Topic Modeling: Theory and Implementation

Ch.15. Machine Learning for Algorithmic Trading. Stefan Jansen. 2020. Packt Publishing
Ch.6. Text Analytics with Python. Dipanjan Sarkar. 2019. Apress
Topic Modeling with LSA, PLSA, LDA & lda2Vec. Joyce Xu. 2018. Medium

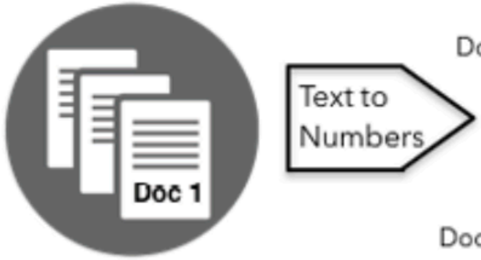
Document-Term Matrix

Bag-of-Words (BOW) the frequency of terms representing a document

Document-Term Matrix DTM) the frequency of terms in a <u>collection</u> of documents

<u>Useful</u> for comparing and classifying documents

High-dimensional



	Term 1	Term 2	...	Term n-1	Term n
Doc 1	0	1	1	1	0
Doc 2	0	1	0	0	0
Doc 3	2	0	3	0	0
⋮	.	.	.	.	.
Doc m-2	1	0	2	0	0
Doc m-1	0	0	1	0	0
Doc m	0	1	0	0	1

m Documents as vectors

Sparse

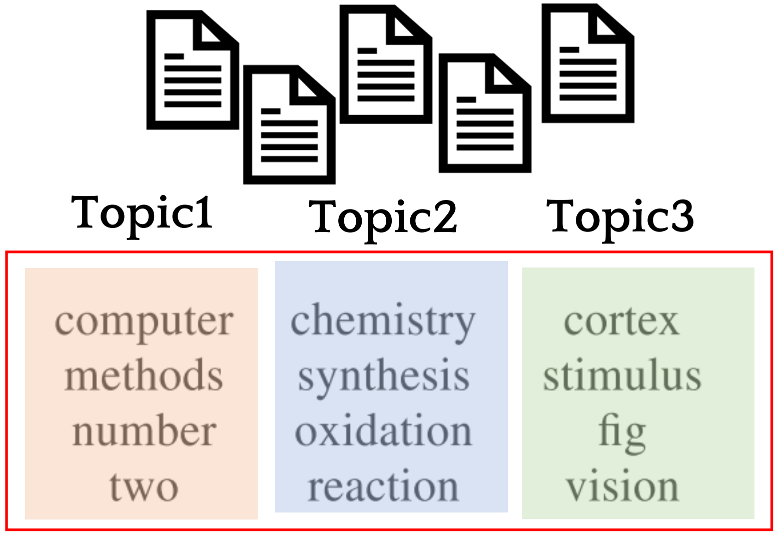
<u>Cannot capture</u> the latent variables/themes or provide document summary

Latent (hidden) variables = The semantics of the documents (meaning)

Source: Jansen, Stefan. 2020. Ch.14. Figure 14.3

Topic Modeling

The process of learning, recognizing, and extracting hidden topics across a collection of documents



Applications

Unsupervised discovery of insightful themes in customer reviews, contracts, news...

(Jansen, 2020; Xu, 2018.)

Topic Modeling Techniques

- Each document consists of a mixture of topics
- Each topic consists of a collection of words

LSI (LSA)	Latent Semantic Indexing
pLSA	probabilistic Latent Semantic Analysis
LDA	Latent Dirichlet Allocation
lda2vec	LDA+word2vec

Latent Semantic Analysis

the semantic document-term relationship by reducing word space dimensionality using SVD

