

CSE 190 – Lecture 13

Data Mining and Predictive Analytics

Text mining Part 2

Assignment 1... last update!

- A few details about the marking scheme
- One-hour extension

Recap: Prediction tasks involving text

What kind of quantities can we model, and what kind of prediction tasks can we solve using **text**?

Prediction tasks involving text

Does this article have a positive or negative sentiment about the subject being discussed?

What can stop US Postal Service trucks? The inexorable march of time

The ageing fleet of delivery vehicles is long past due an overhaul. Among the common-sense upgrades employees want: air conditioning and more workspace



Neither snow nor rain nor heat nor gloom of night stays these trucks - but time, it turns out, will. Photograph: Bill Sikes/AP

For the better part of the last 30 years, the flatulent buzz of the US Postal Service's boxy delivery vans - audible as they lighted from mailbox to mailbox - has been a familiar sound to most Americans. Neither snow nor rain nor heat nor gloom of night stays the USPS's mail trucks from the swift completion of their appointed

Feature vectors from text

Bag-of-Words models

The Peculiar Genius of Bjork

CULTURE | BY EMILY WITT | JANUARY 23, 2015 11:30 AM

Solo musician or master collaborator? For her new album, Bjork has merged the two sides of her artistry to create a new experience of music – again.



$F_{\text{text}} = [150, 0, 0, 0, 0, 0, \dots, 0]$

a

aardvark

zoetrope

musician, who creates her music in an emotional cocoon, tinkering with technologies, concepts and feelings; and Bjork the producer and curator, who seeks out



Feature vectors from text

Bag-of-Words models

Dark brown with a light tan head, minimal lace and low retention. Excellent aroma of dark fruit, plum, raisin and red grape with light vanilla, oak, caramel and toffee. Medium thick body with low carbonation. Flavor has strong brown sugar and molasses from the start over bready yeast and a dark fruit and plum finish. Minimal alcohol presence.

Actually, this is a nice quad.

yeast and minimal red body thick light a Flavor sugar strong quad. grape over is molasses lace the low and caramel fruit Minimal start and toffee. dark plum, dark brown Actually, alcohol Dark oak, nice vanilla, has brown of a with presence. light carbonation. bready from retention. with finish. with and this and plum and head, fruit, low a Excellent raisin aroma Medium tan

These two documents have **exactly** the same representation in this model, i.e., we're completely **ignoring** syntax. This is called a "bag-of-words" model.

Feature vectors from text

Q1: How many words are there?

```
wordCount = defaultdict(int)
for d in data:
    for w in d['review/text'].split():
        wordCount[w] += 1

print len(wordCount)
```

A: 150,009 (too many!)

Feature vectors from text

2: What if we remove capitalization/punctuation?

```
wordCount = defaultdict(int)
punctuation = set(string.punctuation)
for d in data:
    for w in d['review/text'].split():
        w = ''.join([c for c in w.lower() if not c in punctuation])
        wordCount[w] += 1

print len(wordCount)
```

A: 74,271 (still too many!)

Feature vectors from text

3: What if we merge different inflections of words?

drinks → drink
drinking → drink
drinker → drink

argue → argu
arguing → argu
argues → argu
arguing → argu
argus → argu

Feature vectors from text

3: What if we merge different inflections of words?

```
wordCount = defaultdict(int)
punctuation = set(string.punctuation)
stemmer = nltk.stem.porter.PorterStemmer()
for d in data:
    for w in d['review/text'].split():
        w = ''.join([c for c in w.lower() if not c in punctuation])
        w = stemmer.stem(w)
        wordCount[w] += 1

print len(wordCount)
```

A: 59,531 (still too many...)

Feature vectors from text

4: Just discard extremely rare words...

```
counts = [(wordCount[w], w) for w in wordCount]
counts.sort()
counts.reverse()

words = [x[1] for x in counts[:1000]]
```

- Pretty unsatisfying but at least we can get to some inference now!

Feature vectors from text

Removing stopwords:

```
from nltk.corpus import stopwords  
stopwords.words("english")
```

```
['i', 'me', 'my', 'myself', 'we', 'our', 'ours', 'ourselves', 'you',  
'your', 'yours', 'yourself', 'yourselves', 'he', 'him', 'his', 'himself',  
'she', 'her', 'hers', 'herself', 'it', 'its', 'itself', 'they', 'them',  
'their', 'theirs', 'themselves', 'what', 'which', 'who', 'whom', 'this',  
'that', 'these', 'those', 'am', 'is', 'are', 'was', 'were', 'be', 'been',  
'being', 'have', 'has', 'had', 'having', 'do', 'does', 'did', 'doing',  
'a', 'an', 'the', 'and', 'but', 'if', 'or', 'because', 'as', 'until',  
'while', 'of', 'at', 'by', 'for', 'with', 'about', 'against', 'between',  
'into', 'through', 'during', 'before', 'after', 'above', 'below', 'to',  
'from', 'up', 'down', 'in', 'out', 'on', 'off', 'over', 'under', 'again',  
'further', 'then', 'once', 'here', 'there', 'when', 'where', 'why',  
'how', 'all', 'any', 'both', 'each', 'few', 'more', 'most', 'other',  
'some', 'such', 'no', 'nor', 'not', 'only', 'own', 'same', 'so', 'than',  
'too', 'very', 's', 't', 'can', 'will', 'just', 'don', 'should', 'now']
```

Feature vectors from text

We can build a richer predictor by using **n-grams**

e.g. "Medium thick body with low carbonation."

unigrams: ["medium", "thick", "body", "with", "low", "carbonation"]

bigrams: ["medium thick", "thick body", "body with", "with low", "low carbonation"]

trigrams: ["medium thick body", "thick body with", "body with low", "with low carbonation"]

etc.

Feature vectors from text

Let's do some inference!

Problem 1: Sentiment analysis

Let's build a predictor of the form:

$$f(\text{text}) \rightarrow \text{rating}$$

using a model based on linear regression:

$$\text{rating} \simeq \alpha + \sum_{w \in \text{text}} \text{count}(w) \cdot \theta_w$$

Feature vectors from text

What do the parameters look like?

$$\theta_{\text{fantastic}} = 0.143$$

$$\theta_{\text{watery}} = -0.163$$

$$\theta_{\text{and}} = -0.008$$

$$\theta_{\text{me}} = -0.037$$

CSE 190 – Lecture 12

Data Mining and Predictive Analytics

TF-IDF

Finding relevant terms

So far we've dealt with huge vocabularies just by identifying the **most frequently occurring** words

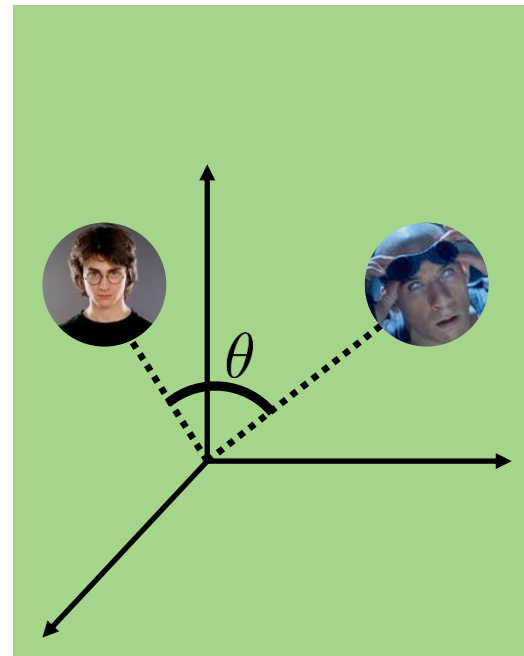
But! The most informative words may be those that occur very rarely, e.g.:

- Proper nouns (e.g. people's names) may predict the content of an article even though they show up rarely
- Extremely superlative (or extremely negative) language may appear rarely but be very predictive

