Matchmaking for Industry

Estimating Expertise from Issue Tickets with Topic Modeling

Philip Robinson

Presented to PyData PDX
Work from NASA - Jet Propulsion Lab

June 6, 2020

Presentation Overview

Philip Robinson

That's me!

Figure: JunGlow Love, 2015 PDX

- @timedebtor
- @probinso
- probinso@protonmail.com
- Work at GrammaTech
- Alumni at OHSU & WWU

Problem Description

Jet Propulsion Laboratory
California Institute of Technology

NASA's Jet Propulsion Lab uses a custom ticketing system to assign experts to "mission critical late stage anomalies". The ticketing system is used to document progress, contributors, and solutions against these tickets. A ticket manager is responsible for assigning the first contributors & experts to solve a ticket.

- Recruiting impactful teams is very time expensive
- Incomplete understanding of candidate expertise or ticket topic
- Uneven distribution of assignment
- Find experts from other divisions and projects

What is our generalized goal?

We have a ticketing system used to track progress in solving specialized tasks.

1. A first response needs to build a candidate solution team
2. We have a textual description of solved tasks with attribution
3. We would like to automatically identify candidates

Can we estimate expertise of candidates given a ticket?

Proposal - Author Modeling

Model Expertise of SME by tickets and attributions

Author modeling has been used to measure attribution[14] and contribution[1]. Author-Topic Modeling (ATM) establishes a strategy to map both authors and documents to the same topic-space over a vocabulary[16].

If I am the author, then I should be an expert on the contents

What is Topic Modeling

I love LDA based topic modeling

Topic modeling[13] is a text processing technique for learning topics from a collection of documents. This is usually used as a strategy to describe documents in a comparable low dimensional space or an exploratory tool for grouping document collections.

Examples

In practice, this requires many more documents

The Tourist huddles in the station While slowly night gives way to dawn; He finds a certain fascination In knowing all the trains are gone.

The Governess up in the attic Attempts to make a cup of tea; Her mind grows daily more erratic From cold and hunger and ennui.

The Journalist surveys the slaughter, The best in years without a doubt; He pours himself a gin and water and wonders how it came about.

- Food
- Travel
- Time

From this annotation we know that Document 2 and 3 are about Food and Time

What can I do with topic models?

- Cluster documents by topic
- Define interpretable topics that describe a corpus
- Dimmentionality reduction of documents
- Corpus aware document similarities

Topic modeling can usually be extended to address many other problems, and document embedding can be used to inform downstream models.

Latent Dirichlet Allocation[3]

Bayesian extension to PLSA[7] extends LSA/SVD[6]

- Represent document as Bag-of-Words[1]
- Model/Fit topics as mixture of words
- Documents are projected into or sampled from topic-space-distribution

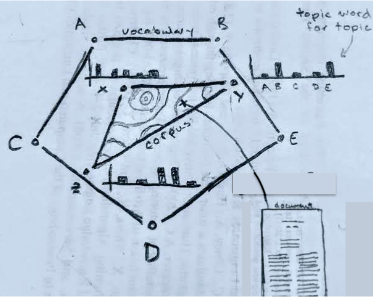


Figure: Latent Dirichlet Allocation

[1]equivalent to multinomial over the vocabulary

Looking at top words

Mitigating apophenia is hard, topics difficult to interpret

Topic #1

- server
- connected
- access
- workstation
- outage
- user

Topic #2

- mode
- instrument
- safe
- spacecraft
- anomaly
- recovery

Topic #3

- uplink
- station
- dsn
- lock
- ace
- radiation

Although the model better describes our generation process, from the perspective of topics, it can be difficult to know what these topics actually represent.

Extensions

Extensability is what makes LDA most interesting

→ Author Topic Model[16]
- Correlated Topic Model[9]
- Biterm Topic Model[19, 4]
- Twitter Topic Model[10]
- Supervised LDA[11]
- Hierarchical Dirichlet Process[18]

Approach

Author-Topic-Modeling

- Interpret doc as Bag-of-Words[2]
- Model/Fit topics as mixture of words
- Author & document are projected into topic-space
- Measure distance from author to document