TDM – Term-Document Matrix
DTM – Document-Term Matrix

m x n – a term-document matrix A

m – a term
n – a document

$A =$

	D1	D2	D3	D4
Brown	0.0	3.0	1.5	0.0
Cat	0.0	4.3	0.0	0.0
Coat	0.0	0.0	2.0	0.0

	D1	D2	D3	D4
brown	0.6	3.0	1.2	−0.7
cat	0.1	2.6	0.0	−1.4
coat	0.7	1.1	1.5	0.5

Rui Miguel Forte et al. 2017. R: Predictive Analysis. Ch. 11. Packt Publishing

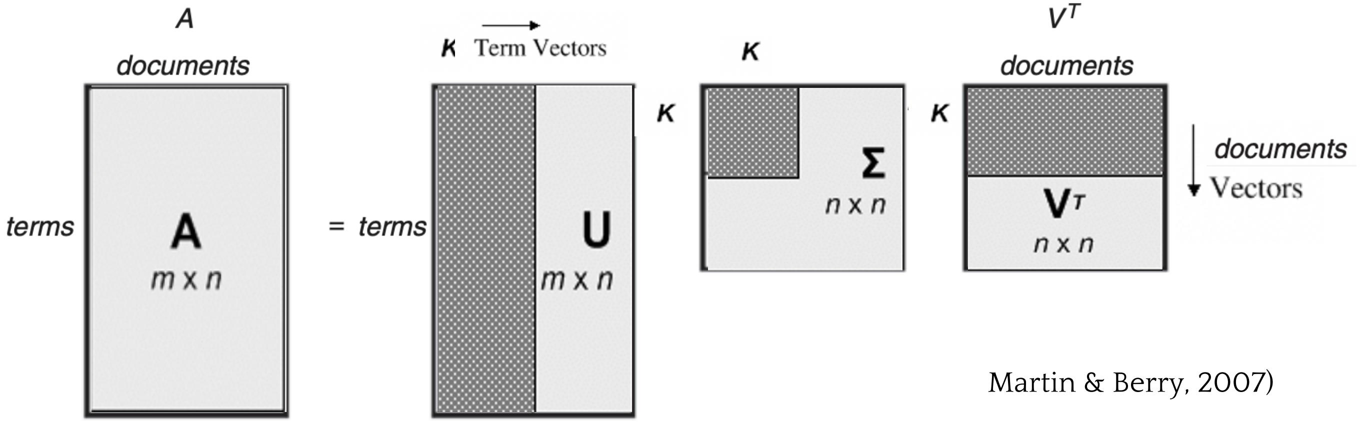
Single Value Decomposition (SVD)

- Find the <u>best approximation</u> of the original data points using <u>fewer dimensions</u>
- Identify and order the dimensions along which data points exhibit the most variation

$$A = USV^T$$

Truncated SVD

- only the k largest S values
- only k columns of U and V



Martin & Berry, 2007)

Latent Semantic Analysis I (BBC News)

Heading

```
Ad sales boost Time Warner profit

Quarterly profits at US media giant TimeWarne
months to December, from $639m year-earlier.

The firm, which is now one of the biggest inve
high-speed internet connections and higher adv
sales rose 2% to $11.1bn from $10.9bn. Its pro
offset a profit dip at Warner Bros, and less u
```

1. BBC dataset - 2,225 News articles (txt) in 5 categories:
   - business
   - entertainment
   - politics
   - sport
   - tech

```python
path = Path('bbc') # after you unzip bbc you should have a folder bbc
files = sorted(list(path.glob('**/*.txt'))) # sudirectories in the bbc folder
doc_list = []
for i, file in enumerate(files):
    with open(str(file), encoding='latin1') as f:
        topic = file.parts[-2] # parse path and extract the category name
        lines = f.readlines()
        heading = lines[0].strip()
        body = ' '.join([l.strip() for l in lines[1:]]) # exclude heading
        doc_list.append([topic.capitalize(), heading, body])
```

2. Create a dataframe with three columns: Category, Heading, Article

```python
docs = pd.DataFrame(doc_list, columns=['Category', 'Heading', 'Article'])
```

Category	Heading	Article
Business	Ad sales boost Time Warner profit	Quarterly profits at US media giant TimeWarne...
Business	Dollar gains on Greenspan speech	The dollar has hit its highest level against ...
Business	Yukos unit buyer faces loan claim	The owners of embattled Russian oil giant Yuk...
Business	High fuel prices hit BA's profits	British Airways has blamed high fuel prices f...
Business	Pernod takeover talk lifts Domecq	Shares in UK drinks and food firm Allied Dome...

sparse matrix: 2,175 x 2,917

3. Split and TF-IDF Vectorize

```python
train_docs, test_docs = train_test_split(docs,
                                         stratify=docs.Category,
                                         test_size=50,
                                         random_state=42)
```

```python
vectorizer = TfidfVectorizer(max_df=.25,
                             min_df=.01,
                             stop_words='english',
                             binary=False)
train_dtm = vectorizer.fit_transform(train_docs.Article)
```

Source: Jansen, Stefan. 2020. Ch.15.

