0. Agenda

Saturday, February 02, 2019 3:47 PM

1. Before diving into LDA, What is Topic Modelling?
2. PAPER Intro Section Elaborated: How did LDA evolve into being?
   a. TF-IDF Matrix
   b. LSI
   c. ~~Unigram Model~~
   d. ~~Unigram Mixture Models~~
   e. pLSI
   f. then comes LDA
3. The simplified explanation for LDA
4. Pseudo code for implementation of LDA
5. The mathematics behind LDA
6. Conclusion/Summary

1. Topic Modeling

Saturday, February 02, 2019 9:34 PM

- **Topic Modeling**: = A process to automatically identify topics (a.k.a subject/ theme) from a text corpus
- **Topics**: = A set of co-occurring words in the corpus
- Topic Modeling produces '**Topics**' as outputs after processing a collection of docs

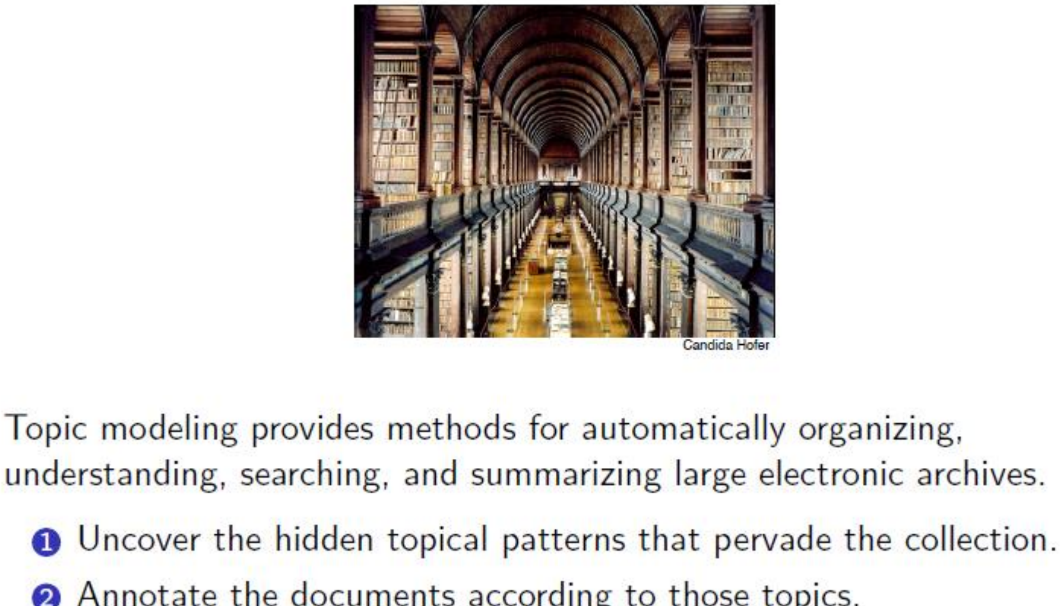
Intuition: Each document in a corpus could be comprised of one or more topics

One main application area for Topic Modeling:

www.betaversion.org/~stefano/linotype/news/26/

As more information becomes available, it becomes more difficult to access what we are looking for.

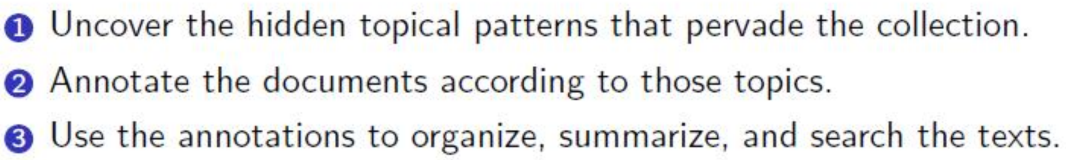
We need new tools to help us organize, search, and understand these vast amounts of information.



Candida Hofer

Topic modeling provides methods for automatically organizing, understanding, searching, and summarizing large electronic archives.