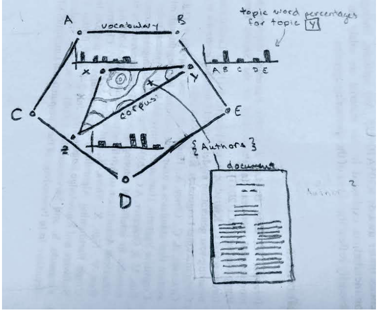


Figure: Latent Dirichlet Allocation

$$T(x) = \texttt{Project } x \texttt{ into topic-space}$$

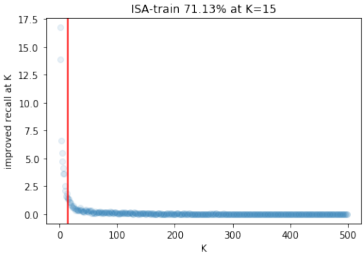
$$R_d = \underset{a \in A}{\text{argsort}} \left\{ Distance(T(a), T(d)) \right\}$$

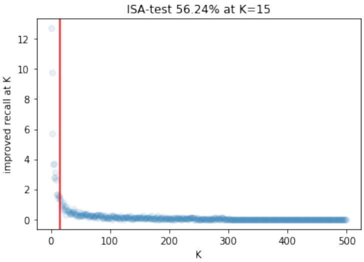
[2] equivalent to multinomial over vocabulary

Ranking

Seems to work

Given Authors/Experts in a "Topic Space" and a mapping from document to "Topic Space", we can rank experts for a document.





K is cutoff for suggested candidates

Receiving Candidates

Results begin at 4 words

It is possible to get interesting results at a document length of 4 words, however it is hard to know why these results are interesting. This is an example of directly searching for experts.

```
'gimbal drive motor friction'
```

	Name	Title	Organization
0	Amanda Donner	Mission Assurance Manager	5150
1	John Trager	NaN	337C
2	Mathew Keuneke	Product Delivery Manager	397A
3	Jessica Bowles-Martinez	Systems Engineer	313G
4	NaN	NaN	NaN

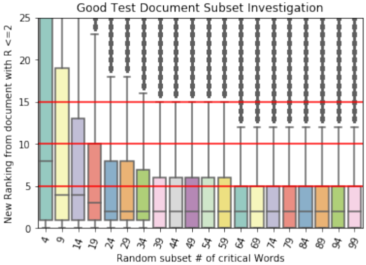
```
'rtg temperature drive curiosity capacity'
```

	Name	Title	Organization
0	John Rakiewicz	NaN	NaN
1	Angela Dorsey	Technologist	335S
2	Otfrid Liepack	Deputy System Manager	394G
3	Mohammad Shahabuddin	Flight Software Engineer	348D
4	Megan Lin	Delivery Manager	397S

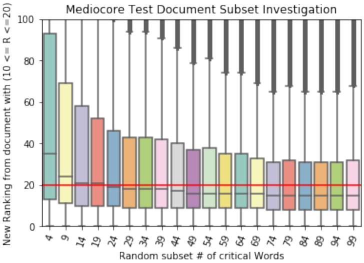
How does word count effect recall?

Best results at 30 words

We are interested in understanding how much text is required to inform our model prediction. For these plots, we randomly subset texts for known ticket-expert pairs and observe the expert's new ranking.



Expert found in top 2
24 critical words



expert found in 10-20 range
29 critical words

Distance Measures

Testing distances is cheaper than understanding them

Hellinger distance[8] commonly used for distance of points in Probability Simplex. The domain of the Dirichlet distribution can be thought of as a simplex over multinomial distributions.[3]

[3] Most online examples use cosine distance, without justification

Evaluation

Does our fit Dirichlet distribution describe our data or our understanding?

- perplexity
- predictive power (Recall / Precision)
- coherence[15, 12]
- visualization[17, 5]
- topic stability[20]
- topic significance[2]

Perplexity

perplexity for prediction

Perplexity is a measure of how poorly the model describes the data. Low perplexity indicates the distribution is a good description of the sample. For LDA this prioritizes dimensional reduction.

$$Perplexity(q) = b^{-\frac{1}{N}\sum_{x \in X} log_b q(x)}$$

Coherence

coherence for scorable EDA[5]

Topic coherence measures take the set of N top words from a topic and sums a `confirmation measure`[4] over the word pairs. Probabilities are estimated from sliding window over train and test corpora.