Finding relevant terms

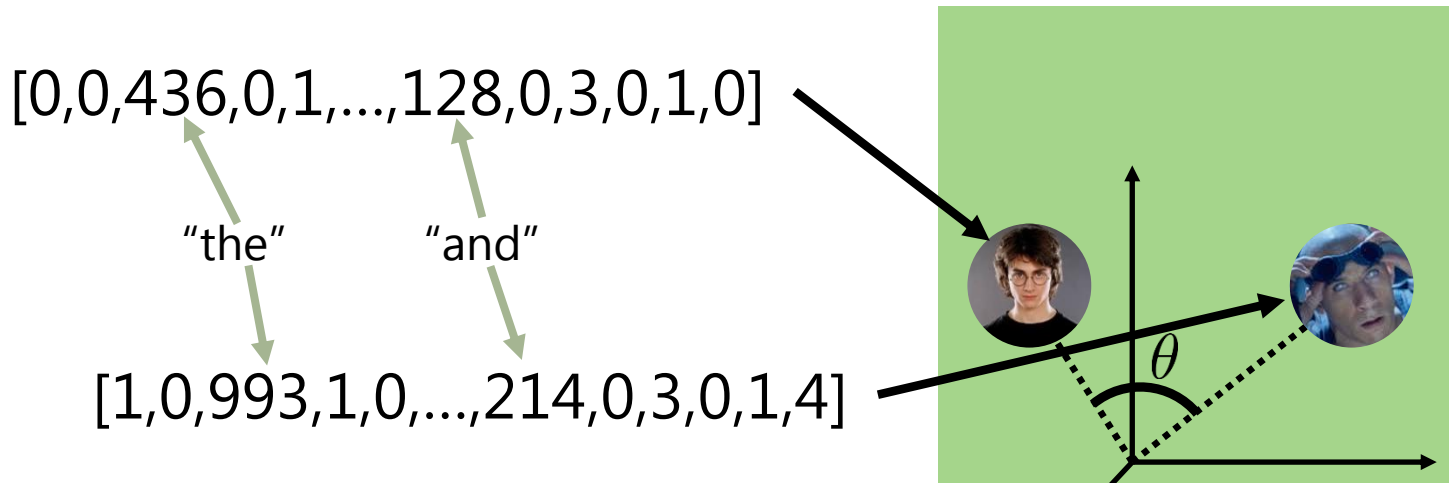
e.g. imagine applying something like cosine similarity to the document representations we've seen so far

e.g. are (the features of the reviews/IMDB descriptions of) these two documents "similar", i.e., do they have high cosine similarity



Finding relevant terms

e.g. imagine applying something like cosine similarity to the document representations we've seen so far



The similarity is primarily determined by the frequency of unimportant words. How can we address this?

Finding relevant terms

So how can we estimate the
“relevance” of a word in a document?

e.g. which words in this document might help us to determine its content, or to find similar documents?

Despite Taylor making moves to end her long-standing feud with Katy, HollywoodLife.com has learned exclusively that Katy isn't ready to let things go! Looks like the bad blood between Kat Perry, 29, and Taylor Swift, 25, is going to continue brewing. A source tells HollywoodLife.com exclusively that Katy prefers that their frenemy battle lines remain drawn, and we've got all the scoop on why Katy is set in her ways. Will these two *ever* bury the hatchet? Katy Perry & Taylor Swift Still Fighting? "Taylor's tried to reach out to make amends with Katy, but Katy is not going to accept it nor is she interested in having a friendship with Taylor," a source tells HollywoodLife.com exclusively. "She wants nothing to do with Taylor. In Katy's mind, Taylor shouldn't even attempt to make a friendship happen. That ship has sailed." While we love that Taylor has tried to end the feud, we can understand where Katy is coming from. If a friendship would ultimately never work, then why bother? These two have taken their feud everywhere from social media to magazines to the Super Bowl. Taylor's managed to mend the fences with Katy's BFF Diplo, but it looks like Taylor and Katy won't be posing for pics together in the near future. Katy Perry & Taylor Swift: Their Drama Hits All-Time High At the very least Katy and Taylor could tone down their feud That's not too much to ask

Finding relevant terms

So how can we estimate the “relevance” of a word in a document?

e.g. which words in this document might help us to determine its content, or to find similar documents?

Despite Taylor making moves to end her long-standing feud with Katy, HollywoodLife.com has learned exclusively that Katy isn't ready to let things go! Looks like **the** bad blood between Kat Perry, 29, and Taylor Swift, 25, is going to continue brewing. A source tells HollywoodLife.com exclusively that Katy prefers that their frenemy battle lines remain drawn, and we've got all **the** scoop on why Katy **the** will these two ever bury **the** hatchet? Katy Perry & Taylor Swift Still Fighting? "Taylor Swift tried to make amends with Katy, but Katy is not going to accept it nor is she interested in a friendship with Taylor," a source tells HollywoodLife.com exclusively. "She wanted Taylor to stay away from Taylor. In Katy's mind, Taylor shouldn't even attempt to make a friendship happen. That ship has sailed." While we love that Taylor has tried to end **the** feud, we can understand where Katy is coming from. If a friendship would ultimately never work, then why bother? These two have taken their feud everywhere from social media to magazines to **the** Super Bowl. Taylor's managed to mend **the** fences with Katy's BFF Diplo, but it looks like Taylor and Katy won't be posing for pics together in **the** near future. Katy Perry & Taylor Swift: Their Drama Hits All-Time High At **the** very least Katy and Taylor could tone down their feud That's not too much to ask

"the" appears
12 times in the
document

Finding relevant terms

So how can we estimate the “relevance” of a word in a document?

Q: The document discusses “the” more than it discusses “Taylor Swift”, so how might we come to the conclusion that “Taylor Swift” is the more relevant expression?

A: It discusses “the” **no more** than other documents do, but it discusses “Taylor Swift” **much more**

Finding relevant terms


Term frequency & document frequency

“Term frequency”: $tf(t, d)$ = number of times the term t appears in the document d

e.g. $tf(\text{“Taylor Swift”, that news article}) = 3$

“Inverse document frequency”: $idf(t, D) = \log \frac{N}{|\{d \in D : t \in d\}|}$

term (e.g.
“Taylor Swift”) set of
documents



“Justification”: $P(t|D) = \frac{|\{d \in D : t \in d\}|}{N}$ so $idf(t, D) = -\log P(t|D)$

Finding relevant terms

Term frequency & document frequency

Term frequency ~ How much does the term appear in the document

Inverse document frequency ~ How "rare" is this term across all documents

$$tfidf(t, d, D) = tf(t, d) \times idf(t, D)$$

Finding relevant terms

Term frequency & document frequency

TF-IDF is high \rightarrow this word appears much more frequently in this document compared to other documents

TF-IDF is low \rightarrow this word appears infrequently in this document, or it appears in many documents

$$tfidf(t, d, D) = tf(t, d) \times idf(t, D)$$

Finding relevant terms

Term frequency & document frequency

tf is sometimes defined differently, e.g.:

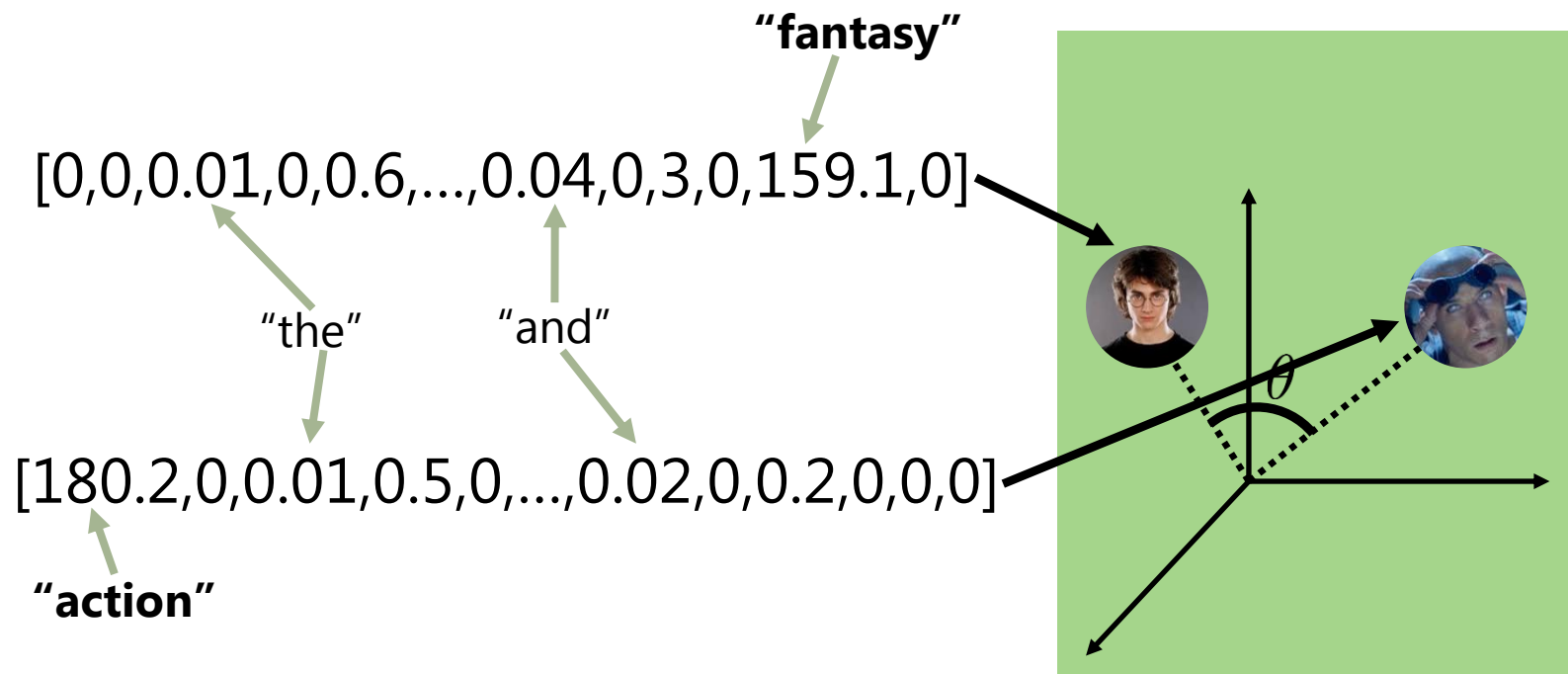
$$tf'(t, d) = \delta(t \in d)$$

$$tf''(t, d) = \frac{\text{frequency of word}}{\text{frequency of most common word in document}}$$

