42578 Advanced Business Analytics

Text Analytics

Stanislav Borysov [stabo@dtu.dk]

Text data

- **Language is one of the primary means of communicating**

- **Huge in size**
  - Google processes 5.13B queries/day (2013)
  - Twitter receives 340M tweets/day (2012)
  - Facebook has 2.5 PB of user data + 15 TB/day (4/2009)
  - eBay has 6.5 PB of user data + 50 TB/day (5/2009)
  - …

- **80% data is unstructured (IBM, 2010)**

Adapted from “Introduction to Text Mining”, Hongning Wang (CS@UVa)

What is “Text Mining”?

- “Text mining, also referred to as **text data mining**, roughly equivalent to text analytics, refers to the process of deriving high-quality information from text.” - wikipedia

- “Another way to view text data mining is as a process of **exploratory** data analysis that leads to **heretofore unknown** information, or to answers for questions for which the answer is not currently known.” - Hearst, 1999

Adapted from “Introduction to Text Mining”, Hongning Wang (CS@UVa)

How to perform text mining?

- Text Mining = Data Mining + Text Data

Adapted from “Introduction to Text Mining”, Hongning Wang (CS@UVa)

Text mining around us

- Sentiment analysis

Adapted from "Introduction to Text Mining", Hongning Wang (CS@UVa)

Text mining around us

- Sentiment analysis

Adapted from "Introduction to Text Mining", Hongning Wang (CS@UVa)

Text mining around us

- Document summarization

Adapted from “Introduction to Text Mining”, Hongning Wang (CS@UVa)

Text mining around us

- Document summarization

Adapted from “Introduction to Text Mining”, Hongning Wang (CS@UVa)

Text mining around us

- Trends

https://medium.com/analytics-vidhya/the-scuffle-between-two-algorithms-neural-network-vs-support-vector-machine-16abe0eb4181

Adapted from “Introduction to Text Mining”, Hongning Wang (CS@UVa)

Text mining around us

- News recommendation

Adapted from “Introduction to Text Mining”, Hongning Wang (CS@UVa)

Text mining around us

- Text analytics in financial services

https://plot.ly/~suman.dna/4.embed

Adapted from “Introduction to Text Mining”, Hongning Wang (CS@UVa)

Text mining around us

- Text analytics in transportation

+

Adapted from “Introduction to Text Mining”, Hongning Wang (CS@UVa)

Text mining around us

Text Mining has numerous applications in any industry

Government

Detect fraudulent activity.
Spot emerging trends and public concerns.

Finance

Retention of current customer base using call center transcriptions or transcribed audio. Identification of potentially fraudulent activities.

Insurance

Identify fraudulent claims.
Track competitive intelligence.
Brand management

Retail

Identify the most profitable customers and the underlying reasons for their loyalty.
Brand management

Manufacturing

Reduce time to detect root cause of product issues.
Identify trends in market segments.

Telecommunications

Help prevent churn and suggest up-sell/cross-sell opportunities for individual customers.

Life Sciences

Identify adverse events.
Recommend appropriate research materials.

MS4252 Big Data Analytics, City University of Hong Kong

Some common text mining tasks

Adapted from “Introduction to Text Mining”, Hongning Wang (CS@UVa)

Some common text mining tasks

- Document categorization
  - Adding structures to a text corpus

Adapted from "Introduction to Text Mining", Hongning Wang (CS@UVa)

Some common text mining tasks

- Named Entity Recognition

https://imanage.com/blog/named-entity-recognition-ravn-part-1/
https://wordlift.io/blog/en/entity/named-entity-recognition/

Adapted from “Introduction to Text Mining”, Hongning Wang (CS@UVa)

Some common text mining tasks

- Text clustering
  - Identifying structures in a text corpus

https://imanage.com/blog/named-entity-recognition-ravn-part-1/
https://wordlift.io/blog/en/entity/named-entity-recognition/

Adapted from “Introduction to Text Mining”, Hongning Wang (CS@UVa)

Some common text mining tasks

- Topic modeling
  - Identifying structures in a text corpus

Adapted from "Introduction to Text Mining", Hongning Wang (CS@UVa)

Some common text mining tasks

- Document retrieval
  - Given a query q rank documents by relevance (similarity)

Adapted from “Introduction to Text Mining”, Hongning Wang (CS@UVa)

Challenges in text mining

- **Data collection is “free text”**
  - Data is not well-organised
    - Semi-structured or unstructured
  - Natural language text contains ambiguities on many levels
    - Lexical, syntactic, semantic, and pragmatic
  - Learning techniques for processing text typically need annotated training examples
    - Expensive to acquire at scale

- **Where to start?**

Example: Stock price prediction

- **model DJI_b(t) = f(news_headlines(t))**

- **Data row for t = 2016-06-24:**
  - DJI_b = 0 // decreased or stayed the same; “1” - increased
  - news_headline_1 = ‘David Cameron to Resign as PM After EU Referendum.’
  - …

Tokenisation

Document → List of tokens

‘David Cameron to Resign as PM After EU Referendum.’