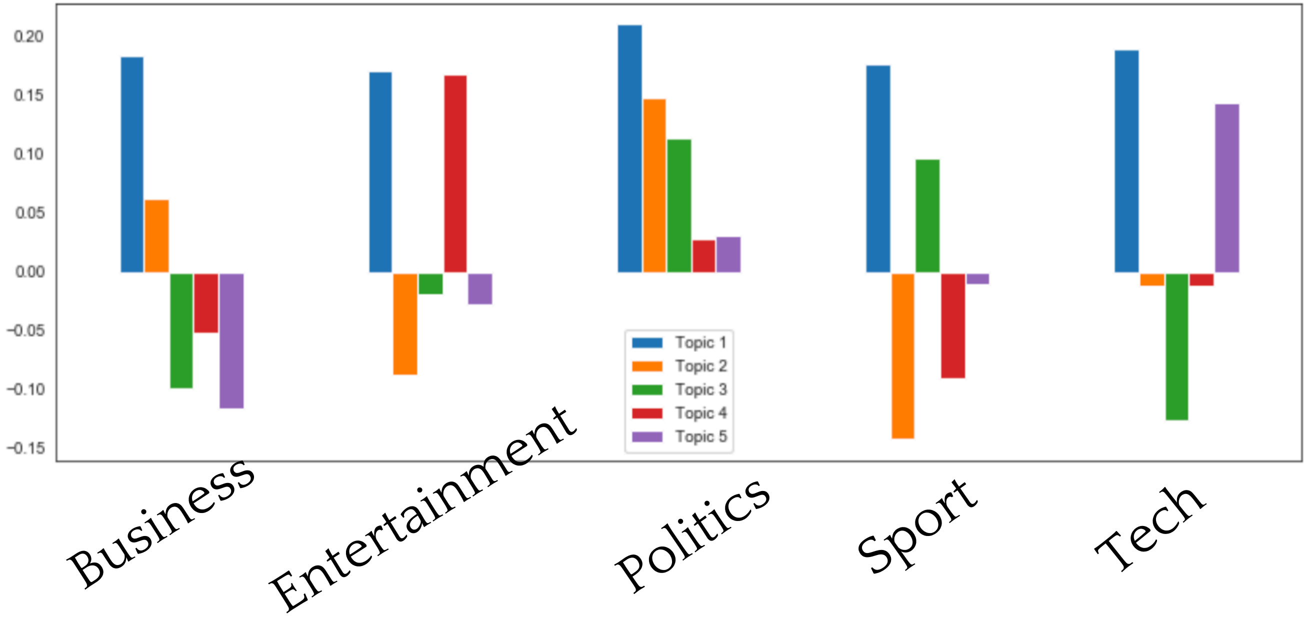
Latent Semantic Analysis II

4 SVD Model: sklearn.decomposition.TruncatedSVD

```
svd_model = TruncatedSVD(n_components=n_components, n_iter=5, random_state=42)
```

5 Pipeline

```
svd_transformer = Pipeline([('tfidf', vectorizer),
                            ('svd', svd_model)])
svd_matrix = svd_transformer.fit_transform(train_docs.Article)
```