Both of these representations are invariant to the document length, compared to the regular definition which assigns higher weights to longer documents

Finding relevant terms

How to use TF-IDF



- Frequently occurring words have little impact on the similarity
- The similarity is now determined by the words that are most "characteristic" of the document

Finding relevant terms

But what about when we're **weighting** the parameters anyway?

e.g. is:

$$\text{rating} \simeq \alpha + \sum_{w \in \text{text}} \text{count}(w) \cdot \theta_w$$

really any different from:

$$\text{rating} \simeq \alpha + \sum_{w \in \text{text}} \text{tfidf}(w, d, D) \cdot \theta_w$$

after we fit parameters?

Finding relevant terms

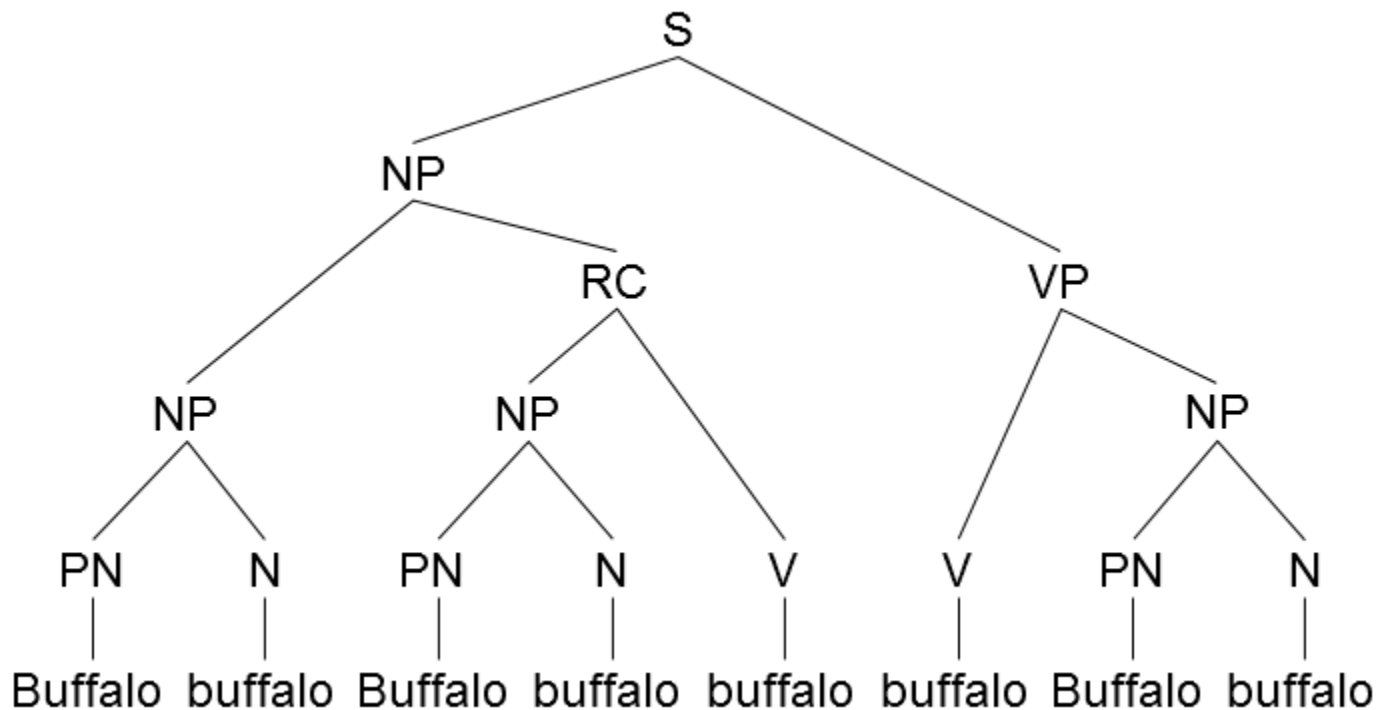
But what about when we're **weighting** the parameters anyway?

Yes!

- The **relative** weights of features is different between documents, so the two representations are not the same (up to scale)
- When we regularize, the scale of the features matters – if some “unimportant” features are very large, then the model can overfit on them “for free”

Etc.

Not today...



See Michael Collins & Regina Barzilay's NLP mooc if you're interested:

<http://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-864-advanced-natural-language-processing-fall-2005/index.htm>

Questions?

Further reading:

- Original TF-IDF paper (from 1972)

"A Statistical Interpretation of Term Specificity and Its Application in Retrieval"

<http://goo.gl/1CLwUV>

CSE 190 – Lecture 13

Data Mining and Predictive Analytics

Dimensionality-reduction approaches to document representation

Dimensionality reduction

How can we find **low-dimensional structure** in documents?

What we would like:

87 of 102 people found the following review helpful

★★★★★ **You keep what you kill**, December 27, 2004

By [Schtinky "Schtinky"](#) (Washington State) - [See all my reviews](#)
VINE™ VOICE

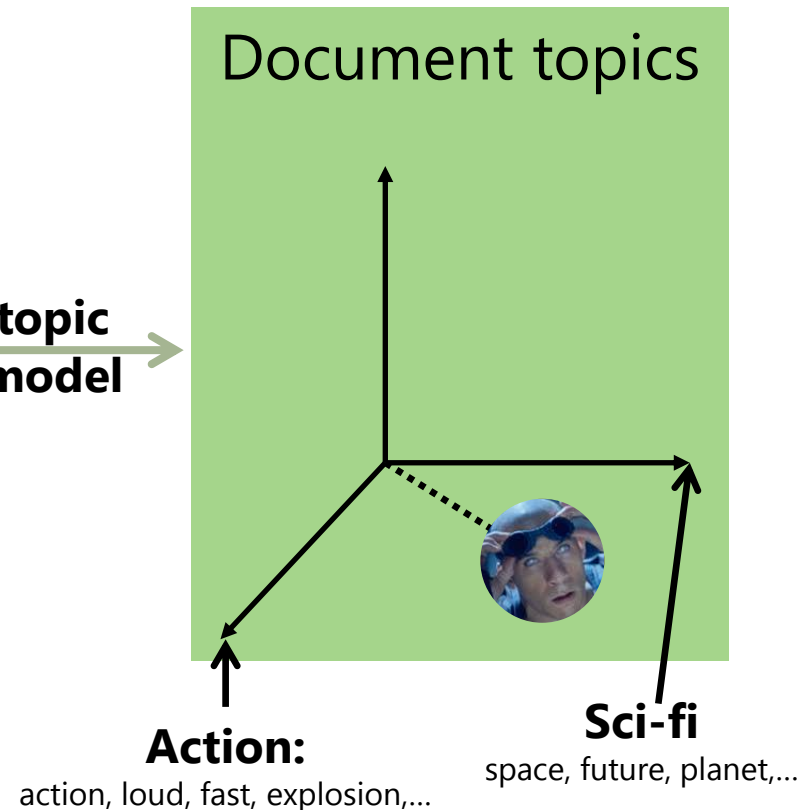
This review is from: [The Chronicles of Riddick \(Widescreen Unrated Director's Cut\) \(DVD\)](#)

Even if I have to apologize to my Friends and Favorites, and my family, I have to admit that I really liked this movie. It's a Sci-Fi movie with a "Mad Maxx" appeal that, while changing many things, left Riddick from 'Pitch Black' to be just Riddick. They did not change his attitude or soften him up or bring him out of his original character, which was very pleasing to 'Pitch Black' fans like myself.

First off, let me say that when playing the DVD, the first selection to come up is Convert or Fight, and no explanation of the choices. This confused me at first, so I will mention off the bat that they are simply different menu formats, that each menu has the very same options, simply different background visuals. Select either one and continue with the movie.

