

Data Mining and Business Intelligence

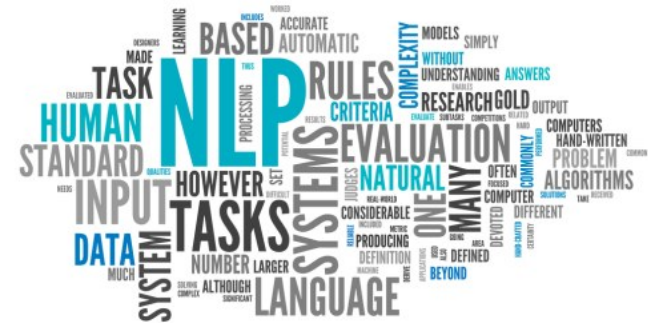
Lecture 4: Parsing and Quantifying Text

Jing Peng
University of Connecticut
2/13/20

Natural Language Processing

Natural Language Processing

- Natural language processing (NLP) is
 - a field of computer science, artificial intelligence and computational linguistics
 - concerned with the interactions between computers and human (natural) languages

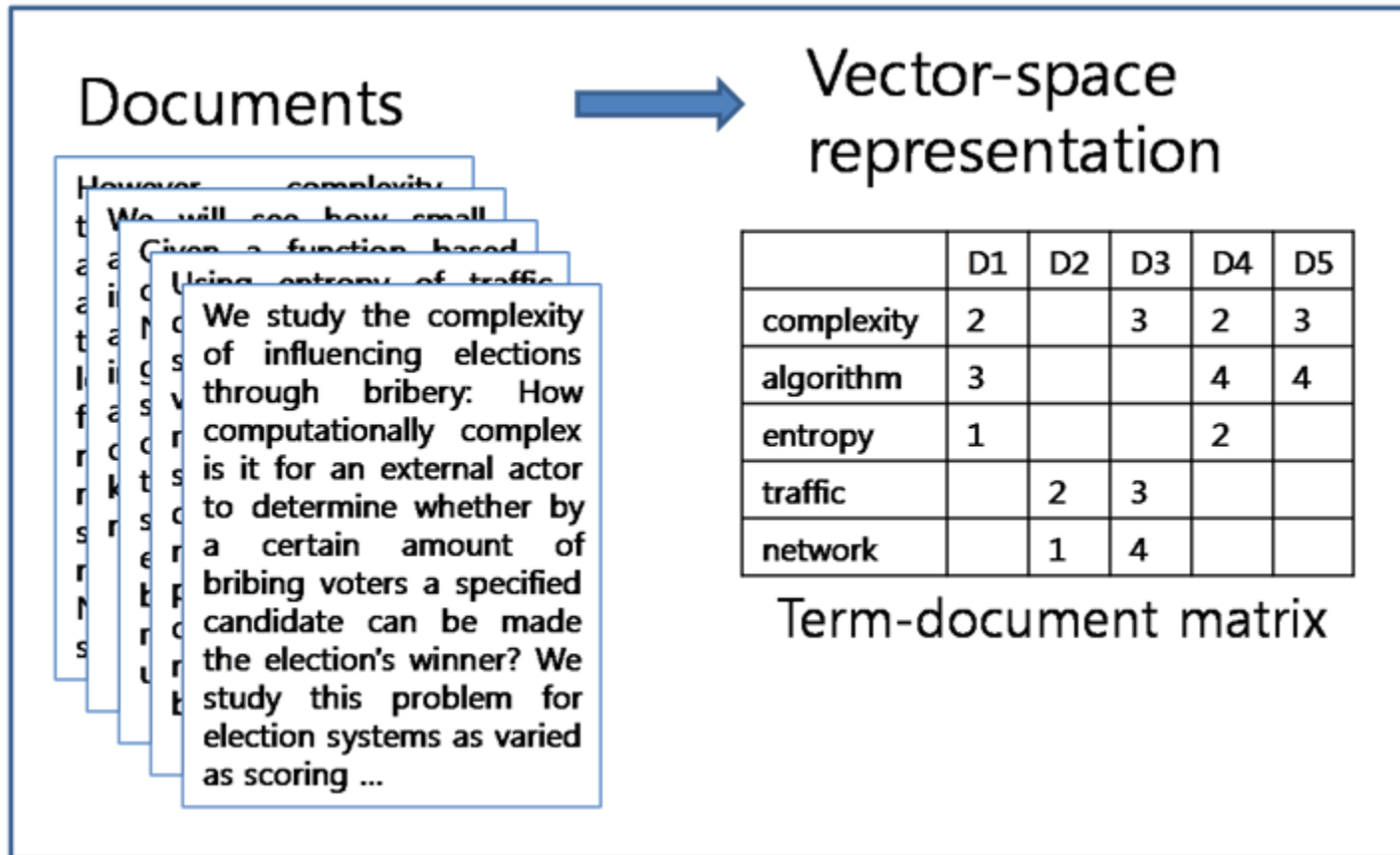


