$
- For each document $i \in \{1, \ldots, m\}$
  - For each position $j \in \{1, \ldots, n_i\}$
    - Choose a topic $z_{ij} \sim \mathrm{Mult}(\theta_i)$
    - Choose a word $w_{ij} \sim \mathrm{Mult}(\beta_{z_{ij}})$

Mathematically

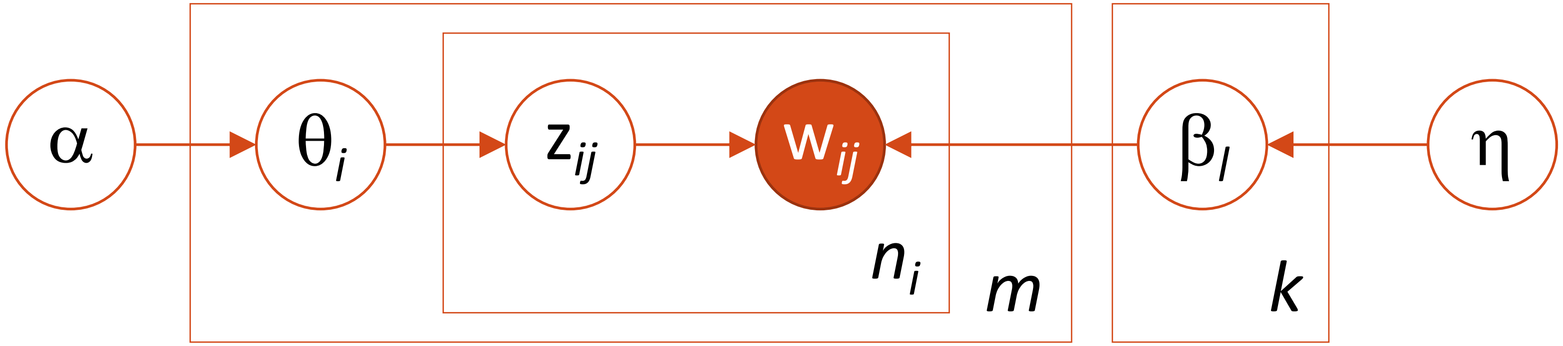


Equivalent to the following joint distribution

$$p(\beta_{1:k},\theta_{1:m},z_{1:m},w_{1:m})$$

$$=\prod_{l=1}^{k} p(\beta_l \mid \eta)\prod_{i=1}^{m} p(\theta_i \mid \alpha)\left(\prod_{j=1}^{n_i} p(z_{ij} \mid \theta_i)\,p(w_{ij} \mid \beta_{1:k},z_{ij})\right)$$

Reversing the logic



In reality, we don't know the topics

- Or, equivalently, the θ_i and β_l distributions

We actually know the documents

- ***How to uncover the hidden topic structure?***

Posterior inference

How to compute the distribution of the topic structure given the observed documents (aka the posterior)?

$$p(\beta_{1:k}, \theta_{1:m}, z_{1:m} \mid w_{1:m}) = \frac{p(\beta_{1:k}, \theta_{1:m}, z_{1:m}, w_{1:m})}{p(w_{1:m})}$$

Problem: computing the marginal $p(w_{1:m})$

- ***Intractable:*** would require examining every possible instantiation of the hidden variables

Gibbs sampling

Consider a document-by-term matrix (tf entries)

	w_1	w_2	w_3	w_4	w_5	w_6	w_7	w_8
d_1		**2**		**5**	**4**			
d_2			**7**		**1**		**3**	
d_3	**1**	**3**		**2**				**1**
d_4						**2**	**4**	
d_5		**5**		**6**	**8**	**2**		

Gibbs sampling

Randomly assign topics in $\{1, 2, \ldots, k\}$, say $k = 3$ topics

	w_1	w_2	w_3	w_4	w_5	w_6	w_7	w_8
d_1		**2**		**5**	**4**			
d_2			**7**		**1**		**3**	
d_3	**1**	**3**		**2**				**1**
d_4						**2**	**4**	
d_5		**5**		**6**	**8**	**2**		

Gibbs sampling

Randomly assign topics in $\{1, 2, \ldots, k\}$, say $k = 3$ topics

	w_1	w_2	w_3	w_4	w_5	w_6	w_7	w_8
d_1		2:2		5:1	4:3			
d_2			7:1		1:3		3:2	
d_3	1:3	3:2		2:2				1:3
d_4						2:3	4:1	
d_5		5:1		6:3	8:2	2:1		