['David', 'Cameron', 'to', 'Resign', 'as', 'PM', 'After', 'EU', 'Referendum', '.']

Tokenization

Document → List of tokens

‘David Cameron to Resign as PM After EU Referendum.’

1-gram ['David', 'Cameron', 'to', 'Resign', 'as', 'PM', 'After', 'EU', 'Referendum', '.']

2-gram [('David', 'Cameron'), ('Cameron', 'to'), ('to', 'Resign'), ('Resign', 'as'), ('as', 'PM'), ('PM', 'After'), ('After', 'EU'), ('EU', 'Referendum'), ('Referendum', '.')]

...

n-gram

Word embeddings

1-gram

Words → Vectors

Document	**Categorical variables**	**One-hot encoding**
['David',	0: ‘David'	‘David’ = (**1**, 0, 0, 0, 0, 0, 0, 0, 0, 0)
'Cameron',	1: ’Cameron’	‘Cameron’ = (0, **1**, 0, 0, 0, 0, 0, 0, 0, 0)
'to',	2: ‘to’	‘to’ = (0, 0, **1**, 0, 0, 0, 0, 0, 0, 0)
'Resign',	3: ‘Resign’	‘Resign’ = (0, 0, 0, **1**, 0, 0, 0, 0, 0, 0)
'as',	4: ’as’	‘as’ = (0, 0, 0, 0, **1**, 0, 0, 0, 0, 0)
'PM',	5: ’PM’	‘PM’ = (0, 0, 0, 0, 0, **1**, 0, 0, 0, 0)
'After',	6: ‘After’	‘After’ = (0, 0, 0, 0, 0, 0, **1**, 0, 0, 0)
'EU',	7: ’EU’	‘EU’ = (0, 0, 0, 0, 0, 0, 0, **1**, 0, 0)
'Referendum',	8: ‘Referendum'	‘Referendum’ = (0, 0, 0, 0, 0, 0, 0, 0, **1**, 0)
‘.']	9: ’.’	‘.’ = (0, 0, 0, 0, 0, 0, 0, 0, 0, **1**)

Word embeddings

1-gram

Words → Vectors

Document

['David',
'Cameron',
'to',
'Resign',
'as',
'PM',
'After',
'EU',
'Referendum',
'.']

Categorical variables

0: ‘David'
1: ’Cameron’
2: ‘to’
3: ‘Resign’
4: ’as’
5: ’PM’
6: ‘After’
7: ’EU’
8: ‘Referendum'
9: ’.’

vocabulary

One-hot encoding

‘David’ = (**1**, 0, 0, 0, 0, 0, 0, 0, 0, 0)
‘Cameron’ = (0, **1**, 0, 0, 0, 0, 0, 0, 0, 0)
‘to’ = (0, 0, **1**, 0, 0, 0, 0, 0, 0, 0)
‘Resign’ = (0, 0, 0, **1**, 0, 0, 0, 0, 0, 0)
‘as’ = (0, 0, 0, 0, **1**, 0, 0, 0, 0, 0)
‘PM’ = (0, 0, 0, 0, 0, **1**, 0, 0, 0, 0)
‘After’ = (0, 0, 0, 0, 0, 0, **1**, 0, 0, 0)
‘EU’ = (0, 0, 0, 0, 0, 0, 0, **1**, 0, 0)
‘Referendum’ = (0, 0, 0, 0, 0, 0, 0, 0, **1**, 0)
‘.’ = (0, 0, 0, 0, 0, 0, 0, 0, 0, **1**)

Word embeddings

2-gram

Words → Vectors

Document
[('David', 'Cameron'),
('Cameron', ‘to’),
…]

Categorical variables
0: ('David', 'Cameron')
1: ('Cameron', ‘to’)
…

One-hot encoding
('David', 'Cameron') = (**1**, 0, …)
('Cameron', ‘to’) = (0, **1**, …)
…

Document vectorisation: Bag-of-Words (BoW)

- Document = sum of vectorised words
- **Grammar and word ordering is ignored**

Document vectorisation: Bag-of-Words (BoW)