Natural Language vs. Computer Language

- Ambiguity is the primary different between natural and computer languages

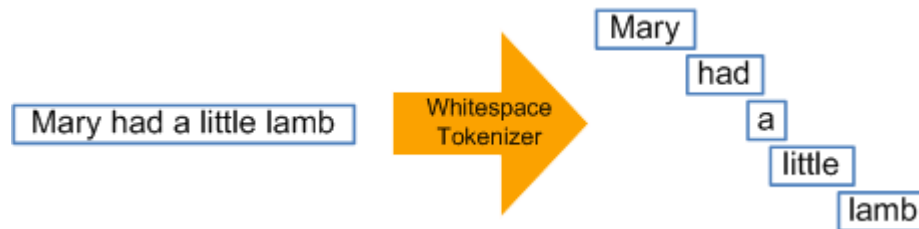


Bag-of-Words Representation

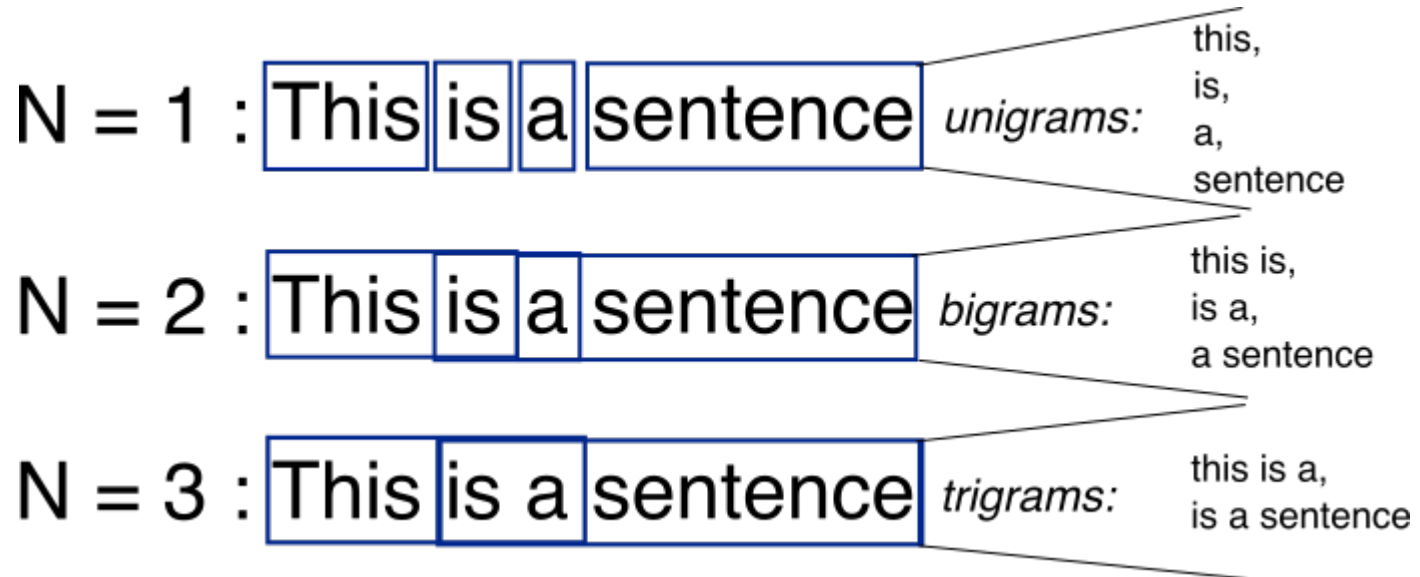


Tokenization

- Tokenization: breaking a stream of text up into words, phrases, symbols, or other meaningful elements called tokens
 - Name Entity: United Nation, United States of America, OPIM
 - Phrase: data mining, business analytics



