Identifying & Scoring Topics in Amazon Reviews to Facilitate Informed Purchases

Kunal Kotian
Sep 11, 2018

Background

- 7 week class project requiring the development of a web app utilizing data to solve a practical problem
- 5 person team; my role – leading the modeling effort
- Motivation:
  Equip shoppers with information to help make informed purchases
- Focus on health supplements

Comparing Fish Oils on Amazon.com

Product A

See more choices

New Chapter Fish Oil Supplement - Wholemega Wild Alaskan Salmon Oil with Omega...

991

4.4 out of 5 stars

Product B

Number One Nutrition Krill Oil 500mg, 60 count

176

4.2 out of 5 stars

Same Products on our Web App

Product A

New Chapter Wholemega

★★★★★

Out of 213 reviews...

Buy it now!

Product B

#1 Recommended Krill Oil Omega 3

★★★★★

Out of 84 reviews...

Buy it now!

Relatively greater discussion of **efficacy**

Product Page Showing *Other Topics*

- Top 3 other topics shown
- Each topic shows top 5 words most frequent words in it, also present in the product's reviews

New Chapter Wholemega Whole Fish Oil, 30 Softgels

★★★★★

Out of 213 reviews...

Buy it now!

Topic Scores

Cost :

Efficacy :

Service :

Other Topics

- Flavor Taste : Taste, Flavor, Oil, Fish, Add
- Purchase Related : Amazon, Price, Brand, Quality, Vitamin
- Chronic Ailments : Doctor, Level, Supplement, Cholesterol, Research

Data

- Source: Amazon reviews collected between May 1996 and July 2014 by Prof. Julian McAuley, UCSD[1]
- Only *Health and Personal Care* category products (2.9 M)
- Filtered out
  - Reviews with over 2000 words (~1%)
  - Reviews with less than 5 stars
    - Topic scoring model only sees reviews relating to positive customer experiences
    - "*Customers who liked this product discussed X aspect to Y extent*"
- 26,818 products and 217,530 reviews

1: http://jmcauley.ucsd.edu/data/amazon/

Data Pre-processing

Tokenization

- Convert each review from a string → list of words (tokens)
- For each *word token*, stored its
  1. Lemmatized form
  2. Part-of-speech tag → for phrase filtering

‘This chapter **is** **better** than the last one.’

['this’, 'chapter’, '**be**’, '**good**’, 'than’, 'the’, 'last’, 'one’]
['DET', 'NOUN', 'VERB', 'ADJ', 'ADP', 'DET', 'ADJ', 'NOUN', 'PUNCT']

Phrase/Collocation Detection

- For every pair of words, check if NPMI[1] > threshold

Pr("x y")

$$npmi = \frac{\log \frac{p(x,y)}{p(x)p(y)}}{-\log p(x,y)}$$

- Words x and y *never* co-occur → NPMI = -1
- Words x and y *randomly* co-occur → NPMI = 0
- Words x and y *always* co-occur → NPMI = 1

1: G. Bouma, "Normalized (pointwise) mutual information in collocation extraction," Proceedings of GSCL, pp. 31–40, 2009

Phrase/Collocation Detection

- 2 passes of phrase detection → Bigram and trigram phrases
- Reject phrases that are not information-rich about *meaningful topics*[1]
  - Bigram phrases: [noun or adjective]_[noun]
  - Trigram phrases: [noun or adjective]_[any part-of-speech]_[noun or adjective]

	Accept	Reject
Bigram phrase	great_taste	at_least
Trigram phrase	health_food_store	could_not_believe

- Top phrases before filtering: this_product, do_not, seem_to
- Top phrases after filtering: fish_oil, side_effect, high_quality

1: D. Johnson, V. Malhotra, and P. Vamplew, "More effective web search using bigrams and trigrams," Webology, vol. 3, no. 4, p. 35, 2006

Scoring
Efficacy, *Cost*, & *Service*

Topic Score

- First, attempted to guide LDA to learn efficacy, cost, service (ECS)
- Finally built rules-based model (steps):
  1. Build 3 lists of target words representing each of ECS
  2. Topic score: For each **product p**, and for each **topic t** out of ECS,

$$\text{topic t's score} = \frac{\#\text{ of reviews containing at least 1 word } \in [\text{t's target words}]}{\text{total \# of reviews for that product p}}$$

Topic Score

- Why use binary indicators instead of counts for scoring?

