

A Practical Algorithm for Topic Modling with Provable Guarantees

Sanjeev Arora

Rong Ge

Yoni Halpern

David Mimno

Ankur Moitra

David Sontag

Yihcen Wu

Michael Zhu

Presented by: Vanush Vaswani and Kristy Hughes

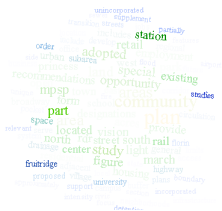
- 1 Introduction
- 2 Topic Modelling
- 3 Algorithm
- 4 Efficiently Finding Anchor Words
- 5 Topic Recovery via Bayes' Rule
- 6 Experimental Results
- 7 Conclusion

Information Overload



Effective Organisation



[illegible]

- ① Introduction
- ② **Topic Modelling**
- ③ Algorithm
- ④ Efficiently Finding Anchor Words
- ⑤ Topic Recovery via Bayes' Rule
- ⑥ Experimental Results
- ⑦ Conclusion

Topics

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
...

Topics are distributions over words

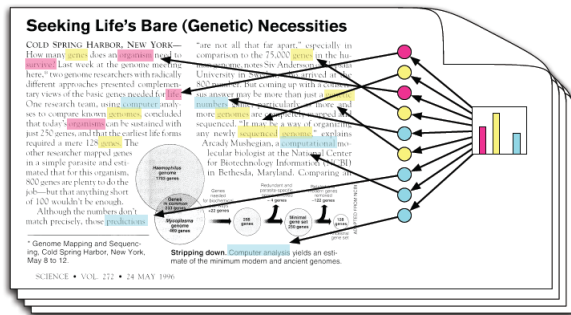


Documents

Documents have
distribution of
topics

Documents

Topic proportions and
assignments



Topic Modelling

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 Anderson, a geneticist at the University of Sydney, 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 sequenced and analyzed. "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.

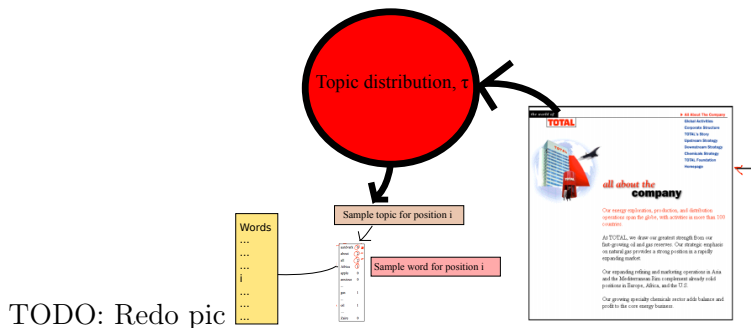
SCIENCE • VOL. 272 • 24 MAY 1996

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

Topic proportions and assignments

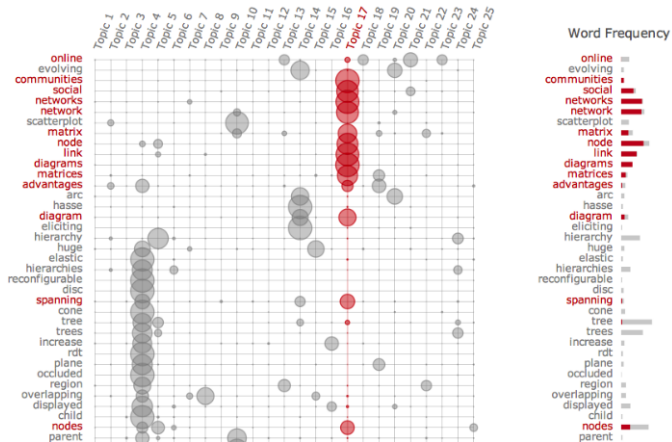
Task

- Assume documents are generated by probabilistic model with unknown variables
- Infer hidden structure onto document
- Situate new document into model