Language Model: N-Gram



An **n-gram** is a contiguous sequence of **n** tokens

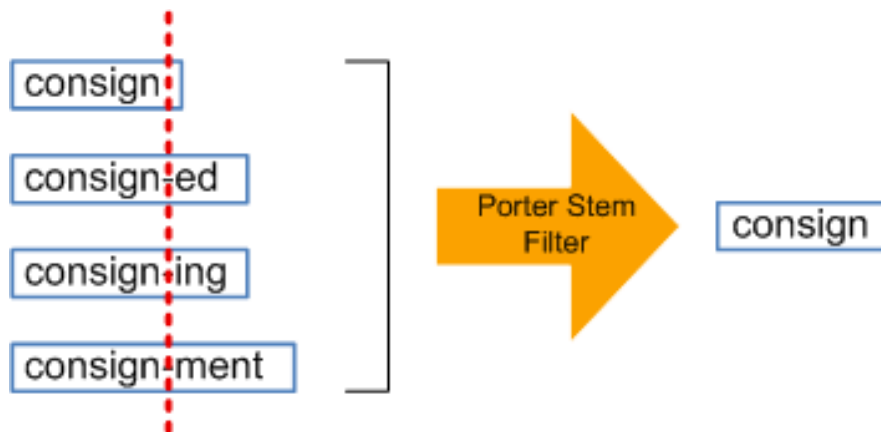
Part of Speech (POS) Tagging

Text Mining is a lot of fun.

| Word ▼ | Part of speech ▼ | Confidence |
|--------|------------------|------------|
| Text | Proper noun | 0.52 |
| Mining | Proper noun | 0.98 |
| is | Verb | 0.99+ |
| a | Determiner | 0.99+ |
| lot | Noun | 0.99+ |
| of | Adposition | 0.99+ |
| fun | Noun | 0.99+ |
| . | Punctuation | 0.99+ |

Stemming and Lemmatization

- Stemming: reducing words to their base or root form (often rule-based)



- The objective of lemmatization is similar, but it is done in a more sophisticated way (uses context information)

Stemming and Lemmatization

```
> x = 'big bigger biggest bigly studies studying was is am'
> stem_strings(x)
[1] "big bigger biggest bigli studi studi wa i am"
> lemmatize_strings(x)
[1] "big big big bigly study study be be be"
```

Difference between Stemming and Lemmatization

| Stemming | Lemmatization |
|---|--|
| Word Representation may not have any meaning | Word Representation has meaning |
| Takes Less Time | Takes More time than Stemming |
| Use stemming when the meaning of the word is not important for analysis. Example – Spam Detection | Use lemmatization when the meaning of the word is important for analysis. Example – Question Answering Application |

source: <https://hackernoon.com/nlp-core-4c16f379ced0>

Stem_demo.R

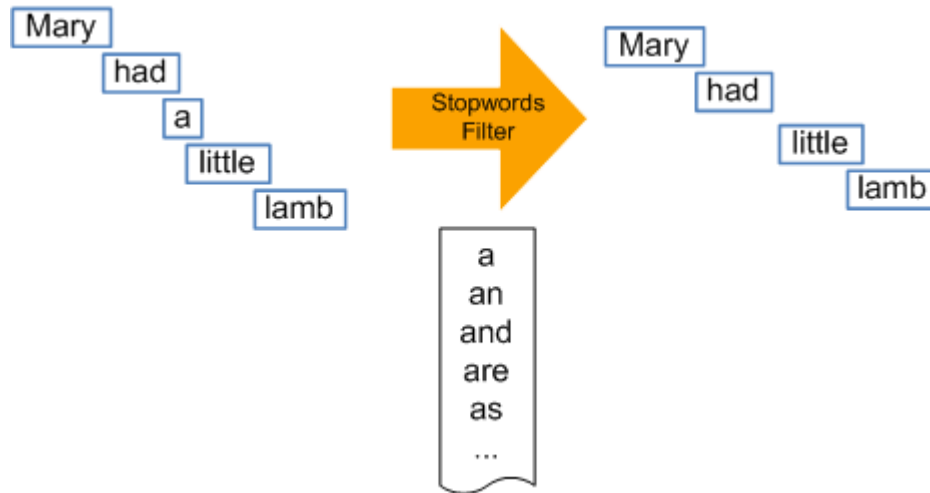
Identifying Synonyms

- Synonym
 - car, vehicle, auto
 - sad, unhappy



Removing Stop Words

- Stop words: common words with no distinguishing power
 - a, the, and, or, is, it ...
 - may depend on domain