- Document = sum of vectorised words
- Doc1 = 'David Cameron to Resign as PM After EU Referendum.'

vocabulary size = 10

	David	Cameron	to	Resign	as	PM	After	EU	Referendum	.
Doc1	1	1	1	1	1	1	1	1	1	1

Document vectorisation: Bag-of-Words (BoW)

- Document = sum of vectorised words

- Doc1 = 'David Cameron to Resign as PM After EU Referendum.'
- Doc2 = ‘Prime Minister Resigns After European Union referendum.'
- Doc3 = 'Giant panda twins born at a zoo in European Union.’

vocabulary size = 24

	David	Cameron	to	Resign	as	PM	After	EU	Referendum	.	Prime	Minister	Resigns	European	Union	referendum	Giant	panda	twins	born	at	a	zoo	in
Doc1	1	1	1	1	1	1	1	1	1	1														
Doc2							1			1	1	1	1	1	1	1								
Doc3										1				1	1		1	1	1	1	1	1	1	1

Document vectorisation: Bag-of-Words (BoW)

- Document = sum of vectorised words

- Doc1 = 'David Cameron to Resign as PM After EU Referendum.'
- Doc2 = ‘Prime Minister Resigns After European Union referendum.'
- Doc3 = 'Giant panda twins born at a zoo in European Union.’

vocabulary size = 24

	David	Cameron	to	Resign	as	PM	After	EU	Referendum	.	Prime	Minister	Resigns	European	Union	referendum	Giant	panda	twins	born	at	a	zoo	in
Doc1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Doc2	0	0	0	0	0	0	1	0	0	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
Doc3	0	0	0	0	0	0	0	0	0	1	0	0	0	1	1	0	1	1	1	1	1	1	1	1

document-term matrix

Document vectorisation: Bag-of-Words (BoW)

These vectors can now be used for

- a ML model for DJI_b(t) = f(news_headlines(t))
- Clustering
- Information retrieval

…

vocabulary size = 24

	David	Cameron	to	Resign	as	PM	After	EU	Referendum	.	Prime	Minister	Resigns	European	Union	referendum	Giant	panda	twins	born	at	a	zoo	in
Doc1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Doc2	0	0	0	0	0	0	1	0	0	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
Doc3	0	0	0	0	0	0	0	0	0	1	0	0	0	1	1	0	1	1	1	1	1	1	1	1

document-term matrix

Document-term matrix (D)

d_{ij}
- $tf(d_i,t_j)$ - number of the term j in the document i
- $tf\text{-}idf(d_i,t_j)$ - normalised tf (discussed in a few slides)
- …

- N - number of documents
- V - vocabulary size - quickly gets to thousands! - For English ~ 10^6
- Usually super-sparse

Document retrieval

- Query document $\boldsymbol{q}$
- Rank documents $\boldsymbol{d_i}$ by similarity to $\boldsymbol{q}$

- **How to do it?**

Document retrieval

- Query document $\boldsymbol{q} = (q_1, \ldots, q_V)$
- Rank documents $\boldsymbol{d_i} = (d_{i1}, \ldots, d_{iV})$ by similarity to $\boldsymbol{q}$

- **Boolean similarity**
  - bool($\boldsymbol{q}$, $\boldsymbol{d_i}$) = # of terms that both present in $\boldsymbol{q}$ and $\boldsymbol{d_i}$

- **Cosine similarity**
  - CosSim($\boldsymbol{q}$, $\boldsymbol{d_i}$) = 1 - cos(angle between $\boldsymbol{q}$ and $\boldsymbol{d_i}$)

$$= 1 - \frac{\sum_{j=1}^{V} q_j d_{ij}}{\sqrt{\sum_{j=1}^{V} q_j^2}\sqrt{\sum_{j=1}^{V} d_{ij}^2}}$$

- …

Document retrieval

- Query document **Q**
- Rank documents **D** by similarity to **Q**

- Doc1 = 'David Cameron to Resign as PM After EU Referendum.'
- Doc2 = ‘Prime Minister Resigns After European Union referendum.'
- Doc3 = 'Giant panda twins born at a zoo in European Union.’

Q = ‘Union After panda.’