Gibbs sampling

Update topic counts

	w_1	w_2	w_3	w_4	w_5	w_6	w_7	w_8
d_1		2:2		5:1	4:3			
d_2			7:1		1:3		3:2	
d_3	1:3	3:2		2:2				1:3
d_4						2:3	4:1	
d_5		5:1		6:3	8:2	2:1		

	z_1	z_2	z_3
w_1	0	0	1
w_2	5	5	0
w_3	7	0	0
w_4	5	2	6
…			

Gibbs sampling

Update topic assignments one word at a time

	w_1	w_2	w_3	w_4	w_5	w_6	w_7	w_8
d_1		2:2		5:1	4:3			
d_2			7:1		1:3		3:2	
d_3	1:3	3:2		2:2				1:3
d_4						2:3	4:1	
d_5		5:1		6:3	8:2	2:1		

	z_1	z_2	z_3
w_1	0	0	1
w_2	5	5	0
w_3	7	0	0
w_4	5	2	6
…			

Gibbs sampling

Update topic assignments one word at a time

	w_1	w_2	w_3	w_4	w_5	w_6	w_7	w_8
d_1		2:?		5:1	4:3			
d_2			7:1		1:3		3:2	
d_3	1:3	3:2		2:2				1:3
d_4						2:3	4:1	
d_5		5:1		6:3	8:2	2:1		

	z_1	z_2	z_3
w_1	0	0	1
w_2	5	3	0
w_3	7	0	0
w_4	5	2	6
…			

Gibbs sampling

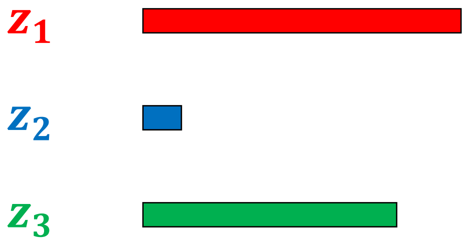
Update topic assignments one word at a time

	w_1	w_2	w_3	w_4	w_5	w_6	w_7	w_8
d_1		2:?		5:1	4:3			

	z_1	z_2	z_3
w_2	5	3	0

How does d_1 like each topic z_i?

$$\frac{n_{d_1,z_i} + \alpha}{\sum_{j=1}^{k} n_{d_1,z_j} + \alpha}$$



Gibbs sampling

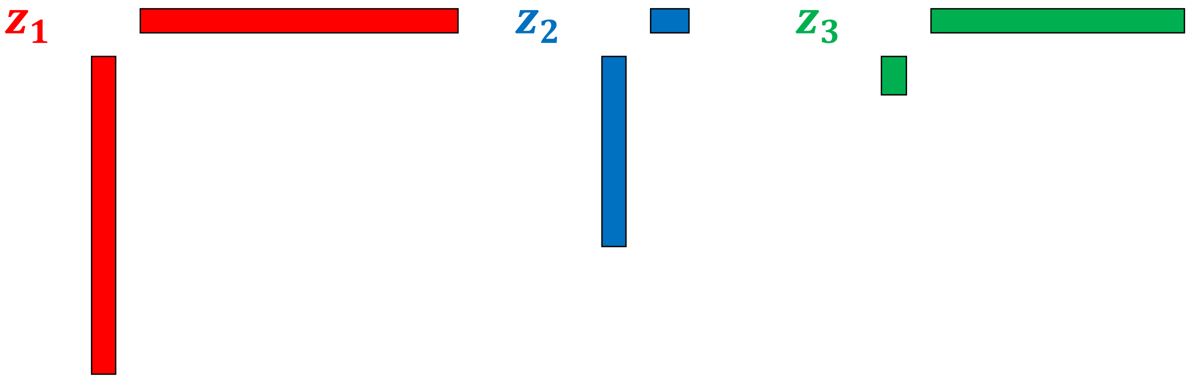
Update topic assignments one word at a time

	w_1	w_2	w_3	w_4	w_5	w_6	w_7	w_8
d_1		2:?		5:1	4:3			

	z_1	z_2	z_3
w_2	5	3	0

How does d_1 like each topic z_i?