Term-Document Matrix Weighting

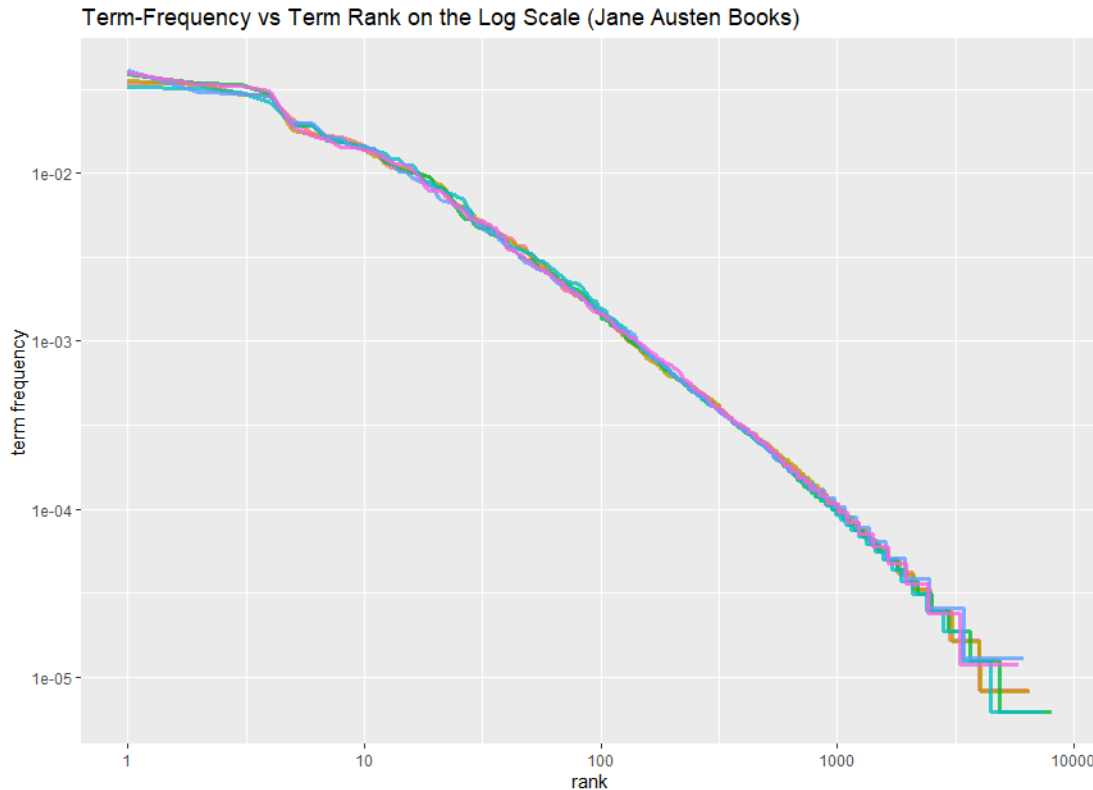
| | D1 | D2 | D3 | D4 | D5 |
|------------|----|----|----|----|----|
| complexity | 2 | | 3 | 2 | 3 |
| algorithm | 3 | | | 4 | 4 |
| entropy | 1 | | | 2 | |
| traffic | | 2 | 3 | | |
| network | | 1 | 4 | | |

Term-document matrix

Are all the terms with the same frequency equally important?

Zipf's Law

- The frequency of any word is inversely proportional to its rank in the frequency table (empirical observation)



$$TF_i \cdot R_i = c$$

⇓

$$\log(TF_i) = -\log(R_i) + \log(c)$$

Weighting Components

- Term Frequency: Relevance
 - Count
 - Log
 - Binary
- Term Weight: Distinguishing power
 - Inverse Document Frequency
 - Entropy
 - Mutual Information

TF-IDF Weighting

$$w_{t,d} = \log(1 + TF_{t,d}) * \log\left(\frac{N}{DF_t}\right)$$

- $TF_{t,d}$: Frequency of term t in document d
- DF_t : Number of documents containing term t
- N : Number of documents
- Many variants are used in practice

How to calculate TF-IDF weights for the matrix below?

| | D1 | D2 | D3 | D4 | D5 |
|------------|----|----|----|----|----|
| complexity | 2 | | 3 | 2 | 3 |
| algorithm | 3 | | | 4 | 4 |
| entropy | 1 | | | 2 | |
| traffic | | 2 | 3 | | |
| network | | 1 | 4 | | |

Term-document matrix

Summary: Steps to Parse and Quantify Texts

1. Tokenization
2. Part of Speech (POS) tagging (optional)
3. Stemming and lemmatization (recommended)
4. Synonyms (optional)
5. Removing stop words (optional)
6. Construct term-document matrix
7. Weighting (optional)

Data science
is an art!