	David	Cameron	to	Resign	as	PM	After	EU	Referendum	.	Prime	Minister	Resigns	European	Union	referendum	Giant	panda	twins	born	at	a	zoo	in	Boolean similarity	Cosine similarity
Doc1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0.31
Doc2	0	0	0	0	0	0	1	0	0	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	**3**	**0.53**
Doc3	0	0	0	0	0	0	0	0	0	1	0	0	0	1	1	0	1	1	1	1	1	1	1	1	**3**	0.45

Term frequency - inverse document frequency (tf-idf)

- Common words which are present in almost all documents are not helpful for differentiating
- tf-idf penalises frequent words and increase importance of rare words

- **d_{ij} are weighted: *tf-idf = tf * idf***
  - *tf(t, d)* = number of term *t* in a document *d* (what we had before)
  - *idf(t)* = log[*N* / *df(t)*]
    - *N* is the total number of documents
    - *df(t)* is a number of documents containing term *t*

Term frequency - inverse document frequency (tf-idf)

- Common words which are present in almost all documents are not helpful for differentiating
- tf-idf penalises frequent words and increase importance of rare words

- **d_{ij} are weighted: *tf-idf = tf * idf***
  - *tf(t, d)* = number of term *t* in a document *d* (what we had before)
  - *idf(t)* = log[*N* / *df(t)*]
    - *N* is the total number of documents
    - *df(t)* is a number of documents containing term *t*

- Normalisation of documents (rows of **D**)
  - Document length (sum of d_{ij} over *j*)
  - Cosine (square root of sum of d^2_{ij} over *j*)
  - Max number of term appears in all documents
  - …

Document retrieval: tf-idf

- Query document **Q**
- Rank documents **D** by similarity to **Q**

- Doc1 = 'David Cameron to Resign as PM After EU Referendum.'
- Doc2 = ‘Prime Minister Resigns After European Union referendum.'
- Doc3 = 'Giant panda twins born at a zoo in European Union.’

Q = ‘Union After panda.’

	David	Cameron	to	Resign	as	PM	After	EU	Referendum	.	Prime	Minister	Resigns	European	Union	referendum	Giant	panda	twins	born	at	a	zoo	in	Boolean similarity	Cosine similarity	Cosine similarity (tf-idf matrix)
Doc1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0.31	0.19
Doc2	0	0	0	0	0	0	1	0	0	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	**3**	**0.53**	0.38
Doc3	0	0	0	0	0	0	0	0	0	1	0	0	0	1	1	0	1	1	1	1	1	1	1	1	**3**	0.45	**0.39**

Term frequency - inverse document frequency (tf-idf)

- There are many modifications of the tf-idf

Term frequency (if $\mathrm{tf}_{t,d} > 0$)		Document frequency		Document Normalization	
n (natural)	$\mathrm{tf}_{t,d}$	n (no)	1	n (none)	1
l (logarithm)	$1 + \log \mathrm{tf}_{t,d}$	t (idf)	$\log N / \mathrm{df}_t$	c (cosine)	$1 \Big/ \sqrt{\sum_i w_i^2}$
a (augmented)	$0.5 + \frac{0.5 \cdot \mathrm{tf}_{t,d}}{\max_t \mathrm{tf}_{t,d}}$	p (prob idf)	$\max\{0, \log \frac{N - \mathrm{df}_t}{\mathrm{df}_t}\}$	u (pivoted unique)	$1/u$ (see [MRS08])
b (boolean)	1	(idf smooth)	$\log N/(\mathrm{df}_t + 1)$	b (byte size)	$1/\mathrm{CharLength}^{\alpha}, \alpha < 1$
L (log mean)	$\frac{1 + \log \mathrm{tf}_{t,d}}{1 + \mathrm{mean}_{t \in d} \log \mathrm{tf}_{t,d}}$				
d (double log)	$1 + \log(1 + \log \mathrm{tf}_{t,d})$				
(BM25)	$\frac{k \cdot \mathrm{tf}_{t,d}}{k + b \cdot (L - 1) + \mathrm{tf}_{t,d}}$	(BM25)	$\log \frac{N - \mathrm{df}_t + .5}{\mathrm{df}_t + .5}$		
(three val.)	$\min\{\mathrm{tf}_{t,d}, 2\}$				
(log1p)	$\log(1 + \mathrm{tf}_{t,d})$				
(sqrt)	$\sqrt{\mathrm{tf}_{t,d}}$			(manhattan)	$1/\sum_i w_i$

Erich Schubert, “Advanced Topics in Text Mining”, Heidelberg U (2017)

Some common problems

- Lexical semantics and word senses
  - Identifying which sense of a word (i.e. meaning) is used in a sentence, when the word has multiple meanings

Bass: fish

???

Bass: instrument

Context is important!