How does each topic z_i like w_2?



$$\frac{n_{z_i,w_2} + \eta}{\sum_{j=1}^{k} n_{z_j,w_2} + \eta}$$

Gibbs sampling

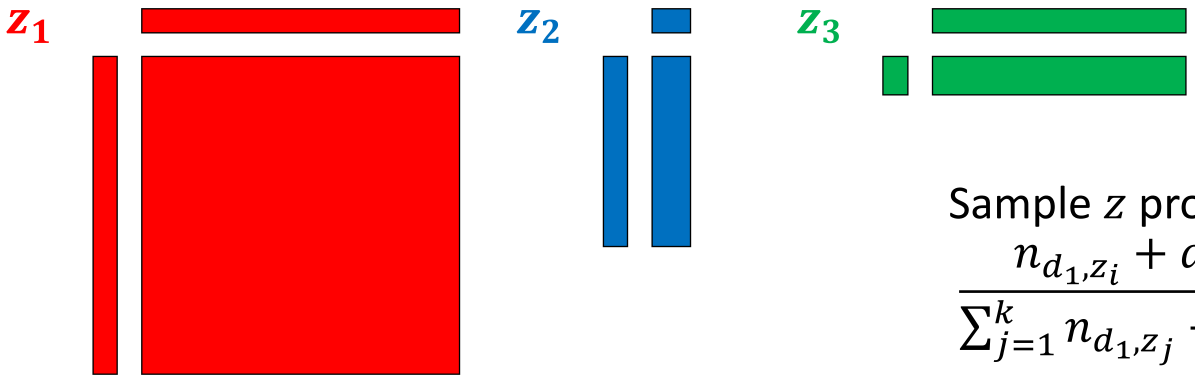
Update topic assignments one word at a time

	w_1	w_2	w_3	w_4	w_5	w_6	w_7	w_8
d_1		2:?		5:1	4:3			

	z_1	z_2	z_3
w_2	5	3	0

How does d_1 like each topic z_i?

How does each topic z_i like w_2?



Sample z proportionally to:

$$\frac{n_{d_1,z_i} + \alpha}{\sum_{j=1}^{k} n_{d_1,z_j} + \alpha} \frac{n_{z_i,w_2} + \eta}{\sum_{j=1}^{k} n_{z_j,w_2} + \eta}$$

Gibbs sampling

Update topic assignments one word at a time

	w_1	w_2	w_3	w_4	w_5	w_6	w_7	w_8
d_1		2:1		5:1	4:3			

	z_1	z_2	z_3
w_2	7	3	0

Gibbs sampling

Update topic assignments one word at a time

	w_1	w_2	w_3	w_4	w_5	w_6	w_7	w_8
d_1		2:1		5:?	4:3			

	z_1	z_2	z_3
w_2	2	3	0

More on this?

Related courses

- Probabilistic graphical models
- Bayesian inference

How to leverage topics?

Vector space model

- $p(i|u) = \cos(\theta_u, \theta_i)$

Item likelihood model

- $p(i|u) = \prod_{w \in i} p(w|\theta_u)^{\mathrm{tf}_{wu}}$

Unified likelihood model

- $p(i|u) = -KL(\theta_u || \theta_i) = -p(w|\theta_u) \log \frac{p(w|\theta_u)}{p(w|\theta_i)}$