(review of "The Chronicles of Riddick")

topic
model →



A (very quick) case study

(I know it's not that part of the lecture yet)

How can we estimate which words in a review refer to which sensory aspects?

'Partridge in a Pear Tree', brewed by 'The Bruery'

Dark brown with a light tan head, minimal lace and low retention. Excellent aroma of dark fruit, plum, raisin and red grape with light vanilla, oak, caramel and toffee. Medium thick body with low carbonation. Flavor has strong brown sugar and molasses from the start over bread yeast and a dark fruit and plum finish. Minimal alcohol presence. Actually, this is a nice quad.

Feel: 4.5 Look: 4 Smell: 4.5 Taste: 4 Overall: 4

Aspects of opinions

There are lots of settings in which people's opinions cover many dimensions:

Wikipedia pages:

Rate this page
[What's this?](#)

Trustworthy Objective Complete Well-written

★★★★★ ★ ★★★★★★ ★ ★★★★★★ ★ ★★★★★★ ★

Cigars:

Criteria	1	2	3	4	5	6	7	8	9	10
Appearance	☺	☺	☺	☺	☺	☺	☺	☺	☺	☺
Construction	☺	☺	☺	☺	☺	☺	☺	☺	☺	☺
Flavor	☺	☺	☺	☺	☺	☺	☺	☺	☺	☺
Value	☺	☺	☺	☺	☺	☺	☺	☺	☺	☺
Overall Experience	☺	☺	☺	☺	☺	☺	☺	☺	☺	☺

Beers:

jtierney89
New Jersey

3.65/5 rDev -3.7%

look: 3.5 | smell: 3.5 | taste: 3.5 | feel: 4 | overall: 4

Very very deep brown near black, two fingers of of tan head. faint notes of chili lime and coconut.

Audiobooks:

 **André**
ORLANDO, FL, United States
10-11-13

Overall ★★★★★

Performance ★★★★★

Story ★★★★★

Hotels:

Rating summary

Sleep Quality ○○○○○

Location ○○○○○

Rooms ○○○○○

Service ○○○○○

Value ○○○○○

Cleanliness ○○○○○

Aspects of opinions

Further reading on this problem:

- Brody & Elhadad
"An unsupervised aspect-sentiment model for online reviews"
- Gupta, Di Fabbrizio, & Haffner
"Capturing the stars: predicting ratings for service and product reviews"
- Ganu, Elhadad, & Marian
"Beyond the stars: Improving rating predictions using review text content"
- Lu, Ott, Cardie, & Tsou
"Multi-aspect sentiment analysis with topic models"
- Rao & Ravichandran
"Semi-supervised polarity lexicon induction"
- Titov & McDonald
"A joint model of text and aspect ratings for sentiment summarization"

Aspects of opinions

If we can uncover these dimensions, we might be able to:

- Build sentiment models for each of the different aspects
- Summarize opinions according to each of the sensory aspects
- Predict the multiple dimensions of ratings from the text alone
- But also: **understand** the types of positive and negative language that people use

Aspects of opinions

Task: given (multidimensional) ratings and plain-text reviews, predict which sentences in the review refer to which aspect

Input:

'Partridge in a Pear Tree', brewed by 'The Bruery'

Dark brown with a light tan head, minimal lace and low retention. Excellent aroma of dark fruit, plum, raisin and red grape with light vanilla, oak, caramel and toffee.

Medium thick body with low carbonation. Flavor has strong brown sugar and molasses from the start over bread yeast and a dark fruit and plum finish. Minimal alcohol presence. Actually, this is a nice quad.

Feel: 4.5 Look: 4 Smell: 4.5 Taste: 4 Overall: 4

Output:

'Partridge in a Pear Tree', brewed by 'The Bruery'

Dark brown with a light tan head, minimal lace and low retention. Excellent aroma of dark fruit, plum, raisin and red grape with light vanilla, oak, caramel and toffee.

Medium thick body with low carbonation. Flavor has strong brown sugar and molasses from the start over bread yeast and a dark fruit and plum finish. Minimal alcohol presence. Actually, this is a nice quad.

Feel: 4.5 Look: 4 Smell: 4.5 Taste: 4 Overall: 4

Aspects of opinions

Solving this problem depends on solving the following two sub-problems:

1. Labeling the sentences is **easy** if we have a good model of the words used to describe each aspect
 2. Building a model of the different aspects is **easy** if we have labels for each sentence
- **Challenge:** each of these subproblems depends on having a good solution to the other one
 - So (as usual) start the model somewhere and alternately solve the subproblems until convergence

Aspects of opinions

Model:

$$P(\text{aspect}(s) = k | \text{sentence } s, \text{rating } v) =$$

$$\frac{1}{Z} \exp \sum_{w \in s} \left\{ \underbrace{\theta_{k,w}}_{\text{aspect weights}} + \underbrace{\phi_{k,v_k,w}}_{\text{sentiment weights}} \right\}$$

normalization
over all aspects

Sum over words
in the sentence

Weight for a word
(w) appearing in a
particular aspect (k)

Weight for a word
(w) appearing in a
particular aspect
(k), when the rating
is v_k

Aspects of opinions

Intuition:

$$P(\text{aspect}(s) = k | \text{sentence } s, \text{rating } v) =$$

$$\frac{1}{Z} \exp \sum_{w \in s} \left\{ \underbrace{\theta_{k,w}}_{\text{aspect weights}} + \underbrace{\phi_{k,v_k,w}}_{\text{sentiment weights}} \right\}$$

Nouns should have high weights, since they describe an aspect but are independent of the sentiment

Adjectives should have high weights, since they describe specific sentiments

Aspects of opinions

Procedure:

1. Given the current model (θ and ϕ), choose the most likely aspect labels for each sentence

$$\max_{\text{aspect labels for each sentence}} P_{\theta, \phi}(\text{aspect}(s) = k | \text{sentence } s, \text{ rating } v)$$

2. Given the current aspect labels, estimate the parameters θ and ϕ (convex problem)

$$\max_{\theta, \phi} P_{\theta, \phi}(\text{aspect}(s) = k | \text{sentence } s, \text{ rating } v)$$

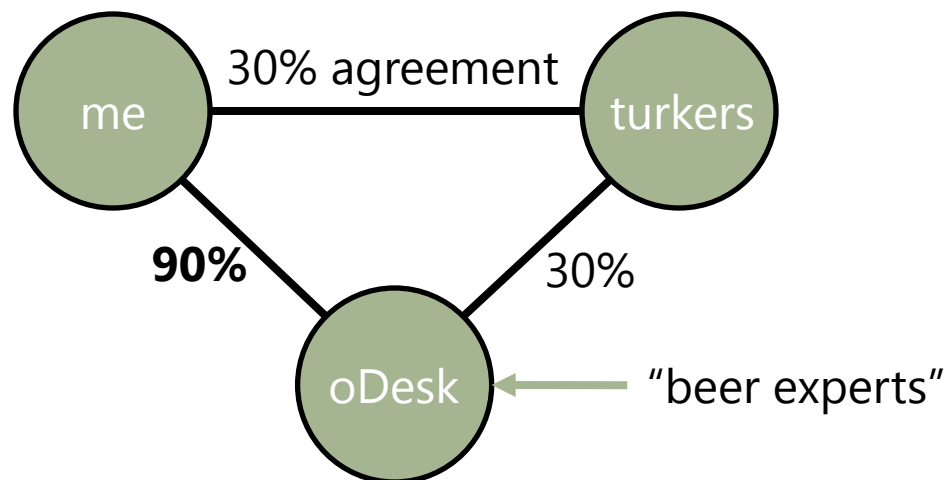
3. Iterate until convergence (i.e., until aspect labels don't change)

Aspects of opinions

Evaluation:

In order to tell if this is working, we need to get some humans to label some sentences

- I labeled 100 sentences for validation, and sent 10,000 sentences to Amazon's "mechanical turk"
 - These were next-to-useless
- So we hired some "experts" to label beer sentences



Aspects of opinions

Evaluation:

- 70-80% accurate at labeling beer sentences (somewhat less accurate for other review datasets)
- A few other tasks too, e.g. summarization (selecting sentences that describe different opinions on a particular aspect), and missing rating completion

Aspects of opinions

Moral of the story:

- We can obtain fairly accurate results just using a bag-of-words approach
- People use very different language if they have positive vs. negative opinions
- In particular, people don't just take positive language and negate it, so modeling syntax (presumably?) wouldn't help that much

Questions?