The Average Topic Assignment per News category

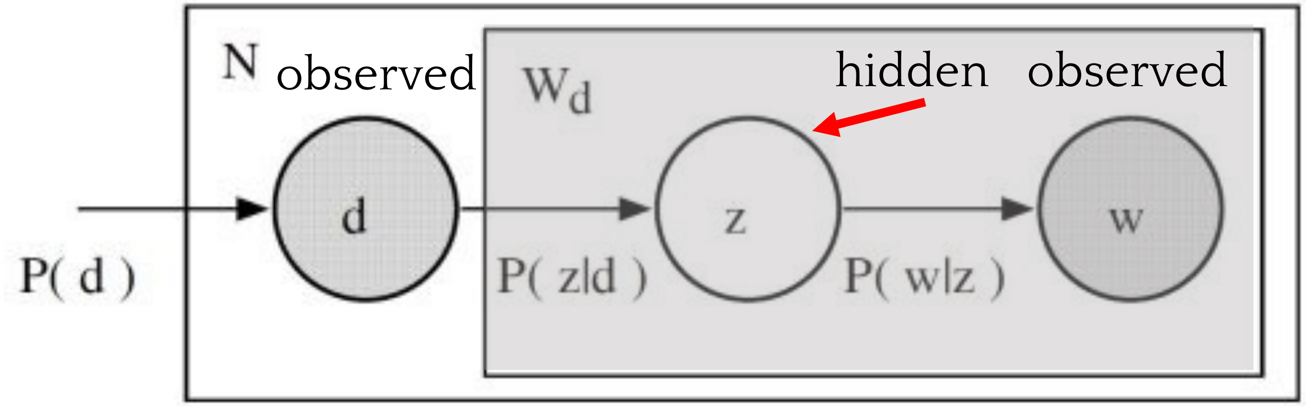
Top-10 words per Topic

Strength: Noise Removal, Semantics
Weakness: Interpretability, Evaluation

Source: Jansen, Stefan. 2020. Ch.15.

Probabilistic Latent Semantic Analysis (pLSA)

a probabilistic method instead of SVD: the probability for word w to appear in a document d



Documents Topics Words

Model Parameters

P(D)

- Probability of a document
- Determined from the corpus

P(z|d)

- Probability of a topic given a document

- Multinomial Distribution (EM)

P(w|z)

- Probability of a word given a topic

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

Strength: Models can be compared using the probabilities assigned to new documents

Source: Jansen, Stefan. 2020. Ch.15.

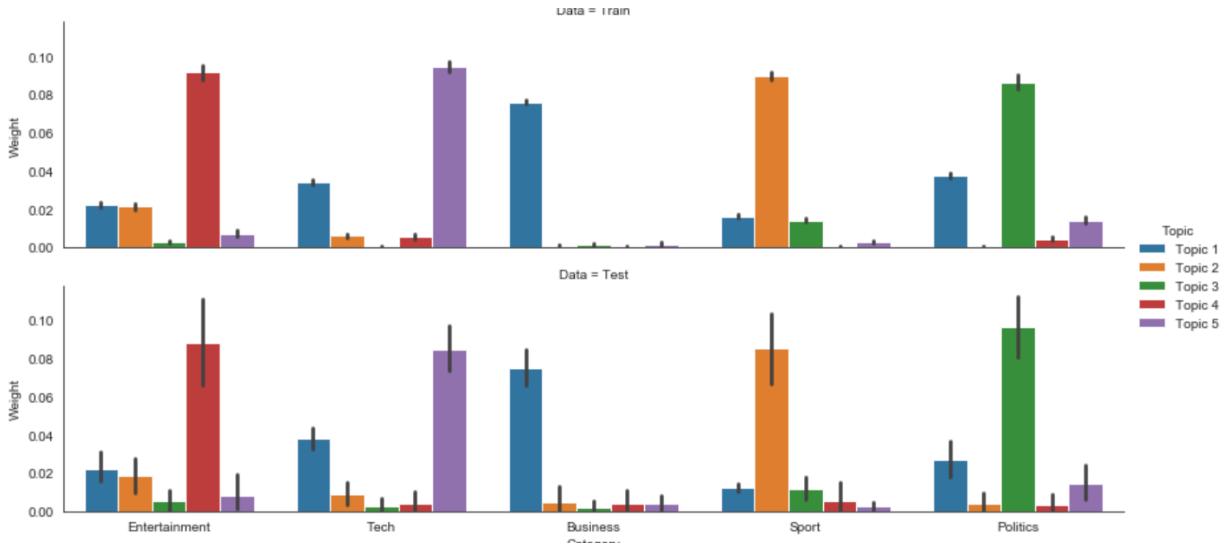
pLSA Implementation

1 Non-negative matrix factorization (NMF) model: sklearn.decomposition.NMF

```
nmf_model = NMF(n_components=n_components,
                random_state=42,
                solver='mu',
                beta_loss='kullback-leibler',
                max_iter=1000)
```