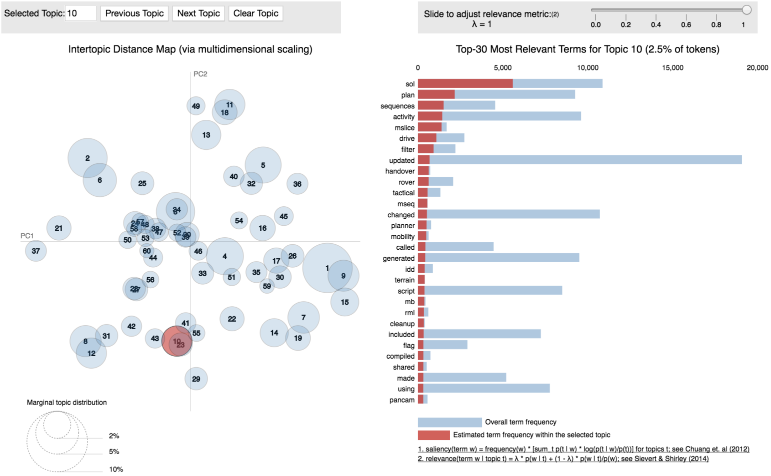
$$C_{Irvine} = \frac{2}{N \cdot N - 1} \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} PMI(w_i, w_j)$$

$$PMI(w_i, w_j) = log(\frac{P(w_i, w_j)}{P(w_i) \cdot P(w_j)})$$

[4] like pointwize mutual information (PMI)
[5] exploratory data analysis

Visualization (pyLDAvis)

Interpretable EDA



Conclusion & Questions

Bibiography I

Khaled Aldebei, Xiangjian He, Wenjing Jia, and Jie Yang.
Unsupervised multi-author document decomposition based on hidden markov model.
In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*, 2016.

Loulwah AlSumait, Daniel Barbará, James Gentle, and Carlotta Domeniconi.
Topic significance ranking of lda generative models.
In *ECML*, 2009.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan.
Latent dirichlet allocation.
J. Mach. Learn. Res., 3:993–1022, 2003.

Weizheng Chen, Jinpeng Wang, Yan Zhang, Hongfei Yan, and Xiaoming Li.
User based aggregation for biterm topic model.
In *ACL*, 2015.

Jason Chuang, Christopher D. Manning, and Jeffrey Heer.
Termite: Visualization techniques for assessing textual topic models.
In *Advanced Visual Interfaces*, 2012.

S. Deerwester, S.T. Dumais, G.W. Furnas, T.K. Landauer, and R.A. Harshman.
Indexing by latent semantic analysis.
Journal of the American Society for Information Science 41, pages 391–407, 1990.

Thomas Hofmann.
Probabilistic latent semantic analysis.
CoRR, abs/1301.6705, 2013.

Kriste Krstovski, David A. Smith, Hanna Wallach, and Andrew McGregor.
Efficient nearest-neighbor search in the probability simplex.
In *2013 Conference on the Theory of Information Retrieval*, September 2013.

Bibiography II

John D. Lafferty and David M. Blei.
Correlated topic models.
In Y. Weiss, B. Schölkopf, and J. C. Platt, editors, *Advances in Neural Information Processing Systems 18*, pages 147–154. MIT Press, 2006.

Kar Wai Lim, Changyou Chen, and Wray L. Buntine.
Twitter-network topic model: A full bayesian treatment for social network and text modeling.
CoRR, abs/1609.06791, 2016.

Jon D. Mcauliffe and David M. Blei.
Supervised topic models.
In J. C. Platt, D. Koller, Y. Singer, and S. T. Roweis, editors, *Advances in Neural Information Processing Systems 20*, pages 121–128. Curran Associates, Inc., 2008.

Dat Quoc Nguyen, Richard Billingsley, Lan Du, and Mark Johnson.
Improving topic models with latent feature word representations.
Transactions of the Association for Computational Linguistics, 3:299–313, 2015.

Li Ren.
A survey on statistical topic modeling.
2013.

Andi Rexha, Mark Kröll, Hermann Ziak, and Roman Kern.
Authorship identification of documents with high content similarity.
Scientometrics, 115(1):223–237, Apr 2018.

Michael Röder, Andreas Both, and Alexander Hinneburg.
Exploring the space of topic coherence measures.
In *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining*, WSDM '15, pages 399–408, New York, NY, USA, 2015. ACM.

Bibiography III

Michal Rosen-Zvi, Thomas Griffiths, Mark Steyvers, and Padhraic Smyth.
The author-topic model for authors and documents.
In *Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence*, UAI '04, pages 487–494, Arlington, Virginia, United States, 2004. AUAI Press.