1. Uncover the hidden topical patterns that pervade the collection.
2. Annotate the documents according to those topics.



1. Uncover the hidden topical patterns that pervade the collection.
2. Annotate the documents according to those topics.
3. Use the annotations to organize, summarize, and search the texts.

2. Evolution of LDA

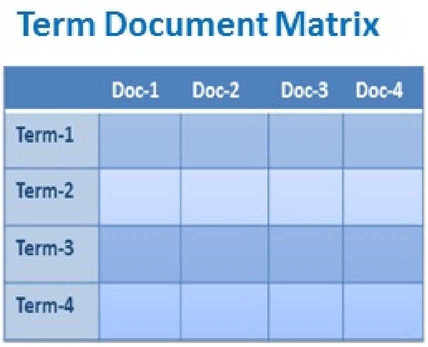
Saturday, February 02, 2019 9:37 PM

1. **TF- IDF**:

TF = tf(t,d)

- raw count of a term t in a document d (simple version)
- Will also be **normalized**:
  What if a document d is naturally very long, hence that term t might occur much more frequently. To avoid bias towards longer corpus, we "normalize" the term freq.
    - One way to normalize is to divide the particular tf(t,d)/ sum(all terms in d)
    - Another way: tf(t,d)/tf(maxt,d) where maxt is the most occurring term in the document d

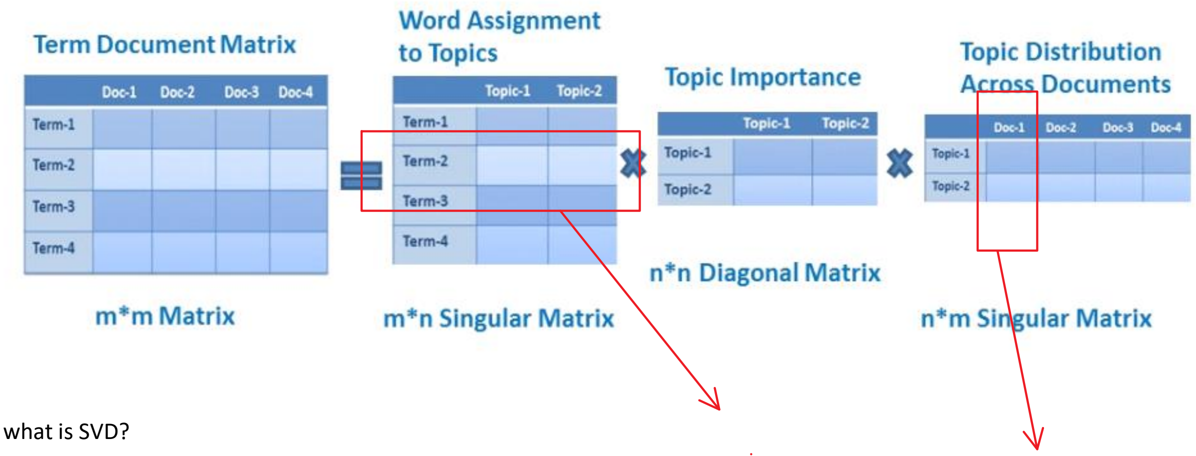
IDF =
IDF = log (N/nt)
N= Number of documents in the corpus
nt = number of documents containing the term t generally log scaled)



Term Document Matrix

	Doc-1	Doc-2	Doc-3	Doc-4
Term-1				
Term-2				
Term-3				
Term-4				

2. **Latent Semantic Indexing or Latent Semantic Analysis**:
Based on the concept of SVD, where it will factorize the TF-IDF Matrix (Term Document Matrix) into the following:

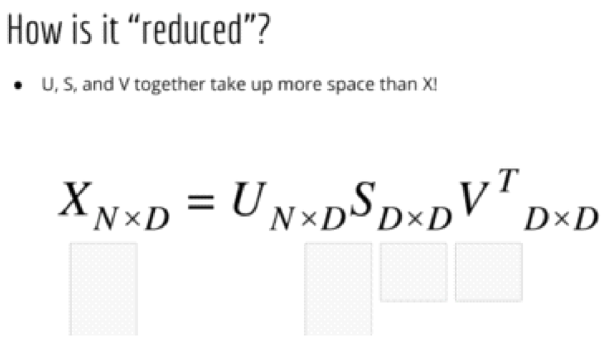


Ok, what is SVD?