Word-topic Matrix

Extracted: Word-topic matrix



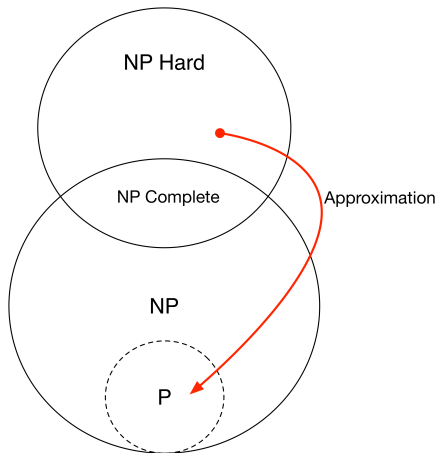
Aim: Find document-topic matrix

Anchor Words

- Word-topic distributions are separable
- There is a word unique to each topic
- Indicates document is partially about that topic
- Can learn parameters in polynomial time provided there is a large enough number of documents

Approximate Inference & Provable Guarantees

- Document-topic inference:
 - NP-hard
- Approximate techniques
- Provably polynomial-time?



- 1 Introduction
- 2 Topic Modelling
- 3 Algorithm**
- 4 Efficiently Finding Anchor Words
- 5 Topic Recovery via Bayes' Rule
- 6 Experimental Results
- 7 Conclusion

Algorithm

Input: Corpus \mathcal{D} , Number of topics K

Output: Word-topic matrix A , topic-topic matrix R

- 1 Compute word-word co-occurrence matrix
- 2 Normalize the matrix
- 3 Find anchor words
- 4 Recover topics

Assumptions:

- Topics may be correlated
- Word-topic distributions are separable

Contributions

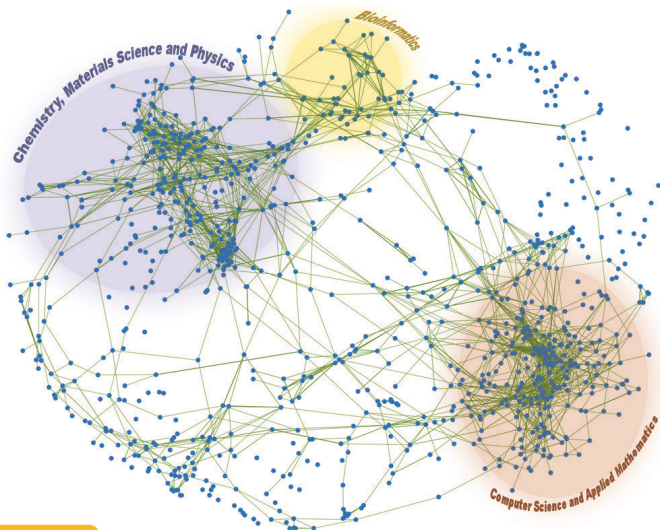
- ① Anchor Selection
 - Combinatorial rather than ILP
 - Stable in the presence of noise
 - polynomial sample complexity
- ② Recovery step
 - Previous matrix-inversion approach sensitive to noise
 - Replaced with Gradient-based inference
- ③ Empirical comparison of algorithms

- 1 Introduction
- 2 Topic Modelling
- 3 Algorithm
- 4 Efficiently Finding Anchor Words
- 5 Topic Recovery via Bayes' Rule
- 6 Experimental Results
- 7 Conclusion

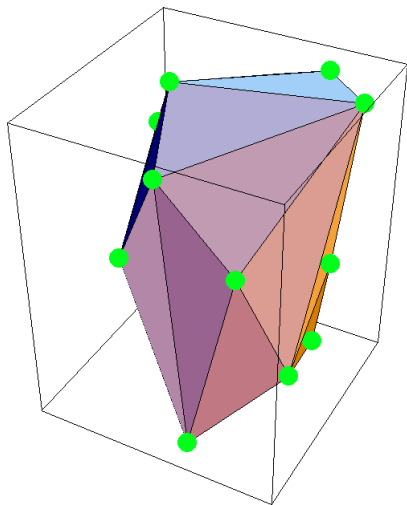
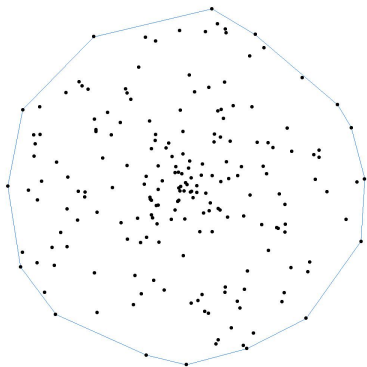
Word-word co-occurrence matrix