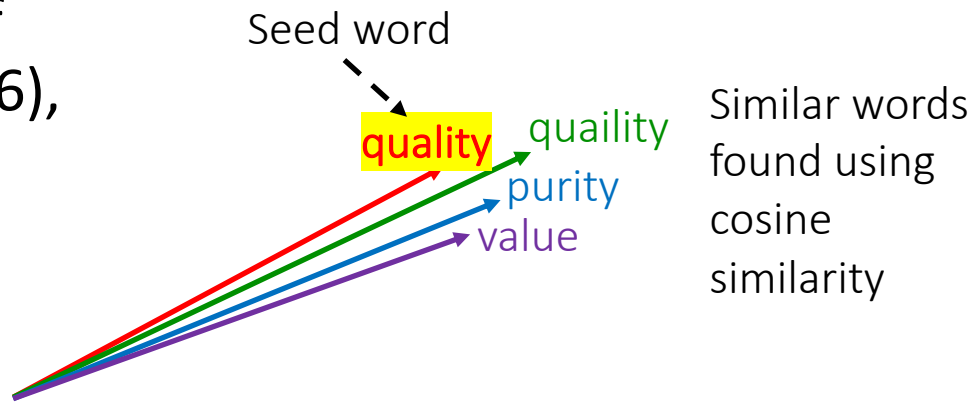
Product ID	Review ID	Review Text	Topic represented? Yes=1/No=0		
			Efficacy	Cost	Service
1001	3	"I bought this for…"	1	0	1

Product ID	Review ID	Review Text	Number of Matching Words		
			Efficacy	Cost	Service
1001	3	"I bought this for…"	15	3	2

- Reviewers' biases in frequency of discussing different topics
- Binary indicators assign equal weight to each review; counts give weight to verbosity

Building Lists of Target Words

- Problems with manually creating lists of target words
  - Unlikely to cover majority of words related to the target topic
  - Unlikely to cover all/most misspellings of target words
- Solution: Word2Vec
  - Learn 100-dim vector representations for each word in the corpus
  - Words appearing in similar contexts → vectors close to each other
  - Used cosine similarity to build lists of target words for efficacy (82), cost (56), service (94)



Uncovering *Other Topics*

Topic Modeling – Latent Dirichlet Allocation

- Generative model
- Bag of words
- # of topics unknown
- Document-topic & topic-word distributions unknown