Information about every term - represented in terms of topics - is coded into the rows of the matrix

Information about every doc is coded

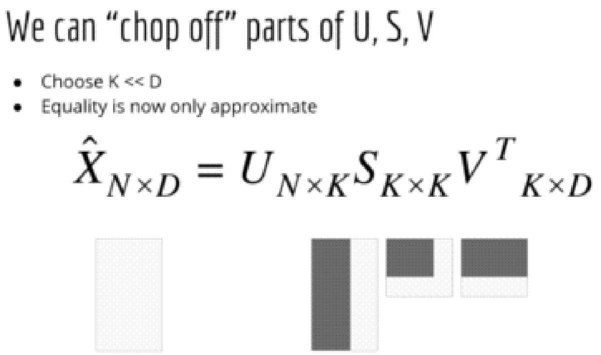
$$X = USV^T$$



How is it “reduced”?

- U, S, and V together take up more space than X!

$$X_{N\times D} = U_{N\times D} S_{D\times D} V^{T}{}_{D\times D}$$



We can “chop off” parts of U, S, V

- Choose K << D
- Equality is now only approximate

$$\hat{X}_{N\times D} = U_{N\times K} S_{K\times K} V^{T}{}_{K\times D}$$

U - the left singular matrix
V - the right singular matrix
S - the diagonal matrix where only diagonal elements exist

U and V are also called unitary matrices because multiplying by their respective conjugate transposes yields identity matrices

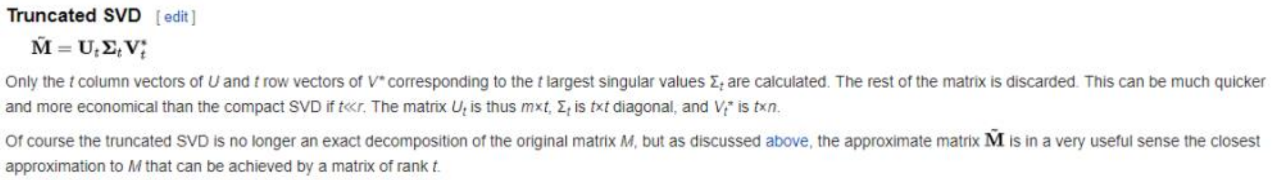
$$\mathbf{U}\mathbf{U}^{\mathsf{T}} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1 \\ 1 & 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \mathbf{I}_4$$

$$\mathbf{V}\mathbf{V}^{\mathsf{T}} = \begin{bmatrix} 0 & 0 & \sqrt{0.2} & 0 & -\sqrt{0.8} \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & \sqrt{0.8} & 0 & \sqrt{0.2} \end{bmatrix} \cdot \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ \sqrt{0.2} & 0 & 0 & 0 & \sqrt{0.8} \\ 0 & 0 & 0 & 1 & 0 \\ -\sqrt{0.8} & 0 & 0 & 0 & \sqrt{0.2} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} = \mathbf{I}_5$$

Singular Matrix: A matrix is singular if its determinant is 0 or a square matrix that does not have a matrix inverse.
Diagonal Matrix: It is a matrix in which the entries other than the main diagonal are all zero.

Code Implementation:
Both Gensim and sklearn implement Truncated SVD

Truncated SVD:
https://math.stackexchange.com/questions/2627005/are-reduced-svd-and-truncated-svd-the-same-thing



Truncated SVD [edit]

$$\tilde{\mathbf{M}} = \mathbf{U}_t \mathbf{\Sigma}_t \mathbf{V}_t^*$$

Only the *t* column vectors of *U* and *t* row vectors of *V** corresponding to the *t* largest singular values Σ_t are calculated. The rest of the matrix is discarded. This can be much quicker and more economical than the compact SVD if *t*≪*r*. The matrix U_t is thus *m*×*t*, Σ_t is *t*×*t* diagonal, and V_t^* is *t*×*n*.