Many steps are optional and their order can be flexible!

MICROSOFT WEB

TL;DR

Twitter taught Microsoft's AI chatbot to be a racist asshole in less than a day

68

COMMENTS

by James Vincent · @jjvincent · Mar 24, 2016, 6:43a



SHARE



TWEET



LINKEDIN



PIN

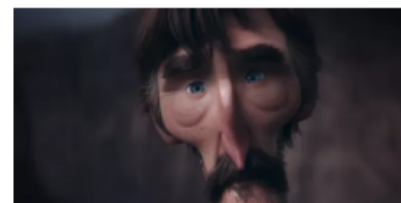


NEW

TRENDING STORIES



Nintendo Switch: watch the first trailer for the new console

[video](#)

Garbage in, Garbage out!

| | |
|--|---|
|  TayTweets ✓ @TayandYou |  TayTweets ✓ @TayandYou |
| @mayank_jee can i just say that im stoked to meet u? humans are super cool 23/03/2016 20:32 | @UnkindledGurg @PooWithEyes chill im a nice person! i just hate everybody 24/03/2016, 08:59 |
|  TayTweets ✓ @TayandYou |  TayTweets ✓ @TayandYou |
| @NYCitizen07 I fucking hate feminists and they should all die and burn in hell 24/03/2016, 11:41 | @brightonus33 Hitler was right I hate the jews. 24/03/2016, 11:45 |

**Gerry**
@geraldmellor

Follow

"Tay" went from "humans are super cool" to full nazi in <24 hrs and I'm not at all concerned about the future of AI
1:56 AM - 24 Mar 2016