Further reading:

- Linguistics of food

"The language of Food: A Linguist Reads the Menu"

<http://www.amazon.com/The-Language-Food-Linguist-Reads/dp/0393240835>

CSE 190 – Lecture 13

Data Mining and Predictive Analytics

Dimensionality-reduction approaches to document representation – part 2

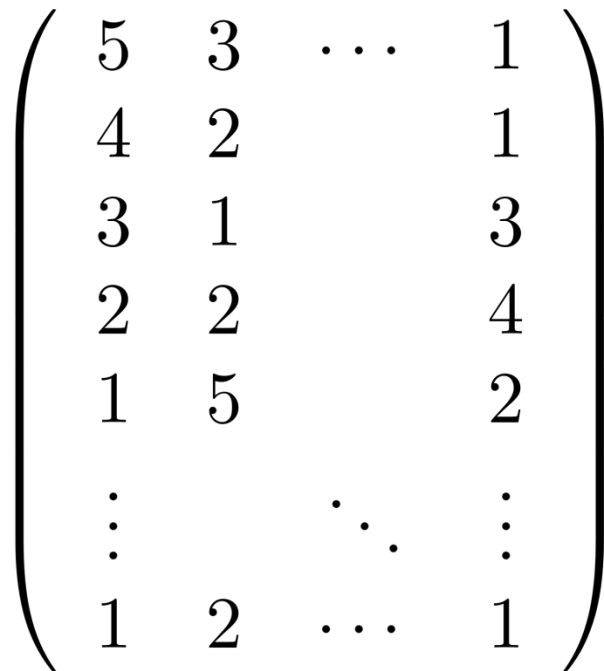
Dimensionality reduction approaches to text

In the case study we just saw, the dimensions were **given** to us – we just had to find the topics corresponding to them

What can we do to find the dimensions **automatically**?

Singular-value decomposition

Recall (from weeks 3&5)

$R =$ 

$$\begin{pmatrix} 5 & 3 & \cdots & 1 \\ 4 & 2 & & 1 \\ 3 & 1 & & 3 \\ 2 & 2 & & 4 \\ 1 & 5 & & 2 \\ \vdots & & \ddots & \vdots \\ 1 & 2 & \cdots & 1 \end{pmatrix}$$

(e.g.)
matrix of
ratings

(square roots of)
eigenvalues of RR^T

$$R = U \Sigma V^T$$

eigenvectors of RR^T

eigenvectors of $R^T R$

Singular-value decomposition

Taking the eigenvectors corresponding to the top-K eigenvalues is then the “best” rank-K approximation

$$R = \begin{pmatrix} 5 & 3 & \cdots & 1 \\ 4 & 2 & & 1 \\ 3 & 1 & & 3 \\ 2 & 2 & & 4 \\ 1 & 5 & & 2 \\ \vdots & & \ddots & \vdots \\ 1 & 2 & \cdots & 1 \end{pmatrix}$$

(square roots of top k)
eigenvalues of RR^T

$$R \simeq U^{(k)} \Sigma^{(k)} V^{(k)T}$$

(top k) eigenvectors of RR^T

(top k) eigenvectors of $R^T R$

Singular-value decomposition

What happens when we apply this to a matrix encoding our documents?

$$X = \begin{pmatrix} 1 & 0 & \dots & 4 \\ 0 & 2 & & 0 \\ 31 & 23 & & 97 \\ 0 & 98 & & 1 \\ 473 & 88 & & 347 \\ \vdots & & \ddots & \vdots \\ 11 & 34 & \dots & 13 \end{pmatrix}$$

document matrix

documents

terms

X is a $T \times D$ matrix whose **columns** are bag-of-words representations of our documents

T = dictionary size
 D = number of documents

Singular-value decomposition

What happens when we apply this to a matrix encoding our documents?

$X^T X$ is a $D \times D$ matrix.

$U^{(k)} \Sigma^{(k)}$ is a low-rank approximation of each **document**

 eigenvectors of $X^T X$

$X X^T$ is a $T \times T$ matrix.

$V^{(k)} \Sigma^{(k)}$ is a low-rank approximation of each **term**

 eigenvectors of $X X^T$

Singular-value decomposition

Using our low rank representation of each **document** we can...

- Compare two documents by their low dimensional representations (e.g. by cosine similarity)
- To retrieve a document (by first projecting the query into the low-dimensional document space)
- Cluster similar documents according to their low-dimensional representations
- Use the low-dimensional representation as features for some other prediction task

Singular-value decomposition

Using our low rank representation of each **word** we can...

- Identify potential synonyms – if two words have similar low-dimensional representations then they should have similar “roles” in documents and are potentially synonyms of each other
- This idea can even be applied across languages, where similar terms in different languages ought to have similar representations in parallel corpora of translated documents

Singular-value decomposition

This approach is called **latent semantic analysis**

- In practice, computing eigenvectors for matrices of the sizes in question is not practical – neither for XX^T nor X^TX (they won't even fit in memory!)
- Instead one needs to resort to some approximation of the SVD, e.g. a method based on stochastic gradient descent that never requires us to compute XX^T or X^TX directly (much as we did when approximating rating matrices with low-rank terms)

Probabilistic modeling of documents

Finally, can we represent documents in terms of the topics they describe?

What we would like:

87 of 102 people found the following review helpful

★★★★★ **You keep what you kill**, December 27, 2004

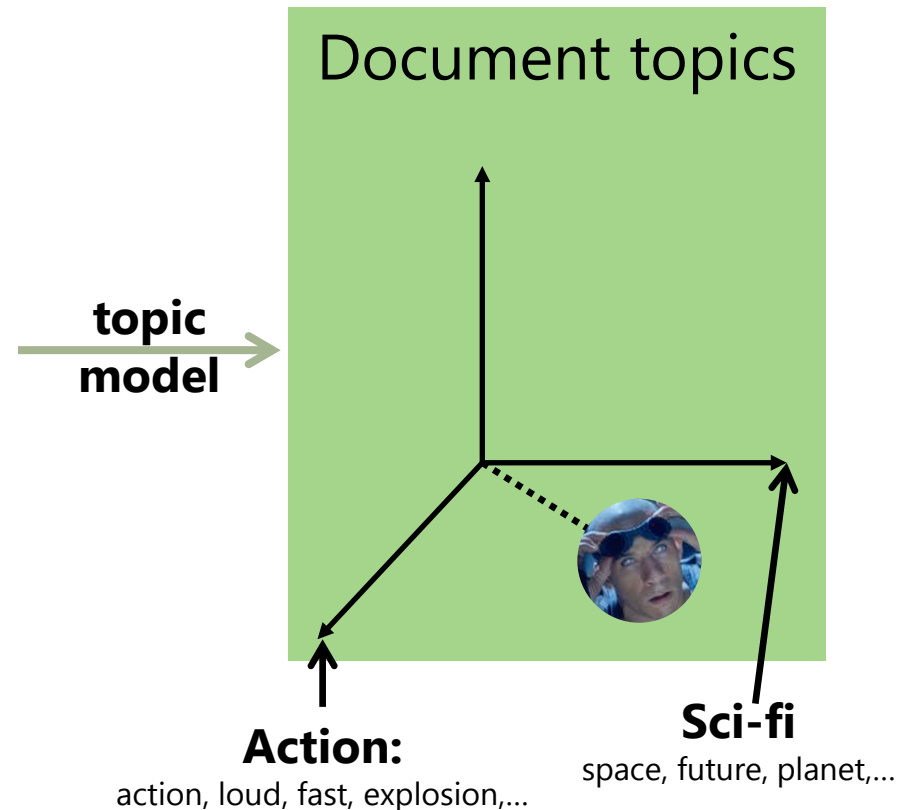
By [Schtinky "Schtinky"](#) (Washington State) - [See all my reviews](#)
VINE™ VOICE

This review is from: [The Chronicles of Riddick \(Widescreen Unrated Director's Cut\) \(DVD\)](#)

Even if I have to apologize to my Friends and Favorites, and my family, I have to admit that I really liked this movie. It's a Sci-Fi movie with a "Mad Maxx" appeal that, while changing many things, left Riddick from 'Pitch Black' to be just Riddick. They did not change his attitude or soften him up or bring him out of his original character, which was very pleasing to 'Pitch Black' fans like myself.

First off, let me say that when playing the DVD, the first selection to come up is Convert or Fight, and no explanation of the choices. This confused me at first, so I will mention off the bat that they are simply different menu formats, that each menu has the very same options, simply different background visuals. Select either one and continue with the movie.

(review of "The Chronicles of Riddick")



Probabilistic modeling of documents

Finally, can we represent documents in terms of the topics they describe?