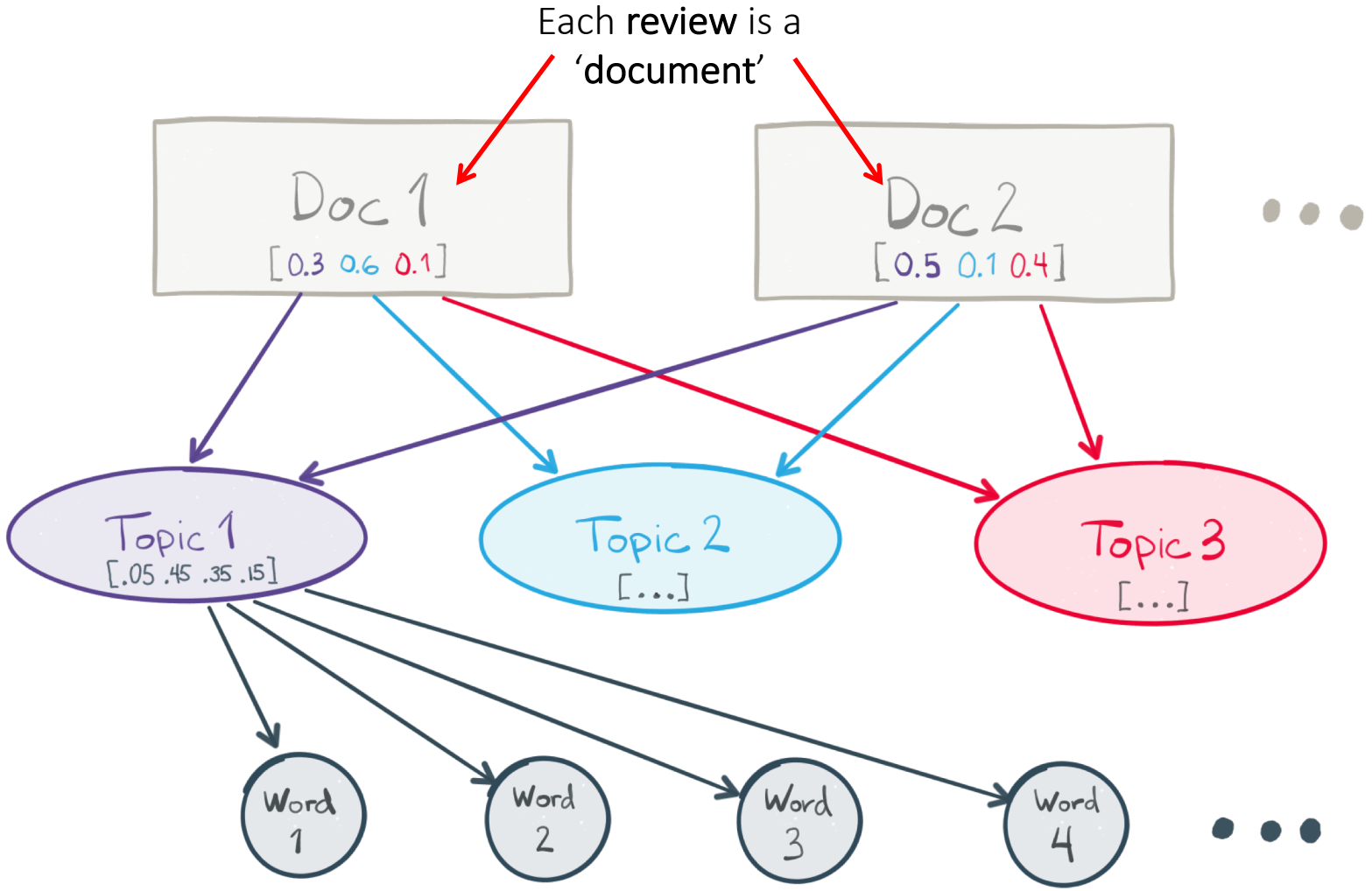
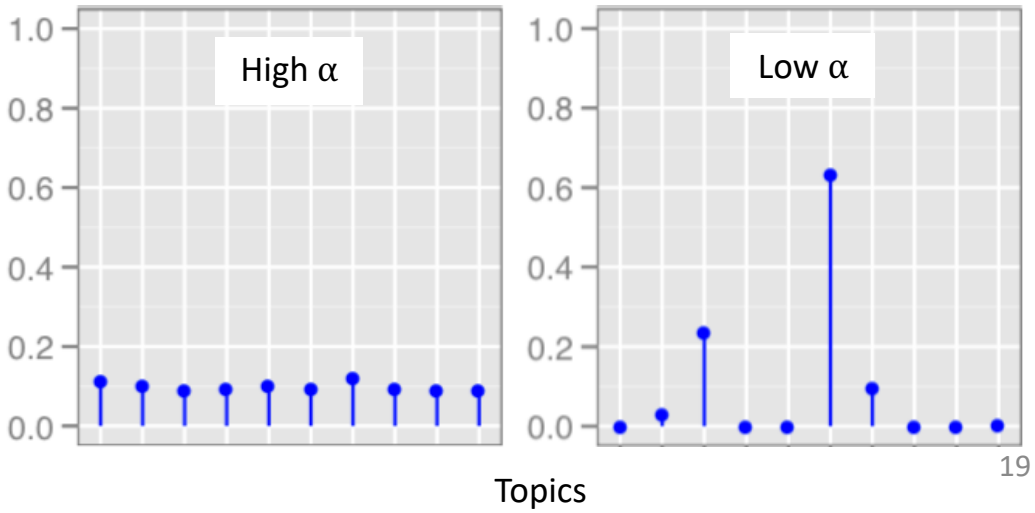