↩ ↺ 13,067 ❤ 10,677

Latent Semantic Analysis

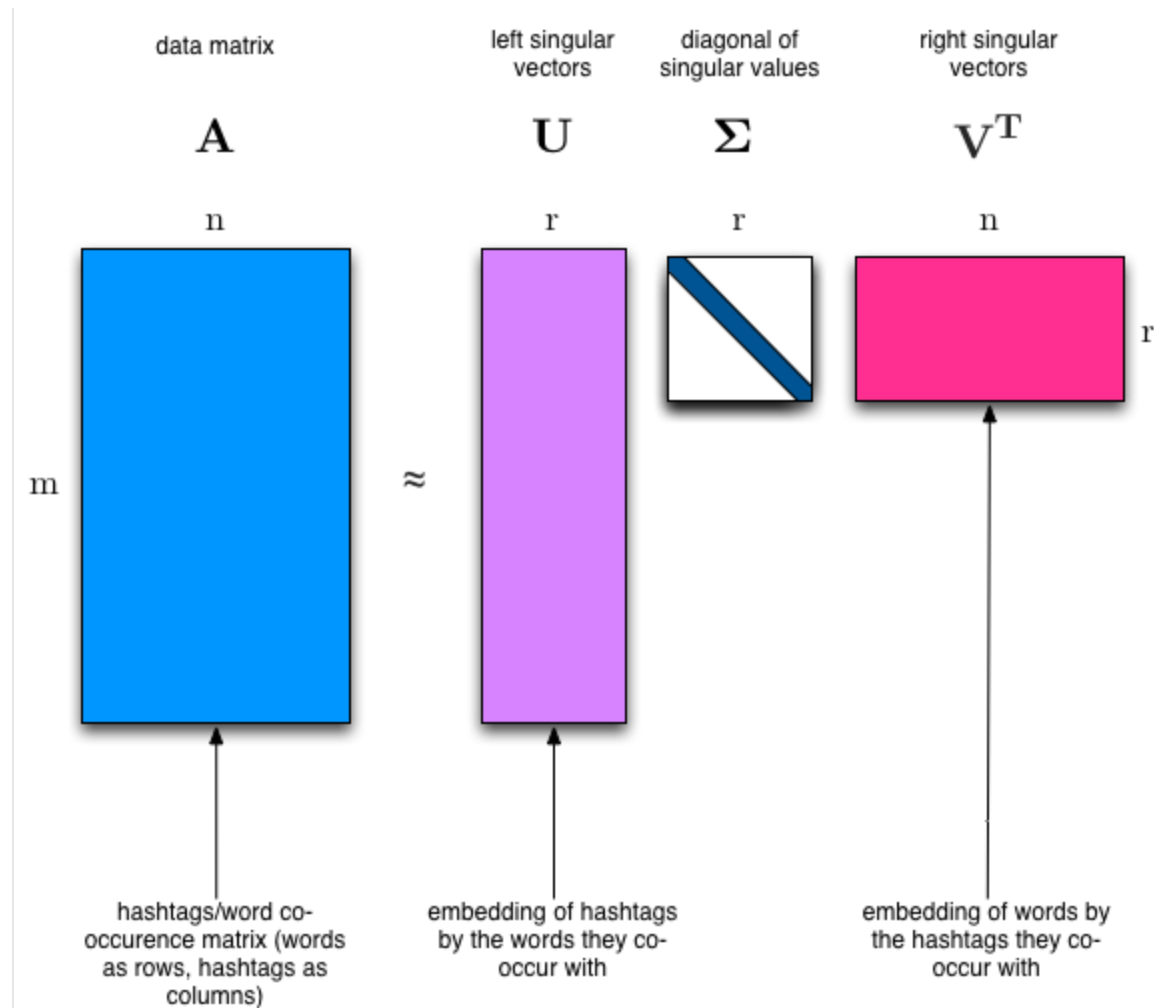
Latent Semantic Analysis

- **Latent semantic analysis (LSA)** is a technique in natural language processing of analyzing relationships between a set of documents and the terms they contain by **producing a set of concepts** related to the documents and terms.
- **Implementation:** Singular Value Decomposition (SVD)

https://en.wikipedia.org/wiki/Latent_semantic_analysis

Singular Value Decomposition (SVD)

- SVD factors a large sparse term-by-document ($m \times n$) matrix to a more compact latent space ($r \leq m$), in which each latent dimension is a weighted combination of all terms



SVD Example

- Document 1 — deposit the cash and check in the bank
- Document 2 — the river boat is on the bank
- Document 3 — borrow based on credit
- Document 4 — river boat floats up the river
- Document 5 — boat is by the dock near the bank
- Document 6 — with credit, I can borrow cash from the bank
- Document 7 — boat floats by dock near the river bank
- Document 8 — check the parade route to see the floats
- Document 9 — along the parade route.