Check: word2vec, char2vec, …

Adapted from "Introduction to Text Mining", Hongning Wang (CS@UVa)

Text Preprocessing (“Normalisation”)

David
Cameron
to
Resign
as
PM
After
EU
Referendum
.
Prime
Minister
Resigns
European
Union
referendum
Giant
panda
twins
born
at
a
zoo
in

Text Preprocessing (“Normalisation”)

David
Cameron
to
Resign
as
PM
After
EU
Referendum
.
Prime
Minister
Resigns
European
Union
referendum
Giant
panda
twins
born
at
a
zoo
in

stopwords

punctuation

abbreviations

capitalisation

verb tenses

proper nouns

Text Preprocessing (“Normalisation”)

David
Cameron
to
Resign
as
PM
After
EU
Referendum
.
Prime
Minister
Resigns
European
Union
referendum
Giant
panda
twins
born
at
a
zoo
in

stopwords

punctuation

abbreviations

capitalisation

verb tenses

proper nouns

Numbers, named entities, synonyms, homonyms, …

Text Preprocessing (“Normalisation”)

- Most common text preprocessing steps (for the BoW representation)
  - remove punctuation
  - remove / replace numbers
  - lowercase
  - remove stopwords
  - stemming / lemmatising

- **Ad hoc!**

Exercise

- **Part 1 - Spam Classification.ipynb**
  Do Sections 1-3

Recap

- **Text preprocessing steps:**

1. Tokenisation
2. Normalisation (remove punctuation, stopwords, stemming, lowercase, etc)
3. Vectorisation (tokenisation, converting documents to vectors)

Topic Modeling

- D is high dimensional (News example: 73608 documents x 43509 terms)
- Computationally expensive
- Difficult to deal with synonyms and related words
  (“car” vs “automobile” vs “vehicle”)

- **Dimensionality reduction can help!**

Topic Modeling: General Idea

- Topic modelling is decomposing D into sets of concepts (“topics”) and how documents are related to the topics

Topic modelling via matrix decomposition: Latent Semantic Indexing (Analysis) - LSI (LSA)

- Is an indexing and retrieval method that uses SVD to identify patterns in the relationships between the terms and documents.
- Is based on the principle that words that are used in the same contexts tend to have similar meanings.
- A key feature of LSI: its ability to extract the conceptual content of a body of text by establishing associations between those terms that occur in similar contexts.
- Could trace back its history to factor analysis applications in mid 1960s, but it started gaining the popularity in late 80s to early 90s. Nowadays, LSI is being used in many applications on a daily basis.

Latent Semantic Indexing (Analysis) - LSI (LSA)

- It is a Singular Value Decomposition (generalisation of eigenvalue decomposition) of the document-term matrix **D=UΣT***

- **D** is a document-term matrix
- **U** is a document-topic map (“topic distribution”)
- **Σ** is an ordered diagonal matrix of singular values (“topic importance”)
- **T** is a term-topic map (“term distribution”)

Latent Semantic Indexing (Analysis) - LSI (LSA)

- It is a Singular Value Decomposition (generalisation of eigenvalue decomposition) of the document-term matrix **D=UΣT***

- **D** is a document-term matrix
- **U** is a document-topic map (“topic distribution”)
- **Σ** is an ordered diagonal matrix of singular values (“topic importance”)
- **T** is a term-topic map (“term distribution”)

Truncate the matrices to K topics (keep enough…)

Latent Semantic Indexing (Analysis) - LSI (LSA)

- It is a Singular Value Decomposition (generalisation of eigenvalue decomposition) of the document-term matrix **D=UΣT***

- **D** is a do
- **U** is a do
- **Σ** is an c nce")
- **T** is a te

Instead of D, U*Σ can now be used for

- a ML model for DJI_b(t)
- Clustering
- Information retrieval
…

Truncate the matrices to K topics (keep enough…)

LSI results for the news titles: Topics (k=25)

0: 0.963*”number" + 0.107*"year" + 0.063*"people" + 0.055*"million" + 0.050*"world" + 0.048*"say" + 0.044*"killed" + 0.041*"new" + 0.039*"amp" + 0.037*”china"

1: 0.489*"say" + 0.214*"new" + -0.211*"number" + 0.199*"government" + 0.188*"world" + 0.162*"country" + 0.154*"israel" + 0.146*"china" + 0.145*"state" + 0.141*"police"

2: -0.805*"say" + 0.304*"new" + 0.282*"world" + 0.164*"china" + 0.144*"government" + 0.139*"country" + 0.122*"year" + 0.089*"state" + 0.077*"people" + 0.075*"war"