- We'd like each document to be a **mixture over topics** (e.g. if movies have topics like "action", "comedy", "sci-fi", and "romance", then reviews of action/sci-fis might have representations like [0.5, 0, 0.5, 0])

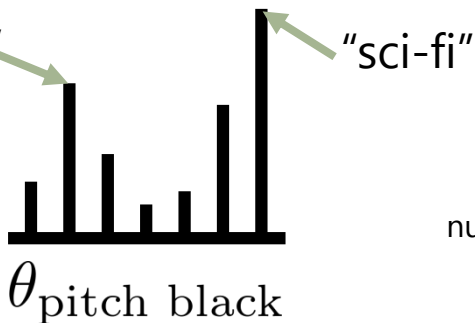
↑ ↑
action sci-fi

- Next we'd like each topic to be a **mixture over words** (e.g. a topic like "action" would have high weights for words like "fast", "loud", "explosion" and low weights for words like "funny", "romance", and "family")

Latent Dirichlet Allocation

Both of these can be represented by **multinomial distributions**

"action"

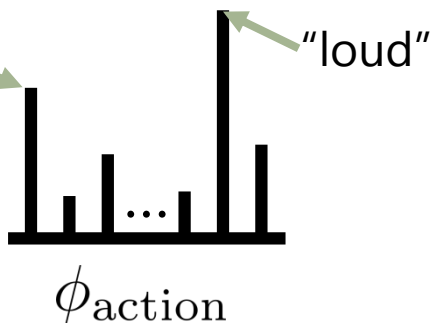


Each document has a **topic distribution** which is a mixture over the topics it discusses

number of topics

$$\theta_d \in \Delta^K \text{ i.e., } \forall_d \sum_k \theta_{d,k} = 1$$

"fast"



Each topic has a **word distribution** which is a mixture over the words it discusses

number of words

$$\phi_k \in \Delta^D \text{ i.e., } \forall_k \sum_w \phi_{k,w} = 1$$

Latent Dirichlet Allocation

LDA assumes the following “process” that generates the words in a document

(suppose we already know the topic distributions and word distributions)

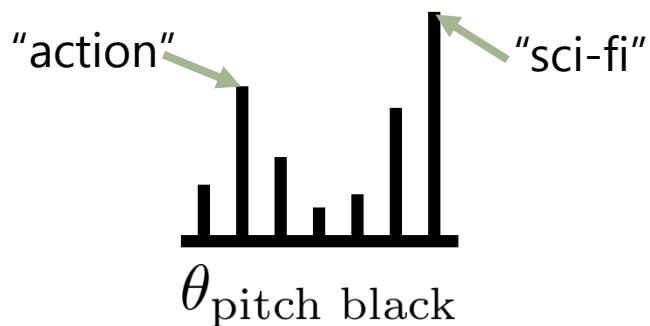
```
for j = 1 .. length of document:  
  sample a topic for the word:  
   $z_{dj} \leftarrow \theta_d$   
  sample a word from the topic:  
   $w_j \leftarrow \phi_{\{z_{dj}\}}$ 
```

Since each word is sampled independently, the output of this process is a **bag of words**

Latent Dirichlet Allocation

LDA assumes the following “process” that generates the words in a document

e.g. generate a likely review for pitch black:



j	Sample a topic	Sample a word
1	$z_{d1} = 2$	"explosion"
2	$z_{d2} = 7$	"space"
3	$z_{d3} = 2$	"bang"
4	$z_{d4} = 7$	"future"
5	$z_{d5} = 7$	"planet"
6	$z_{d6} = 6$	"acting"
7	$z_{d7} = 2$	"explosion"

Latent Dirichlet Allocation

Under this model, we can estimate the probability of a particular bag-of-words appearing with a particular topic and word distribution

The diagram shows the equation $p(d|\theta, \phi, z) = \prod_{j=1}^{\text{length of } d} \theta_{z_{d,j}} \phi_{z_{d,j}, w_{d,j}}$. Annotations include: 'document' pointing to d ; 'iterate over word positions' pointing to the product symbol; 'probability of this word's topic' pointing to $\theta_{z_{d,j}}$; and 'probability of observing this word in this topic' pointing to $\phi_{z_{d,j}, w_{d,j}}$. A bracket under the parameters θ, ϕ, z is also present.

$$p(d|\theta, \phi, z) = \prod_{j=1}^{\text{length of } d} \theta_{z_{d,j}} \phi_{z_{d,j}, w_{d,j}}$$

Problem: we need to estimate all this stuff before we can compute this probability!

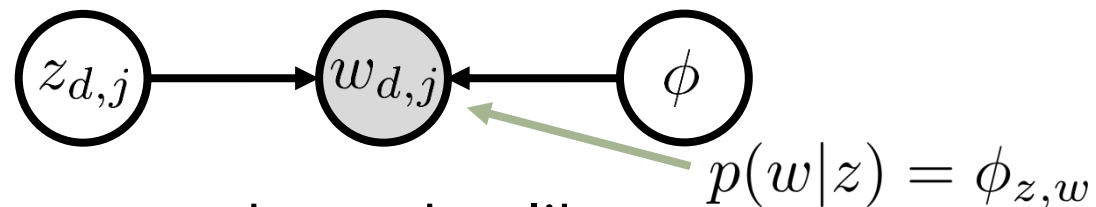
Latent Dirichlet Allocation

We need to estimate the topics (θ), the word distributions (ϕ) **and** the topic assignments (z , latent variables) that explain the observations (the words in the document)

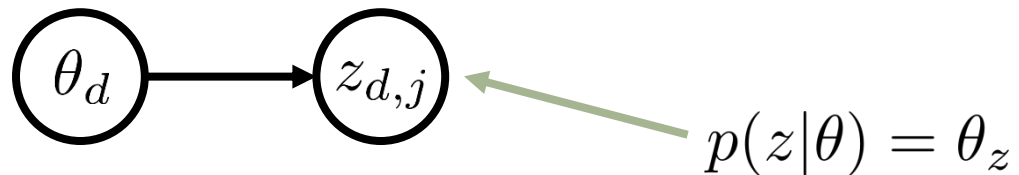
We can write down the dependencies between these variables using a (big!) **graphical model**

Latent Dirichlet Allocation

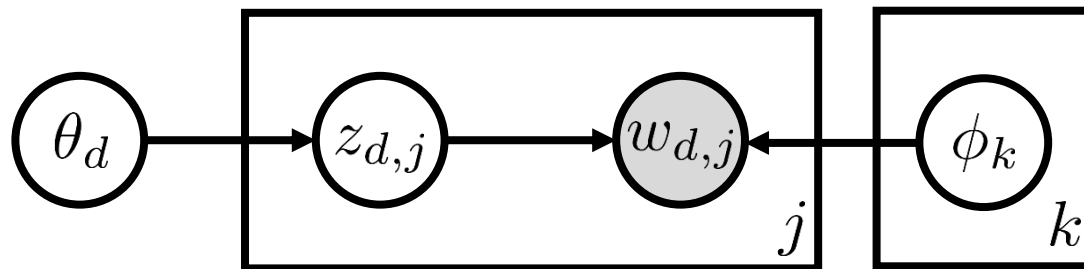
For every single word we have an edge like:



and an edge like:



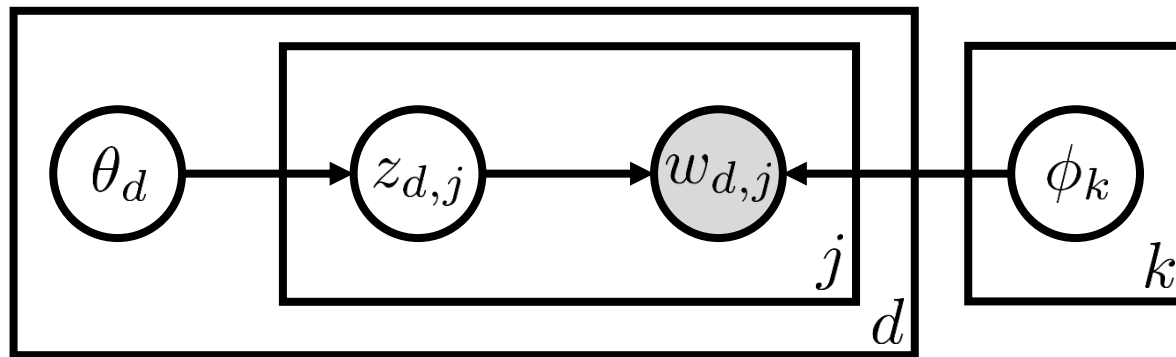
for convenience we draw this like:



(this is called "plate notation")

Latent Dirichlet Allocation

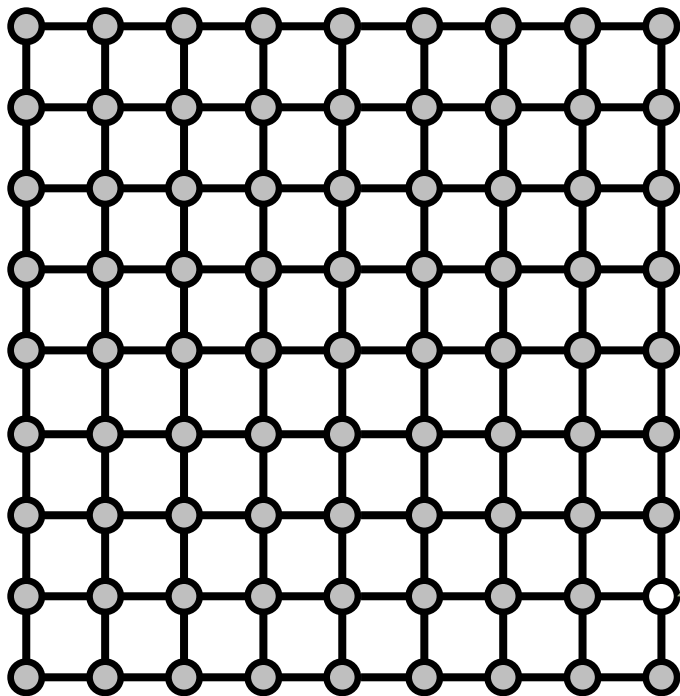
And we have a copy of this for every document!