2 Pipeline

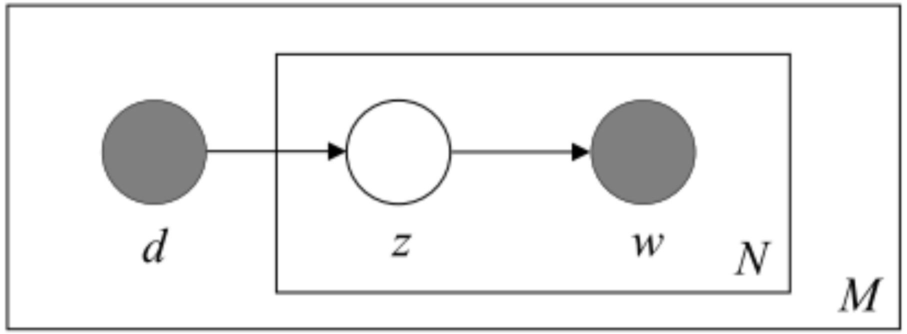
```
nmf_transformer = Pipeline([('tfidf', vectorizer),
                            ('nmf', nmf_model)])
nmf_matrix = nmf_transformer.fit_transform(train_docs.Article)
```



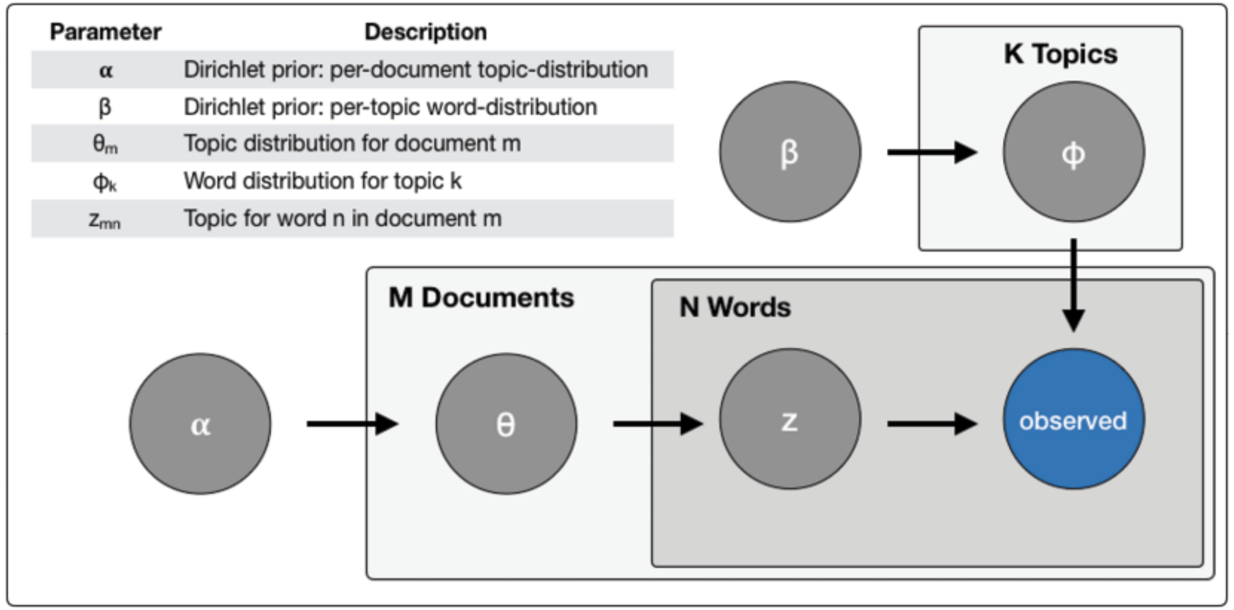
Source: Jansen, Stefan. 2020. Ch.15.

Latent Dirichlet Allocation (LDA) - (Blei, Ng, and Jordan 2003)

- Extends pLSA adding a generative process
  - Hierarchical Bayesian model:
    - topics are probability distribution over words
    - documents are probability distribution over topics
    - topics follow a sparse Dirichlet distribution
  - Can generalize to new documents
- Variants can include metadata (authors, image data)



pLSA: document>topic>word



LDA

Source: Jansen, Stefan. 2020. Ch.15.

LDA Implementation

1. Laten Dirichlet allocation model: decomposition.LatentDirichletAllocation

```
lda_model = LatentDirichletAllocation(n_components=n_components,
                                      n_jobs=-1,
                                      learning_method='batch',
                                      max_iter=10)
```

2. Pipeline

```
lda_transformer = Pipeline([('tfidf', vectorizer),
                            ('lda', lda_model)])
lda_matrix = lda_transformer.fit_transform(train_docs.Article)
```

3. Visualization: pyLDAvis