3: 0.489*"world" + -0.402*"police" + 0.377*"china" + -0.282*"israel" + 0.205*"say" + -0.159*"israeli" + 0.147*"korea" + -0.137*"palestinian" + 0.134*"north" + -0.127*"attack"

4: -0.397*"israel" + -0.371*"korea" + -0.341*"north" + 0.310*"police" + 0.286*"new" + 0.196*"say" + 0.164*"world" + -0.157*"state" + -0.154*"south" + 0.153*"year"

5: -0.465*"world" + 0.399*"china" + -0.342*"israel" + 0.338*"korea" + 0.312*"north" + 0.299*"new" + 0.161*"south" + 0.157*"year" + -0.134*"israeli" + -0.130*"palestinian"

6: 0.540*"new" + -0.397*"police" + 0.332*"year" + -0.315*"world" + 0.274*"israel" + -0.229*"china" + -0.213*"people" + -0.168*"korea" + -0.158*"north" + 0.098*"palestinian"

7: 0.851*"year" + -0.409*"new" + 0.108*"police" + -0.107*"government" + 0.088*"old" + -0.080*"number" + -0.073*"people" + 0.069*"prison" + -0.067*"amp" + 0.058*"woman"

8: -0.682*"government" + 0.391*"police" + 0.364*"new" + -0.192*"country" + -0.166*"people" + 0.153*"world" + 0.150*"korea" + 0.135*"north" + 0.125*"woman" + -0.113*"china"

9: -0.763*"china" + 0.331*"korea" + 0.308*"north" + 0.229*"government" + -0.199*"israel" + 0.129*"world" + 0.107*"south" + -0.098*"chinese" + 0.097*"people" + 0.082*"korean"

10: 0.794*"people" + -0.332*"government" + -0.289*"police" + 0.230*"killed" + 0.100*"right" + -0.099*"uk" + 0.087*"year" + -0.075*"world" + 0.073*"human" + -0.070*"nuclear"

…

LSI results for the news titles: Topics (k=25)

Topic #1

Topic #2

Topic #5

Topic #4

Topic #12

Topic #0

LSI results for the news titles: Documents by topics

- “North Korea seen to fire submarine-launched ballistic missile: South Korea”
- ['north', 'korea', 'seen', 'submarinelaunched', 'ballistic', 'missile', 'south', ‘korea']
- **5:** -1.71, **8:** -1.24, **3:** 0.51, **10:** 0.44, …

Topic #5

Topic #8

Topic #3

Topic #10

LSI results for the news titles: Documents by topics

- “North Korea has threatened to conduct a nuclear test in response to a United Nations move towards a probe into the country's human rights violations”
- ['north', 'korea', 'threatened', 'conduct', 'nuclear', 'test', 'response', 'united', 'nation', 'probe', 'country', 'human', 'right', 'violation']
- **5:** -1.11, **15:** 0.79, **8:** -0.74, **12:** -0.74, **1:** 0.74, **18:** -0.69, …

Topic #5

Topic #15

Topic #8

Topic #12

Topic #1

Topic #18

LSI results for the news titles: Documents by topics

- “Argentine Court rules: Orang Utans are "non-human-persons" with human rights and therefore need to be released from zoo”
- ['argentine', 'court', 'rule', 'orang', 'utans', 'nonhumanpersons', 'human', 'right', 'need', 'released', ‘zoo']
- **16:** 0.91, **18:** -0.67, **12:** -0.66, **15:** 0.33, …

Topic #16

Topic #18

Topic #12

Topic #15

LSI beyond Text

- Recommender systems (in two weeks)
- Image processing (“Eigenfaces”)
- ...

LSI problems

- Computationally expensive: Complexity $O(\min\{NV^2, N^2V\})$
  - Monte-Carlo sampling, approximations…
- Difficult to interpret the decomposition results (e.g., negative values)

- **Another approach: Probabilistic modelling**

Probabilistic topic modeling

- **U** is a document-topic map ("topic distribution")
- **T** is a term-topic map ("term distribution")

- **Come up with a probabilistic model that generates documents**
  U ~ P(topics | documents)
  T ~ P(terms | topics)

Erich Schubert, "Advanced Topics in Text Mining", Heidelberg U (2017)

Probabilistic LSI (pLSI)

- Estimate P(D, W)
- Using a latent variable Z (categorical, K topics)

$$P(D,W) = P(D) \sum_{Z} P(Z|D)P(W|Z)$$

- Can be estimated using Expectation-Maximisation algorithm, MC sampling, variational inference, …

https://medium.com/nanonets/topic-modeling-with-lsa-psla-lda-and-lda2vec-555ff65b0b05