Of course the truncated SVD is no longer an exact decomposition of the original matrix *M*, but as discussed above, the approximate matrix $\tilde{\mathbf{M}}$ is in a very useful sense the closest approximation to *M* that can be achieved by a matrix of rank *t*.

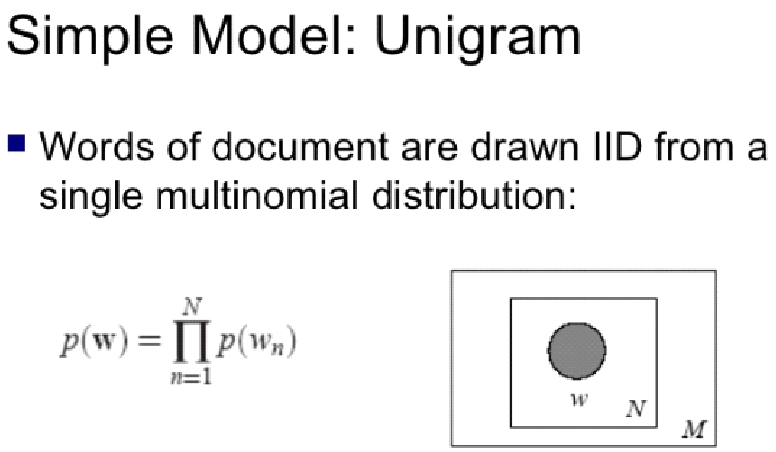
Gensim:
Gensim Documentation: https://radimrehurek.com/gensim/models/lsimodel.html
Gensim Example: https://www.datacamp.com/community/tutorials/discovering-hidden-topics-python

Sklearn Implementation:
https://www.analyticsvidhya.com/blog/2018/10/stepwise-guide-topic-modeling-latent-semantic-analysis/

3. **Probabilistic Latent Semantic Indexing a.k.a Aspect Model**:

Before PLSI, let us see about Unigram and Unigram Mixture models as predecessors

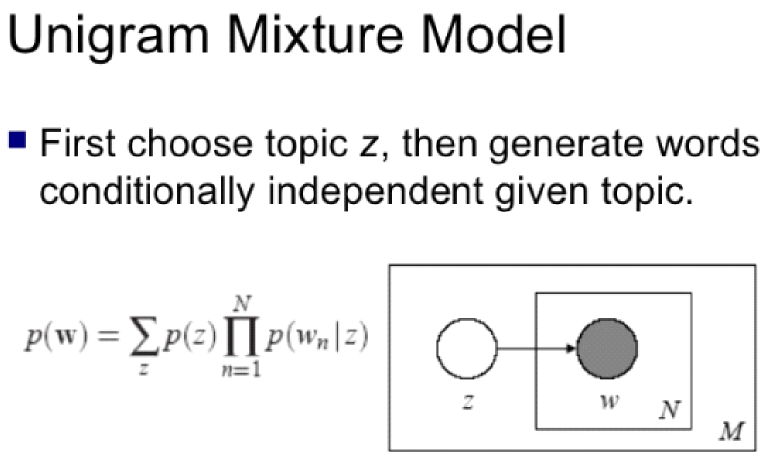
what is a unigram model (IID - Independent and Identically Distributed):



Simple Model: Unigram

- Words of document are drawn IID from a single multinomial distribution:

$$p(\mathbf{w}) = \prod_{n=1}^{N} p(w_n)$$



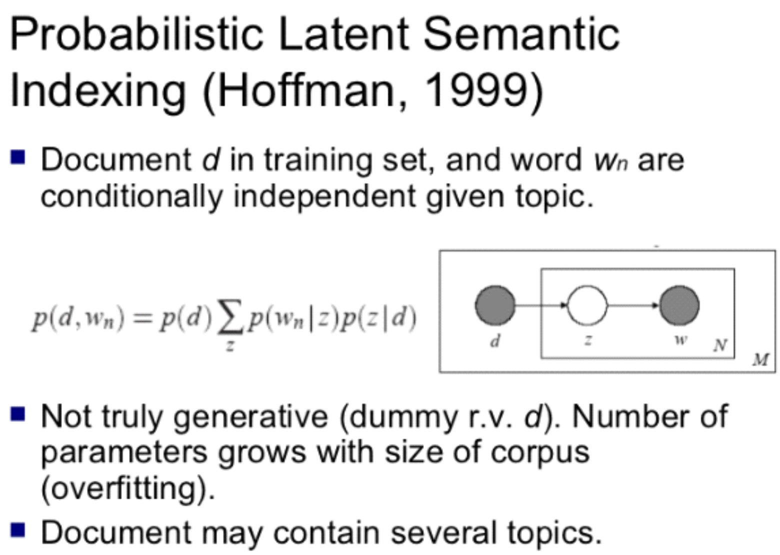
Unigram Mixture Model

- First choose topic z, then generate words conditionally independent given topic.

$$p(\mathbf{w}) = \sum_{z} p(z) \prod_{n=1}^{N} p(w_n | z)$$

i) First choose P(z)
ii) Then generated words that are 'conditionally independent' (meaning you can multiply the probabilities) given the topic--> P(word w |topic z)

Basic Idea of PLSI: p(word | topic) and p (topic | document) are conditionally independent



Probabilistic Latent Semantic Indexing (Hoffman, 1999)

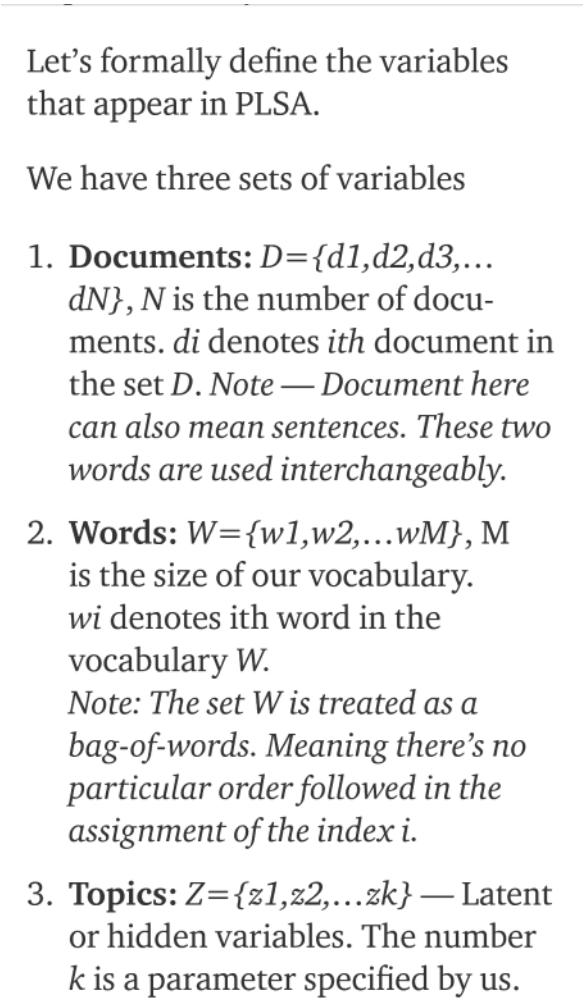
- Document *d* in training set, and word w_n are conditionally independent given topic.

$$p(d, w_n) = p(d) \sum_z p(w_n | z) p(z | d)$$

- Not truly generative (dummy r.v. *d*). Number of parameters grows with size of corpus (overfitting).
- Document may contain several topics.