Carson Sievert and Kenneth Shirley.
LDAvis: A method for visualizing and interpreting topics.
In *Proceedings of the Workshop on Interactive Language Learning, Visualization, and Interfaces*, pages 63–70, Baltimore, Maryland, USA, June 2014. Association for Computational Linguistics.

Yee Whye Teh, Michael I. Jordan, Matthew J. Beal, and David M. Blei.
Hierarchical dirichlet processes.
Journal of the American Statistical Association, 101(476):1566–1581, 2006.

Xiaohui Yan, Jiafeng Guo, Yanyan Lan, and Xueqi Cheng.
A biterm topic model for short texts.
In *Proceedings of the 22Nd International Conference on World Wide Web*, WWW '13, pages 1445–1456, New York, NY, USA, 2013. ACM.

Yi Yang, Shimei Pan, Jie Lu, Mercan Topkara, and Yangqiu Song.
The stability and usability of statistical topic models.
ACM Trans. Interact. Intell. Syst., 6(2):14:1–14:23, July 2016.

Text Pre-Processing

Cleaning applies to most 'simple' NLP problems

Text normalization is custom to your corpus. Many of the steps are the same, but their application changes with the type of documents.

- Normalize text
  - Lowercase text
  - ⋆ Remove non-informative text patterns
- Tokenization & (Stemming — Lemmatization)
  - ⋆ pick a stemmer
  - Stem `(applies, applying, apply) -> (appli)`
  - ⋆ Un-Stem `(appli) -> (apply)`
- Focus corpus (remove "stop words")
  - drop most frequent words
  - `nltk` English stop-words
  - Remove extremely rare words

Non-Informative Delinquent Cases

Evaluation metrics are only informative given reasonable parameters

You can often reduce perplexity by having fewer topics. Maximizing coherence is more resilient to this effect.

Verifying your intent

You may not need interpretable topics!

Base LDA, on it's own, isn't that great. Understanding LDA allows you to understand the extensions, which are pretty cool.

Not all evaluation metrics have been written for the extensions, so you may have to come up with proxies.

Steps to Success

- Perform text level EDA to customize cleaning processing
- Pick a model type
- Evaluation takes care
  - Identify a model-fit measure
  - Identify a performance strategy

Simply put, LDA attempts to generalize truncated SVD with a generative bias to how we write papers.

Latent Semantic Analysis

SVD on Vocabulary x Document matrix

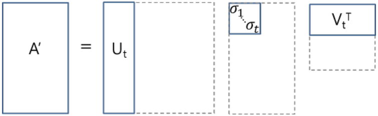
Given: D documents covering W words

- Create A_{DxW} counting or `tfidf` matrix

  $a_{i,j} = tf_{i,j} \times log \frac{D}{df_j}$

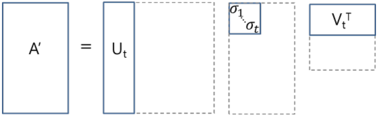
- Compute Singular Value Decomposition
- Select the number of description topics t

$$A' \approx U_t S_t V_t^T$$



Understand Latent Semantic Analysis

Topics are principle components of entire document collection



- **U:** document-topic matrix, topic contributes to document
- **S:** singular values
- **V:** word-topic matrix, topic contribute to words

- Overfit as consequence of topics strict mathematical definition
- Topics are better interpreted as mathematical than intuitive
- Cost of finding one topic is the same as finding all possible topics

Goals of generative models

A generative model

- Assume/Generalize how data could have been generated
- Fit distributions that describe generalization
- Ask questions about the generalization in relation to data
- Ask questions about data in relation to the generalization

Generative models are much easier to extend, because they abstract the model from it's linear algebra dependencies.

Topic modeling generalizes how a document is generated by claiming that words come from topics, and documents have multiple topics.[6]

[6] this is not a language model

Probabilistic Latent Semantic Analysis

Generative model for SVD

$$P(d, w) :\rightarrow \texttt{document-term matrix}$$

- $P(z|d)$ is the probability z contributes to d
- $P(w|z)$ is the probability w contributes to z

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

$$P(Z|D) \text{ and } P(W|Z) \sim \texttt{Multinomial}$$

Understand Probabilistic Latent Semantic Analysis

A mapping to SVD

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

`remembering`

$$A \approx U_t S_t V_t^T$$

First generate the topic Z then generate the word W

- P(D) is not parameterized, we don't observe new documents
- Tends to be softer than LSA, but still overfits (grows with D)
- No longer use `tfidf` best replaced with `stopwords`[7]

[7] Usually top $.5 - 2\%$ of vocab