Probabilistic LSI (pLSI)

$$P(D,W) = P(D)\sum_{Z} P(Z|D)P(W|Z)$$

- Start with document

$$P(D,W) = \sum_{Z} P(Z)P(D|Z)P(W|Z)$$

- Start with topic

https://medium.com/nanonets/topic-modeling-with-lsa-psla-lda-and-lda2vec-555ff65b0b05

Probabilistic LSI (pLSI)

$$P(D,W) = P(D)\sum_{Z} P(Z|D)P(W|Z)$$

- Start with document

$$P(D,W) = \sum_{Z} P(Z)P(D|Z)P(W|Z)$$

- Start with topic

$$P(D,W) = \sum_{Z} \underline{P(Z)}\,\underline{P(D|Z)}\,\underline{P(W|Z)}$$

D = U Σ T*

https://medium.com/nanonets/topic-modeling-with-lsa-psla-lda-and-lda2vec-555ff65b0b05

Problems with pLSI

Although it looks quite different and approaches the problem in a very different way, pLSA really just adds a probabilistic treatment of topics and words on top of LSA. It is a far more flexible model, but still has a few problems. In particular:

- Because we have no parameters to model P(D), we don't know how to assign probabilities to new documents
- The number of parameters for pLSA grows linearly with the number of documents we have, so it is prone to overfitting
- We will not look at any code for pLSA because it is rarely used on its own. In general, when people are looking for a topic model beyond the baseline performance LSA gives, they turn to LDA. LDA, the most common type of topic model, extends pLSA to address these issues.

Latent Dirichlet Allocation (LDA)

- LDA is a Bayesian version of pLSA. In particular, it uses Dirichlet priors for the document-topic and word-topic distributions, lending itself to better generalization.

$$P(\boldsymbol{W}, \boldsymbol{Z}, \boldsymbol{\theta}, \boldsymbol{\varphi}; \alpha, \beta) = \prod_{i=1}^{K} P(\varphi_i; \beta) \prod_{j=1}^{M} P(\theta_j; \alpha) \prod_{t=1}^{N} P(Z_{j,t} \mid \theta_j) P(W_{j,t} \mid \varphi_{Z_{j,t}}),$$

1. Choose $\theta_i \sim \mathrm{Dir}(\alpha)$, where $i \in \{1, \ldots, M\}$ and $\mathrm{Dir}(\alpha)$ is a Dirichlet distribution with a symmetric parameter α which typically is sparse ($\alpha < 1$)
2. Choose $\varphi_k \sim \mathrm{Dir}(\beta)$, where $k \in \{1, \ldots, K\}$ and β typically is sparse
3. For each of the word positions i, j, where $i \in \{1, \ldots, M\}$, and $j \in \{1, \ldots, N_i\}$
   (a) Choose a topic $z_{i,j} \sim \mathrm{Multinomial}(\theta_i)$.
   (b) Choose a word $w_{i,j} \sim \mathrm{Multinomial}(\varphi_{z_{i,j}})$.

LDA results for the news titles: Topics (k=25)

0: 0.058*"number" + 0.048*"bus" + 0.036*"city" + 0.030*"hit" + 0.027*"water" + 0.022*"japan" + 0.021*"time" + 0.020*"talk" + 0.017*"world" + 0.016*”record"

1: 0.133*"war" + 0.051*"russia" + 0.039*"military" + 0.037*"russian" + 0.034*"troop" + 0.032*"force" + 0.025*"crime" + 0.025*"east" + 0.023*"saudi" + 0.018*"arm"

2: 0.065*"minister" + 0.047*"british" + 0.035*"britain" + 0.032*"prime" + 0.030*"wikileaks" + 0.027*"election" + 0.026*"party" + 0.021*"freedom" + 0.018*"warning" + 0.016*"take"

3: 0.047*"want" + 0.034*"dont" + 0.031*"school" + 0.023*"turn" + 0.021*"control" + 0.017*"die" + 0.017*"son" + 0.017*"release" + 0.015*"block" + 0.013*"europe"

4: 0.078*"state" + 0.035*"president" + 0.033*"nation" + 0.031*"united" + 0.030*"bomb" + 0.028*"security" + 0.026*"africa" + 0.025*"threat" + 0.018*"open" + 0.018*"national"

5: 0.095*"police" + 0.055*"protest" + 0.045*"dead" + 0.034*"man" + 0.031*"arrested" + 0.025*"shot" + 0.025*"mexico" + 0.023*"street" + 0.021*"officer" + 0.018*"violence"