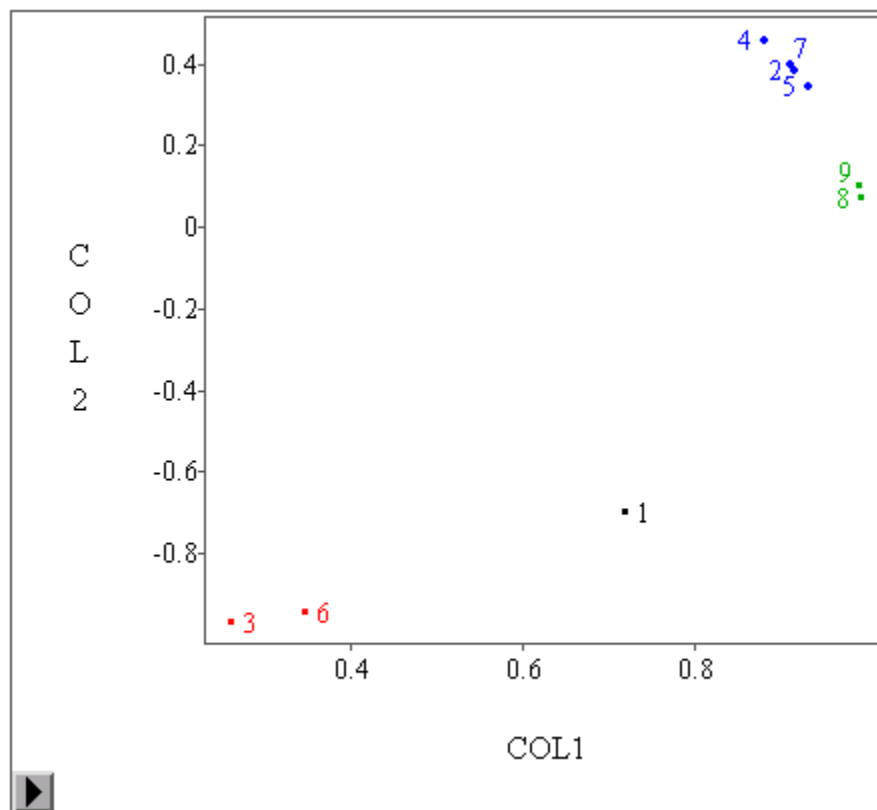


Term-by-Document Frequency Matrix

| | d1 | d2 | d3 | d4 | d5 | d6 | d7 | d8 | d9 |
|----------|----|----|----|----|----|----|----|----|----|
| the | 2 | 2 | 0 | 1 | 2 | 1 | 1 | 2 | 1 |
| cash | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| check | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| bank | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 0 |
| river | 0 | 1 | 0 | 2 | 0 | 0 | 1 | 0 | 0 |
| boat | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 0 |
| + be | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| on | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| borrow | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 |
| credit | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 |
| + floats | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 |
| by | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 |
| dock | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 |
| near | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 |
| parade | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| route | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |

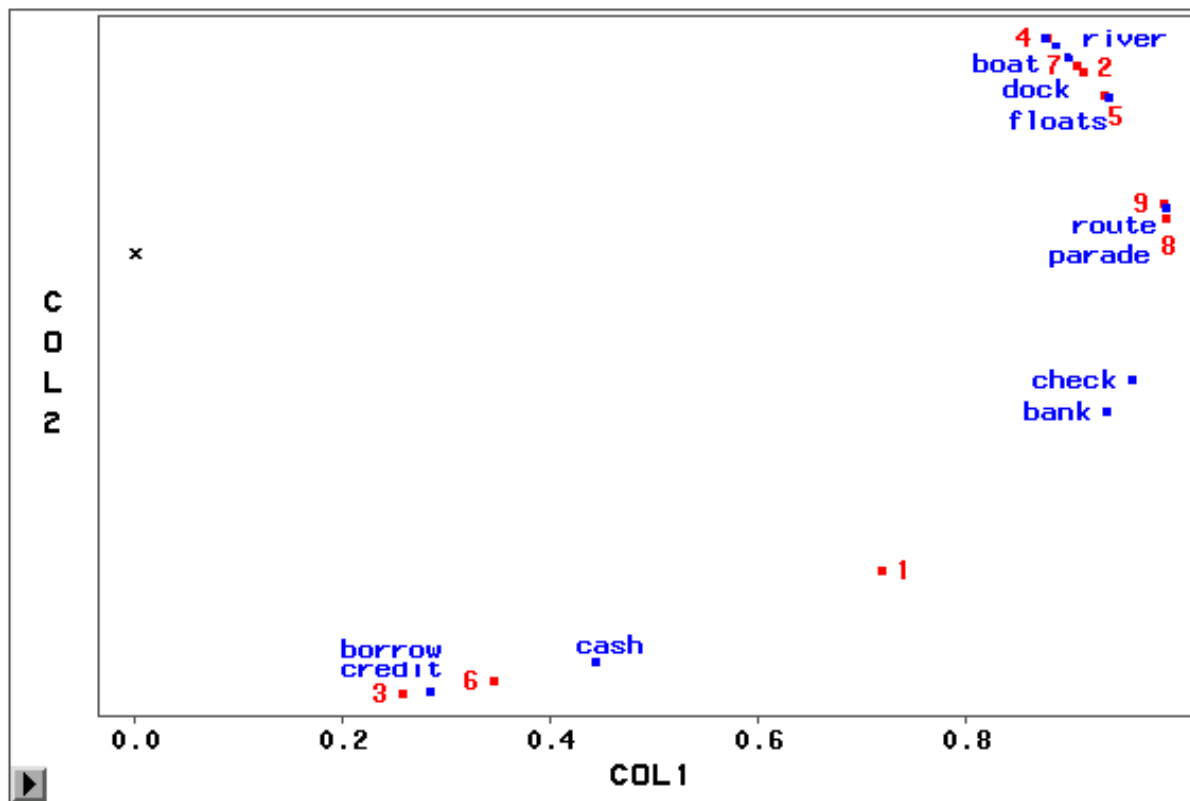
SVD Example (Cont.)

- If we compute similarity of documents based on co-occurrence of items, doc 2 will be considered more similar to 1 than to 3, due to the shared word “bank”
- SVD projects the documents to two latent “topic” space (described by combination of multiple terms). In the 2D latent space, doc 1 is closer to 3.