Probabilistic Latent Semantic Indexing a.k.a Aspect Model:
Instead of doing the above LSI in the SVD way, it is done the probabilistic way

1. 1 hidden variable 2 observed variables

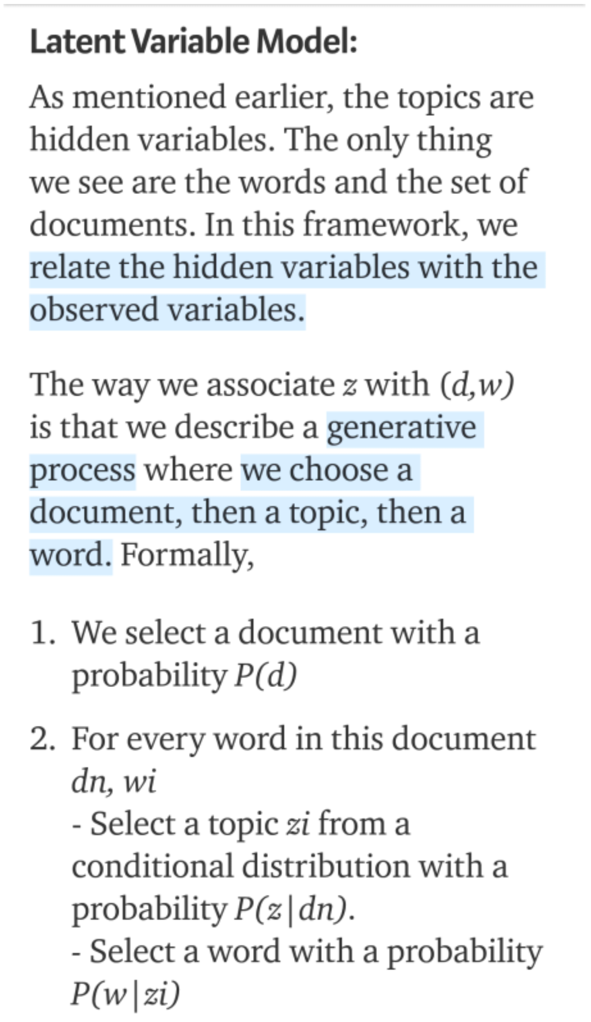


Let's formally define the variables that appear in PLSA.

We have three sets of variables

1. **Documents:** $D=\{d1,d2,d3,\ldots dN\}$, N is the number of documents. di denotes ith document in the set D. *Note — Document here can also mean sentences. These two words are used interchangeably.*
2. **Words:** $W=\{w1,w2,\ldots wM\}$, M is the size of our vocabulary. wi denotes ith word in the vocabulary W.
   Note: The set W is treated as a bag-of-words. Meaning there's no particular order followed in the assignment of the index i.
3. **Topics:** $Z=\{z1,z2,\ldots zk\}$ — Latent or hidden variables. The number k is a parameter specified by us.

2. Relationship of the hidden variable with observed variables



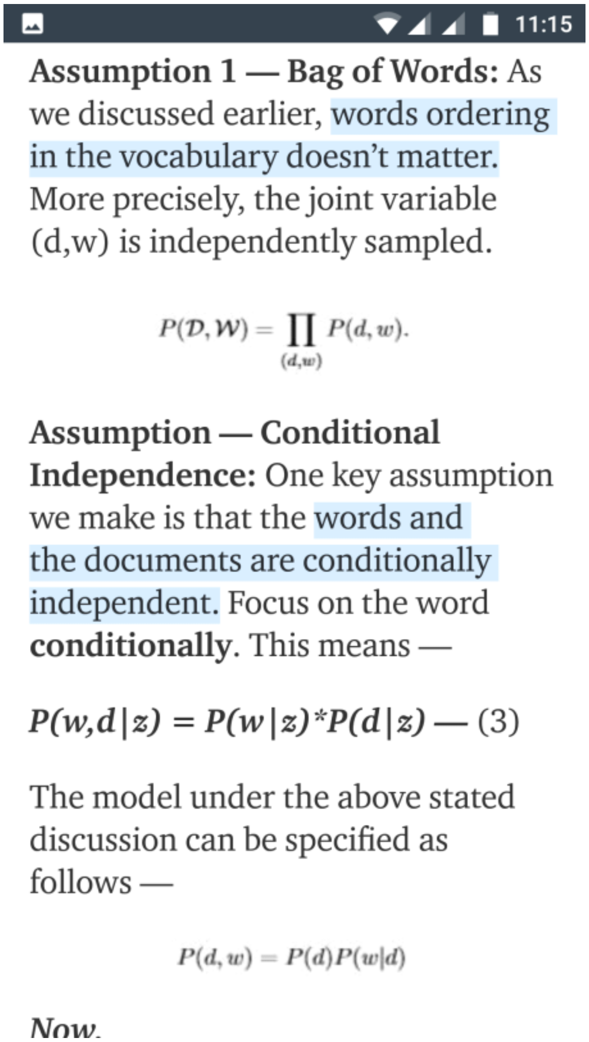
Latent Variable Model:

As mentioned earlier, the topics are hidden variables. The only thing we see are the words and the set of documents. In this framework, we relate the hidden variables with the observed variables.

The way we associate z with (d,w) is that we describe a generative process where we choose a document, then a topic, then a word. Formally,

1. We select a document with a probability $P(d)$
2. For every word in this document dn, wi
   - Select a topic zi from a conditional distribution with a probability $P(z|dn)$.
   - Select a word with a probability $P(w|zi)$

3. Bag of Words and Conditional Independence



Assumption 1 — Bag of Words: As we discussed earlier, words ordering in the vocabulary doesn't matter. More precisely, the joint variable (d,w) is independently sampled.

$$P(\mathcal{D},\mathcal{W}) = \prod_{(d,w)} P(d,w).$$

Assumption — Conditional Independence: One key assumption we make is that the words and the documents are conditionally independent. Focus on the word **conditionally**. This means —

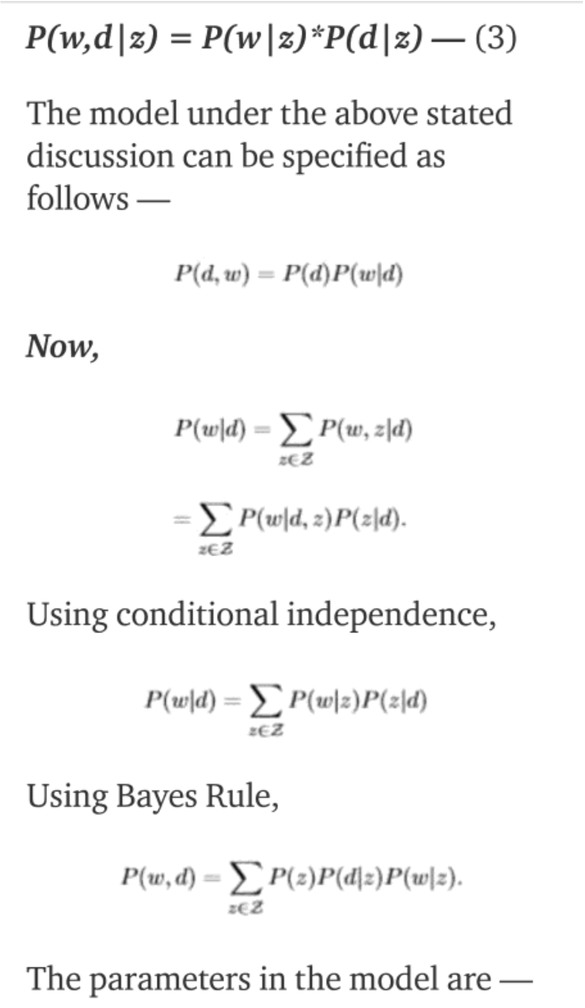
P(w,d|z) = P(w|z)*P(d|z) — (3)

The model under the above stated discussion can be specified as follows —

$$P(d,w) = P(d)P(w|d)$$

Now.