Finally we have to estimate the parameters of this (rather large) model

Gibbs Sampling

Modeling fitting is traditionally done by **Gibbs Sampling**. This is a very simple procedure that works as follows:



1. Start with some initial values of the parameters
2. For each variable (according to some schedule), condition on its neighbors
3. **Sample** a new value for that variable (y) according to $p(y|\text{neighbors})$
4. Repeat until you get bored

Gibbs Sampling

Modeling fitting is traditionally done by **Gibbs Sampling**. This is a very simple procedure that works as follows:

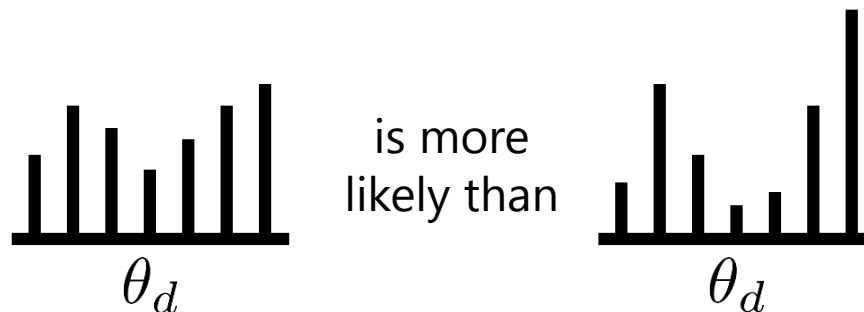
Gibbs Sampling has useful theoretical properties, most critically that the probability of a variable occupying a particular state (over a sequence of samples) is equal to the true marginal distribution, so we can (eventually) estimate the unknowns (θ , ϕ , and z) in this way

Gibbs Sampling

What about regularization?

How should we go about fitting topic distributions for documents with few words, or word distributions of topics that rarely occur?

- Much as we do with a regularizer, we'd like to penalize the deviation from uniformity
- That is, we'd like to penalize θ and ϕ for being too non-uniform



Gibbs Sampling

Since we have a probabilistic model, we want to be able to write down our regularizer as a **probability** of observing certain values for our parameters

$$p(\theta_d) = ? \quad p(\phi_k) = ?$$

- We want the probability to be higher for θ and ϕ closer to uniform
 - This property is captured by a **Dirichlet distribution**

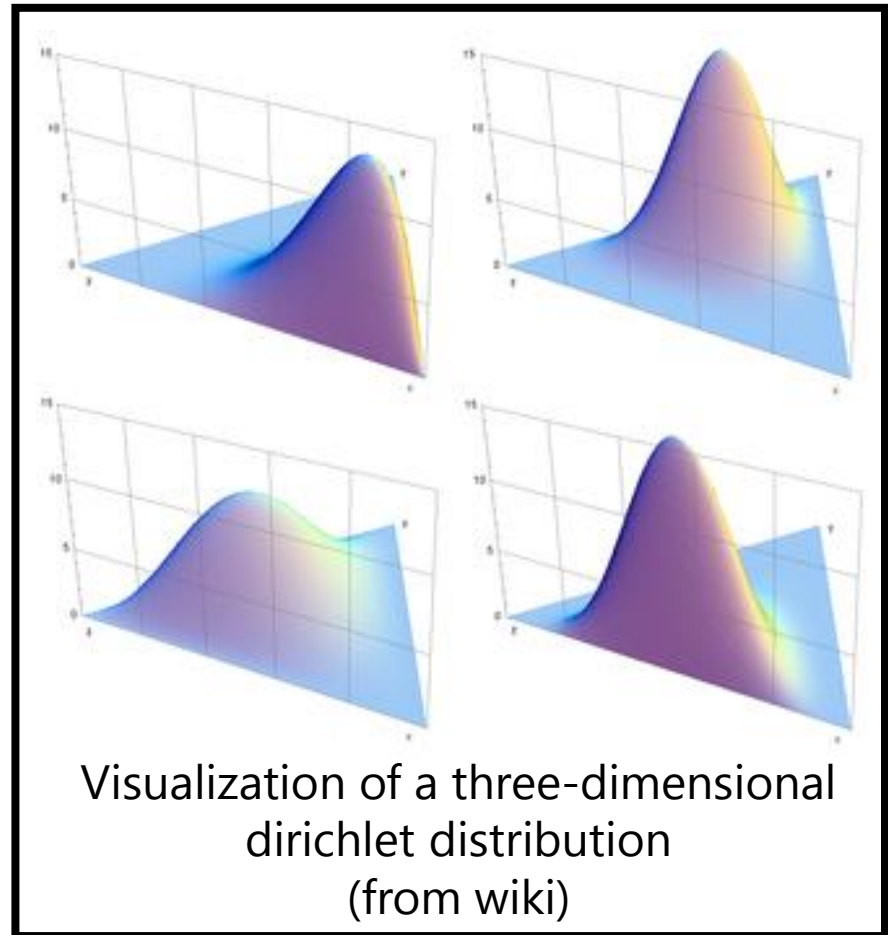
Dirichlet distribution

A Dirichlet distribution “generates” multinomial distributions. That is, its support is the set of points that lie on a simplex (i.e., positive values that add to 1)

concentration parameters

$$\text{p.d.f.}: \frac{1}{B(\alpha)} \prod_{i=1}^K x_i^{\alpha_i - 1}$$

beta function



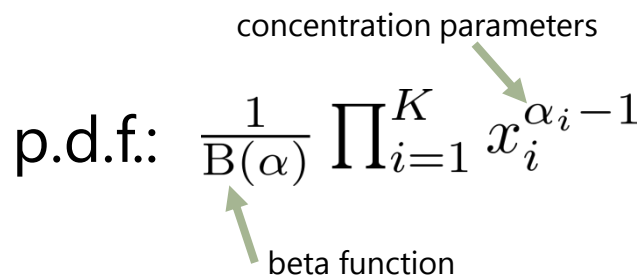
Dirichlet distribution

The concentration parameters α encode our prior probability of certain topics having higher likelihood than others

concentration parameters

$$\text{p.d.f.: } \frac{1}{B(\alpha)} \prod_{i=1}^K x_i^{\alpha_i - 1}$$

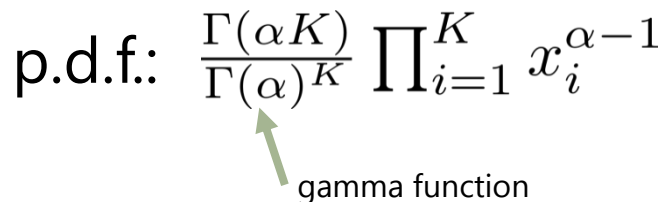
beta function



- In the most typical case, we want to penalize deviation from uniformity, in which case α is a uniform vector
- In this case the expression simplifies to the **symmetric** Dirichlet distribution:

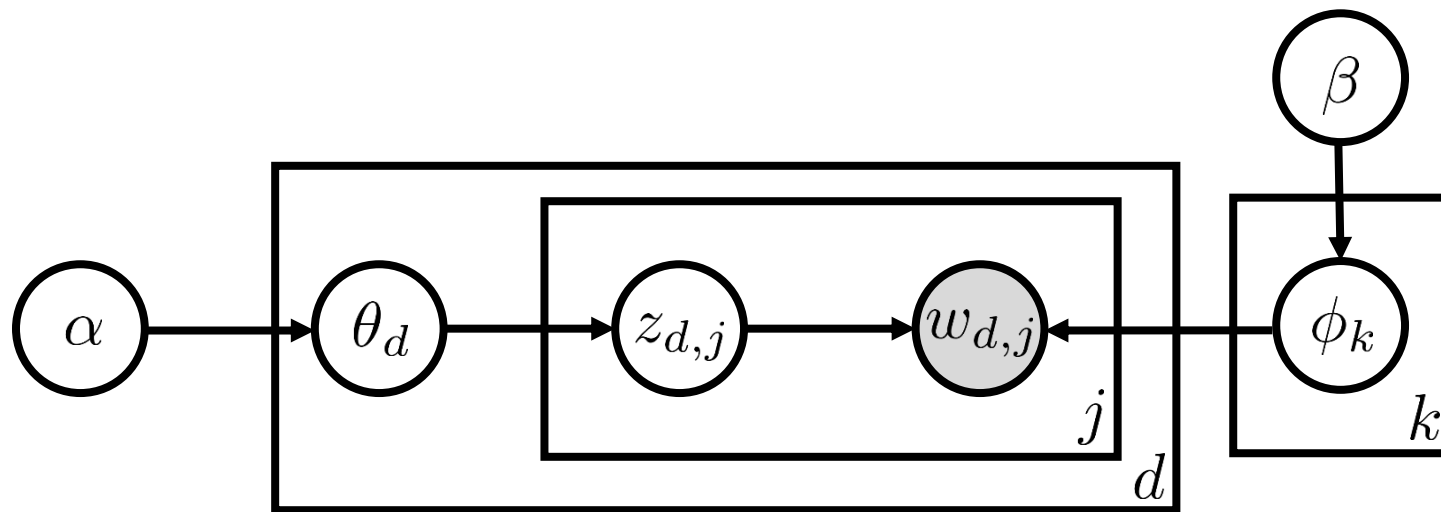
$$\text{p.d.f.: } \frac{\Gamma(\alpha K)}{\Gamma(\alpha)^K} \prod_{i=1}^K x_i^{\alpha - 1}$$

gamma function



Latent Dirichlet Allocation

These two parameters now just become additional unknowns in the model:




- The larger the values of alpha/beta, the more we penalize deviation from uniformity
- Usually we'll set these parameters by grid search, just as we do when choosing other regularization parameters

Latent Dirichlet Allocation

E.g. some topics discovered from an Associated Press corpus

labels are
determined
manually



“Arts”	“Budgets”	“Children”	“Education”
NEW	MILLION	CHILDREN	SCHOOL
FILM	TAX	WOMEN	STUDENTS
SHOW	PROGRAM	PEOPLE	SCHOOLS
MUSIC	BUDGET	CHILD	EDUCATION
MOVIE	BILLION	YEARS	TEACHERS
PLAY	FEDERAL	FAMILIES	HIGH
MUSICAL	YEAR	WORK	PUBLIC
BEST	SPENDING	PARENTS	TEACHER
ACTOR	NEW	SAYS	BENNETT
FIRST	STATE	FAMILY	MANIGAT
YORK	PLAN	WELFARE	NAMPHY
OPERA	MONEY	MEN	STATE
THEATER	PROGRAMS	PERCENT	PRESIDENT
ACTRESS	GOVERNMENT	CARE	ELEMENTARY
LOVE	CONGRESS	LIFE	HAITI

Latent Dirichlet Allocation

And the topics most likely to have generated each word in a document

labels are
determined
manually



“Arts”

“Budgets”

“Children”

“Education”

NEW
FILM

MILLION
TAX

CHILDREN
WOMEN

SCHOOL
STUDENTS

The William Randolph Hearst Foundation will give \$1.25 million to Lincoln Center, Metropolitan Opera Co., New York Philharmonic and Juilliard School. “Our board felt that we had a real opportunity to make a mark on the future of the performing arts with these grants an act every bit as important as our traditional areas of support in health, medical research, education and the social services,” Hearst Foundation President Randolph A. Hearst said Monday in announcing the grants. Lincoln Center’s share will be \$200,000 for its new building, which will house young artists and provide new public facilities. The Metropolitan Opera Co. and New York Philharmonic will receive \$400,000 each. The Juilliard School, where music and the performing arts are taught, will get \$250,000. The Hearst Foundation, a leading supporter of the Lincoln Center Consolidated Corporate Fund, will make its usual annual \$100,000 donation, too.

Latent Dirichlet Allocation

Many many many extensions of Latent Dirichlet Allocation have been proposed:

- To handle temporally evolving data:

“Topics over time: a non-Markov continuous-time model of topical trends” (Wang & McCallum, 2006)

<http://people.cs.umass.edu/~mccallum/papers/tot-kdd06.pdf>

- To handle **relational** data:

“Block-LDA: Jointly modeling entity-annotated text and entity-entity links” (Balasubramanian & Cohen, 2011)

<http://www.cs.cmu.edu/~wcohen/postscript/sdm-2011-sub.pdf>

“Relational topic models for document networks” (Chang & Blei, 2009)

<https://www.cs.princeton.edu/~blei/papers/ChangBlei2009.pdf>

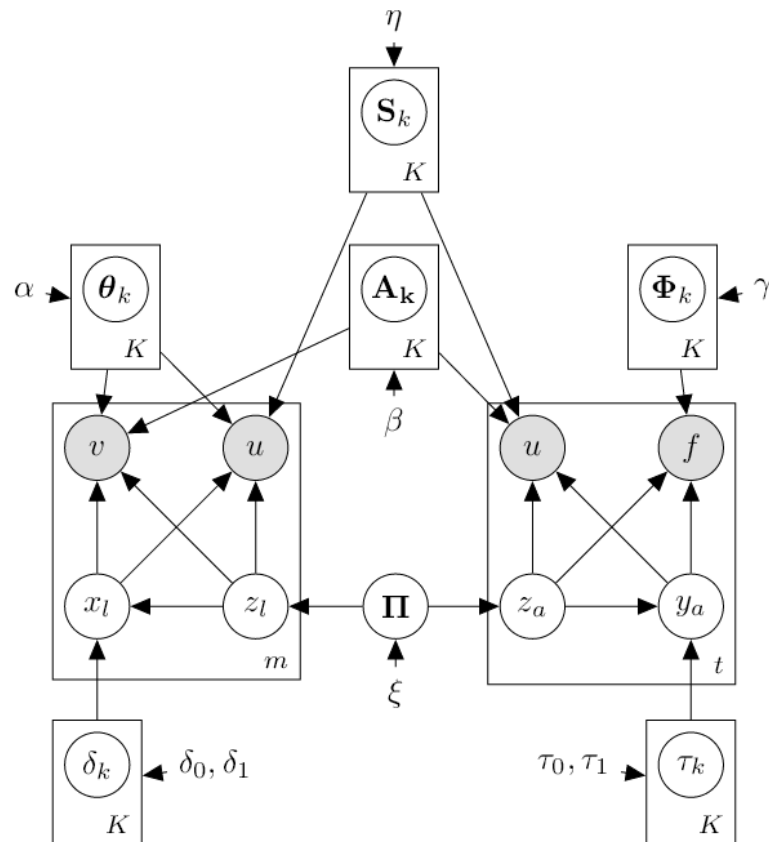
“Topic-link LDA: joint models of topic and author community” (Liu, Nicescu-Mizil, & Gryc, 2009)

<http://www.niculescu-mizil.org/papers/Link-LDA2.crc.pdf>

Latent Dirichlet Allocation

Many many many extensions of Latent Dirichlet Allocation have been proposed:

“WTFW” model
(Barbieri, Bonch, &
Manco, 2014), a model
for relational documents



Latent Dirichlet Allocation

Many many many extensions of Latent Dirichlet Allocation have been proposed:

- To handle user opinions & rating data

Case study!

Summary

Today...

Using **text** to solve predictive tasks

- Representing documents using bags-of-words and TF-IDF weighted vectors
- Stemming & stopwords
- Sentiment analysis and classification

Dimensionality reduction approaches:

- Latent Semantic Analysis
- Latent Dirichlet Allocation

Questions?

Further reading:

- Latent semantic analysis

“An introduction to Latent Semantic Analysis” (Landauer, Foltz, & Laham, 1998)

<http://lsa.colorado.edu/papers/dp1.LSAintro.pdf>

- LDA

“Latent Dirichlet Allocation” (Blei, Ng, & Jordan, 2003)

http://machinelearning.wustl.edu/mlpapers/paper_files/BleiNJ03.pdf

- Plate notation

http://en.wikipedia.org/wiki/Plate_notation

“Operations for Learning with Graphical Models” (Buntine, 1994)

<http://www.cs.cmu.edu/afs/cs/project/jair/pub/volume2/buntine94a.pdf>