Image source: Patrick Harrison, Link: https://github.com/skipgram/modern-nlp-in-python/blob/master/executable/Modern_NLP_in_Python.ipynb

LDA Hyper-parameters

- α: Prior per-document topic distribution
  - Proportional to # of topics representing each review
  - Set to be updated during LDA training
- β: Prior per-topic word distribution
  - Proportional to # of words representing each topic
  - Low β leads to less topic overlap
  - Set to array of 1/num_words
- Number of Topics
  - Train LDA models with different # of topics & pick one with highest coherence score

Each review samples topics from this PMF:



Topic Coherence Score

1. For each topic, order the top n words by their probability *p(word|topic)*
2. For each topic, compute coherence score

$$\text{Coherence} = \sum_{i<j} \text{score}(w_i, w_j)$$

Less common word

More common word

$$score(w_i, w_j) = \log p(w_j|w_i) = \log \frac{doc\_count(w_i, w_j) + 1}{doc\_count(w_i)}$$

Within a topic, to what extent do common words trigger the occurrence of rare words?

3. Mean coherence score across topics gives the model's coherence score

Best LDA Model

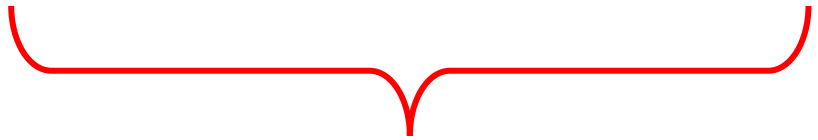
- Topic coherence + manual inspection of topics → **13-topic** LDA model
- 8 coherent topics were labeled
  - Workout- related: [energy, lose, feel, week, weight, pedometer, notice, step, workout, walk, increase, pound, exercise, …]
  - Appearance-related: [skin, hair, nail, face, dry, acne, look, cream, soft, smooth, apply, moisturizer, wrinkle, ...]

Labeled *Other* Topics:

1) Common Ailments
2) Flavor/Taste
3) Workout-related
4) Chronic Ailments
5) Purchase-related
6) Appearance-related
7) Product Form
8) Gut Health

Choosing Top 3 *Other Topics* for Display

Product ID	Review ID	Topic Membership Probability				
		Topic 1	Topic 2	Topic 3	Topic 4	Topic 5
209	1001	0.10	0.70	0.10	0.05	0.05
	1002	0.05	0.50	0.30	0.05	0.10
	1003	0.10	0.40	0.10	0.30	0.10
	∑	0.25	**1.60**	**0.50**	**0.40**	0.25



Displayed topics

Evaluation

- Absence of ground truth for topic labels and scores
- Manually reviewed ECS topic scores & other topics for 6 products
- Proposed solution
  - Hire human reviewers via Amazon's Mechanical Turk or Upwork
  - For a large number of **products**, have reviewers:
    (1) Read all reviews & assign ECS scores on a discrete 5-point scale
    (2) Do the top 3 other topics seem reasonable (yes/no)?
  - Exploratory analysis comparing human answers & model output