LDA variants

Syntactic topic model

- A word or its topic is influenced by syntax

Correlated topic model, hierarchical topic model

- Some topics resemble other topics

Polylingual topic model

- Different languages, same topic mixtures

Relational topic model

- Exploiting link structure

Summary

Content-based recommendation effective

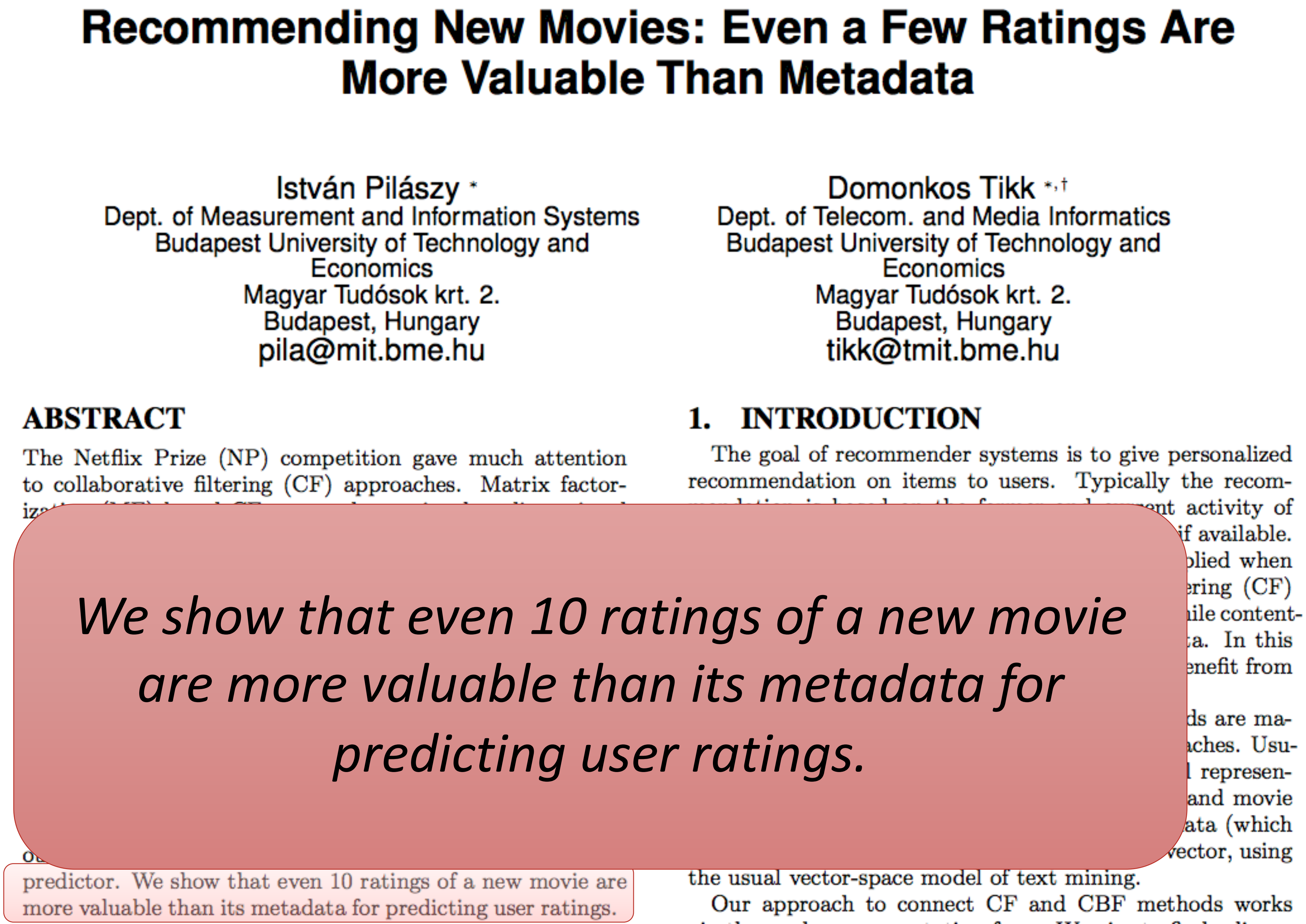
◦ Cold-start items, basket analysis

Build upon a history of research in IR

◦ How to represent and match users and items

Still an active research area

◦ How to go beyond a raw content representation?



Recommending New Movies: Even a Few Ratings Are More Valuable Than Metadata

István Pilászy [*]
Dept. of Measurement and Information Systems
Budapest University of Technology and Economics
Magyar Tudósok krt. 2.
Budapest, Hungary
pila@mit.bme.hu

Domonkos Tikk [*,†]
Dept. of Telecom. and Media Informatics
Budapest University of Technology and Economics
Magyar Tudósok krt. 2.
Budapest, Hungary
tikk@tmit.bme.hu

ABSTRACT

The Netflix Prize (NP) competition gave much attention to collaborative filtering (CF) approaches. Matrix factor-
iz

o

predictor. We show that even 10 ratings of a new movie are more valuable than its metadata for predicting user ratings.

1. INTRODUCTION

The goal of recommender systems is to give personalized recommendation on items to users. Typically the recom-
ent activity of
if available.
plied when
ering (CF)
ile content-
a. In this
enefit from

ds are ma-
aches. Usu-
l represen-
and movie
ata (which
vector, using
the usual vector-space model of text mining.

Our approach to connect CF and CBF methods works

We show that even 10 ratings of a new movie are more valuable than its metadata for predicting user ratings.

References

Probabilistic topic models (CACM 2012)
by David M. Blei