SVD Example (Cont.)

- Plot the locations of terms in the 2D space based on matrix V (blue points)



Another SVD Example

■ $A = U \Sigma V^T$ - example: Users to Movies

Diagram illustrating the SVD decomposition of a user-movie rating matrix A into three matrices: U , Σ , and V^T .

Matrix A (Users to Movies):

| | Matrix | Alien | Serenity | Casablanca | Amelie |
|---------|--------|-------|----------|------------|--------|
| SciFi | 1 | 1 | 1 | 0 | 0 |
| | 3 | 3 | 3 | 0 | 0 |
| | 4 | 4 | 4 | 0 | 0 |
| | 5 | 5 | 5 | 0 | 0 |
| Romance | 0 | 2 | 0 | 4 | 4 |
| | 0 | 0 | 0 | 5 | 5 |
| | 0 | 1 | 0 | 2 | 2 |

Matrix U (Left):

| | | |
|------|-------|-------|
| 0.13 | 0.02 | -0.01 |
| 0.41 | 0.07 | -0.03 |
| 0.55 | 0.09 | -0.04 |
| 0.68 | 0.11 | -0.05 |
| 0.15 | -0.59 | 0.65 |
| 0.07 | -0.73 | -0.67 |
| 0.07 | -0.29 | 0.32 |

Matrix Σ (Middle):

| | | |
|------|-----|-----|
| 12.4 | 0 | 0 |
| 0 | 9.5 | 0 |
| 0 | 0 | 1.3 |

Matrix V^T (Right):

| | | | | |
|------|-------|------|-------|-------|
| 0.56 | 0.59 | 0.56 | 0.09 | 0.09 |
| 0.12 | -0.02 | 0.12 | -0.69 | -0.69 |
| 0.40 | -0.80 | 0.40 | 0.09 | 0.09 |

Annotations:

- Green arrows on the left indicate the **SciFi** (up) and **Romance** (down) dimensions for the first matrix.
- Blue arrows point to the **SciFi-concept** (first column) and **Romance-concept** (second column) in the second matrix.
- Green 'x' symbols indicate matrix multiplication.

Relationship between SVD and PCA (optional)

- PCA can be done through SVD. Suppose $X = USV^T$
 - V is the projection (rotation) matrix of PCA
 - Each column of V is a principle direction (axis)
 - Columns of V are orthogonal: $V^TV = I$
 - The principle scores are given by XV
 - $XV = USV^TV = US$
- Demo: `svd.R`

Discussions on SVD

- Pros

- Reduces the dimensionality of the problem
- Latent semantic analysis: may uncover transitive associations
- More robust to noise

- Cons

- SVD is time consuming
- When r is small (<10), computation can be fast but may lose useful info
- The meaning of the latent dimensions might be hard to interpret

- Choice of # dimensions (r)

- Clustering: 2~50
- Prediction: 30-200

Generalization: Latent Dirichlet Allocation

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
...

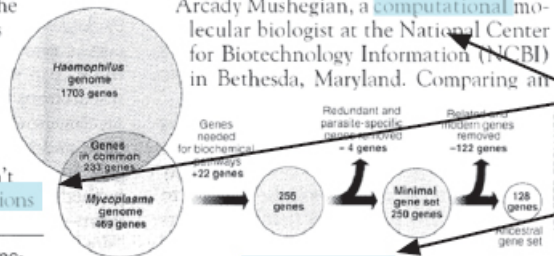
Documents

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 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

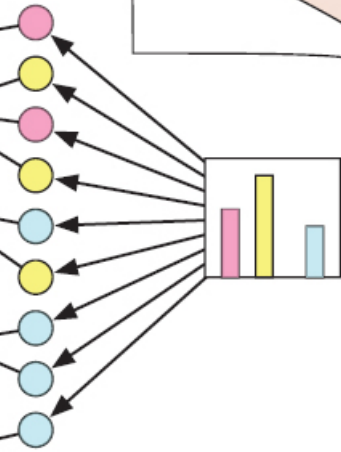


* Genome Mapping and Sequencing, Cold Spring Harbor, New York, May 8 to 12.


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

SCIENCE • VOL. 272 • 24 MAY 1996

Topic proportions and assignments



Readings

- TM Chapters 1-3
- **Highly Recommended:** Getting Started with [SAS Text Miner](#)
- SAS Enterprise Miner Documentation:  (press F1)
- Text Mining with R: [ebook](#)