	bank	California	Canada	career	careers	employers	employment	federal	human	job	jobs	listings	openings	opportunities	positions	recruiters	resources	resume	resumes	retirement	search	state	texas	unemployment	work
bank	--																								
California	--																								
Canada	1	--																							
career	3	3	--																						
careers			2	9	--																				
employers		2		11	7	--																			
employment	3	26	22	66	10	16	--																		
federal	1	1	5		1		11	--																	
human		4	12	1	1			4	--																
job	34	14	2	49	8	13	92	13	2	--															
jobs		18	6	62	11	27	204	19	2	74	--														
listings		4	2	15	4	9	68	2	55	44		--													
openings		4		7	2	9	28			49	30		--												
opportunities	4	8	3	51	9	13	181	9		84	106	25	19	--											
positions		1		8	2	10	19			16	20	9	13	21	--										
recruiters				10	4	3	9			5	4	2	2	5	2	--									
resources		4	12		1			4	74	3	2						--								
resume		4	3	5		2	3	1	1	10	3			1	2	1		--							
resumes				8	3	3	11			5	16	1		8	5			15	--						
retirement		1	1						3			2		1						--					
search			3	4	6			10			18	6		6	2			3	1		--				
state			4	1			18		1	12	7	6		3			1			2		--			
texas	2			1			18			12	6		1	2						9			--		
unemployment																						2	2	--	
work			2	1		3	3	2		2	8	2	4	7	5				1			1	2	--	

Words as vertices



Convex Hull



Computing Convex Hull

- Efficient for 2 dimensions - $O(n \log n)$
- Inefficient for $n > 2$ dimensions
- Complexity depends on method and approximation used
- Previous method: ILP
- New method: Recursive greedy
 - 1 Compute subspace span of current convex hull
 - 2 Find point furthest from this sub-span
 - 3 Add point to convex hull
 - 4 Repeat until K points found

TODO: Work out how the whole convex hull - words as vertices work. I think what we have here is wrong because there is no approximation

- ① Introduction
- ② Topic Modelling
- ③ Algorithm
- ④ Efficiently Finding Anchor Words
- ⑤ **Topic Recovery via Bayes' Rule**
- ⑥ Experimental Results
- ⑦ Conclusion

Problem

TODO: What is topic recovery

Previous method

TODO: Overview of approach

Matrix Inversion

TODO: How matrix inversion works TODO: How this recovered topics

New method

TODO: Overview of approach

Bayes' Rule

TODO: Bayes rule and how it relates to new method

- 1 Introduction
- 2 Topic Modelling
- 3 Algorithm
- 4 Efficiently Finding Anchor Words
- 5 Topic Recovery via Bayes' Rule
- 6 Experimental Results**
- 7 Conclusion

Experiments

TODO: overview of the experiments run

Metrics

TODO: Metrics

Documents

TODO: Talk about semi-synthetic documents, real documents and the need for both

Results

TODO: describe results. Iterate through each experiment, and each document type, reporting the computed metrics for each. This may need to be split up into more slides by either experiment or document type

- 1 Introduction
- 2 Topic Modelling
- 3 Algorithm
- 4 Efficiently Finding Anchor Words
- 5 Topic Recovery via Bayes' Rule
- 6 Experimental Results
- 7 Conclusion

Summary

TODO: Put the paper's conclusion into dot point form

Comments

TODO: Do we need to comment on the paper?

Are there things that we wish they had reported but didn't?

Are there things that we really liked that they reported?

Check the marking guidelines about what exactly we need here

Future Work

TODO: They didn't have a future work section but they really should have. We can make one up and maybe comment that they didn't put a future work section

Thanks!

Any questions please email either of us:

Vanush Vaswani

vvas****@uni.sydney.edu.au

Kristy Hughes

khug2372@uni.sydney.edu.au