6: 0.080*"woman" + 0.070*"child" + 0.032*"girl" + 0.029*"men" + 0.027*"family" + 0.024*"sex" + 0.023*"charge" + 0.020*"abuse" + 0.020*"accused" + 0.019*"trial"

7: 0.094*"gaza" + 0.055*"news" + 0.047*"video" + 0.045*"drug" + 0.027*"medium" + 0.023*"bbc" + 0.022*"tv" + 0.020*"site" + 0.016*"banned" + 0.016*"online"

8: 0.066*"right" + 0.051*"law" + 0.041*"human" + 0.026*"european" + 0.023*"torture" + 0.021*"eu" + 0.019*"vote" + 0.018*"gay" + 0.018*"demand" + 0.018*"parliament"

9: 0.065*"china" + 0.061*"nuclear" + 0.045*"world" + 0.034*"power" + 0.032*"india" + 0.016*"economy" + 0.016*"deal" + 0.015*"look" + 0.014*"zimbabwe" + 0.014*"building"

10: 0.034*"aid" + 0.030*"gaza" + 0.028*"food" + 0.028*"hamas" + 0.027*"head" + 0.024*"journalist" + 0.018*"away" + 0.018*"big" + 0.017*"doe" + 0.016*"legal"

…

LDA results for the news titles: Topics (k=25)

Topic #13

Topic #1

Topic #21

Topic #15

Topic #0

Topic #24

LDA results for the news titles: Documents by topics

- “North Korea has threatened to conduct a nuclear test in response to a United Nations move towards a probe into the country's human rights violations”
- ['north', 'korea', 'threatened', 'conduct', 'nuclear', 'test', 'response', 'united', 'nation', 'probe', 'country', 'human', 'right', 'violation']
- 1**5:** 0.75, **21:** 0.12, …

Topic #15

Topic #21

LDA results for the news titles: Documents by topics

- “North Korea has threatened to conduct a nuclear test in response to a United Nations move towards a probe into the country's human rights violations”
- ['north', 'korea', 'threatened', 'conduct', 'nuclear', 'test', 'response', 'united', 'nation', 'probe', 'country', 'human', 'right', 'violation']
- **8:** 0.23, **15:** 0.20, **21:** 0.15, **4:** 0.15, **9:** 0.14, …

Topic #8

Topic #15

Topic #21

Topic #4

Topic #9

LDA results for the news titles: Documents by topics

- “Argentine Court rules: Orang Utans are "non-human-persons" with human rights and therefore need to be released from zoo”
- ['argentine', 'court', 'rule', 'orang', 'utans', 'nonhumanpersons', 'human', 'right', 'need', 'released', ‘zoo']
- **8:** 0.50, **22:** 0.13, **23:** 0.12, **3:** 0.12, …

Topic #8

Topic #22

Topic #23

Topic #3

One more thing…

- State of the art for text data: Deep learning

https://openai.com/blog/better-language-models/

Deep learning for text data

Human-generated input

In a shocking finding, scientist discovered a herd of unicorns living in a remote, previously unexplored valley, in the Andes Mountains. Even more surprising to the researchers was the fact that the unicorns spoke perfect English.

Machine completion

https://openai.com/blog/better-language-models/

Deep learning for text data

Human-generated input

In a shocking finding, scientist discovered a herd of unicorns living in a remote, previously unexplored valley, in the Andes Mountains. Even more surprising to the researchers was the fact that the unicorns spoke perfect English.

Machine completion

The scientist named the population, after their distinctive horn, Ovid's Unicorn. These four-horned, silver-white unicorns were previously unknown to science.

Now, after almost two centuries, the mystery of what sparked this odd phenomenon is finally solved.

Dr. Jorge Pérez, an evolutionary biologist from the University of La Paz, and several companions, were exploring the Andes Mountains when they found a small valley, with no other animals or humans. Pérez noticed that the valley had what appeared to be a natural fountain, surrounded by two peaks of rock and silver snow.

Pérez and the others then ventured further into the valley. "By the time we reached the top of one peak, the water looked blue, with some crystals on top," said Pérez.

Pérez and his friends were astonished to see the unicorn herd…

https://openai.com/blog/better-language-models/

Exercise

- **Part 1 - Spam Classification.ipynb**
  Do Section 4

- **Part 2 - Stock prediction.ipynb [optional]**

Recommended reading

- "*Intuitive Guide to Latent Dirichlet Allocation" [optional]*
  https://towardsdatascience.com/light-on-math-machine-learning-intuitive-guide-to-latent-dirichlet-allocation-437c81220158

Who said that text mining is stressful?

I'm 27 and I feel great!