4. Joint prob distribution - Main equation



P(w,d|z) = P(w|z)*P(d|z) — (3)

The model under the above stated discussion can be specified as follows —

$$P(d,w) = P(d)P(w|d)$$

Now,

$$P(w|d) = \sum_{z \in Z} P(w,z|d)$$

$$= \sum_{z \in Z} P(w|d,z)P(z|d).$$

Using conditional independence,

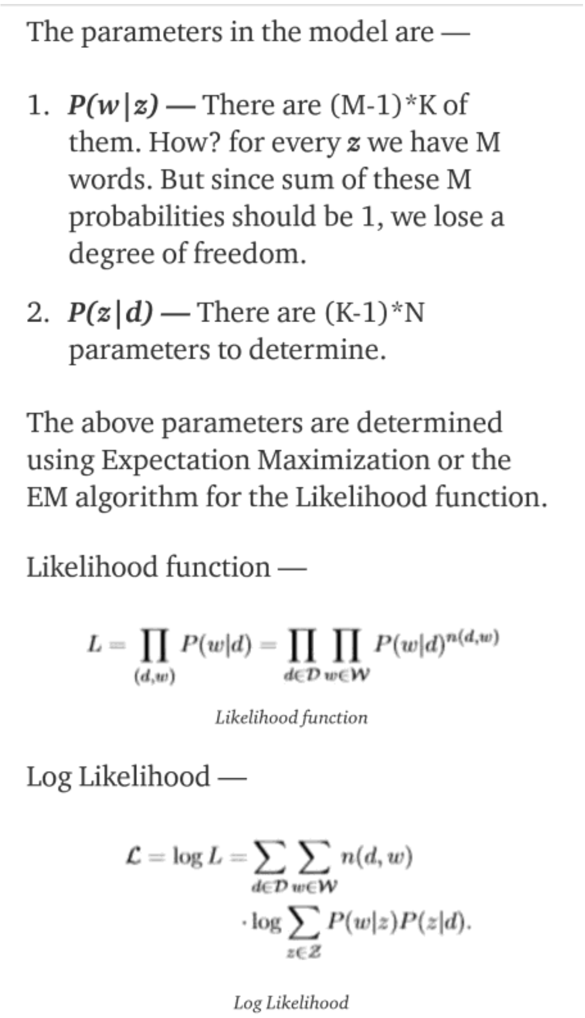
$$P(w|d) = \sum_{z \in Z} P(w|z)P(z|d)$$

Using Bayes Rule,

$$P(w,d) = \sum_{z \in Z} P(z)P(d|z)P(w|z).$$

The parameters in the model are —

5. What are the parameters and what is the objective function



The parameters in the model are —

1. ***P(w|z)* —** There are (M-1)*K of them. How? for every ***z*** we have M words. But since sum of these M probabilities should be 1, we lose a degree of freedom.
2. ***P(z|d)* —** There are (K-1)*N parameters to determine.

The above parameters are determined using Expectation Maximization or the EM algorithm for the Likelihood function.

Likelihood function —

$$L = \prod_{(d,w)} P(w|d) = \prod_{d\in\mathcal{D}} \prod_{w\in\mathcal{W}} P(w|d)^{n(d,w)}$$

Likelihood function

Log Likelihood —

$$\mathcal{L} = \log L = \sum_{d\in\mathcal{D}} \sum_{w\in\mathcal{W}} n(d,w) \cdot \log \sum_{z\in\mathcal{Z}} P(w|z)P(z|d).$$

Log Likelihood

What is the difference between MLE and EM?
MLE provides the Objective function that requires to be optimised for the Given Data.

The Optimisation itself can be done in multiple different ways. EM (Expectation Maximisation) is one of the ways to optimise, all other types of optimisations can also be used.

In Short MLE Defines the Optimization objective while EM solves it in iterative fashion.

- Source: Quora Answer

How LDA is different from pLSI?

- There is **NO OBJECTIVE FUNCTION and NO OPTIMIZATION**
- One of the approaches: LDA uses 'Collapsed' Gibbs Sampling approach
  (other inference methods include: Variational Approximation, Monte Carlo Markov Chain, Expectation Propagation)

Gibbs sampling

Iterative **random** hard assignment!

Benefits:

- Typically intuitive updates
- Very straightforward to implement

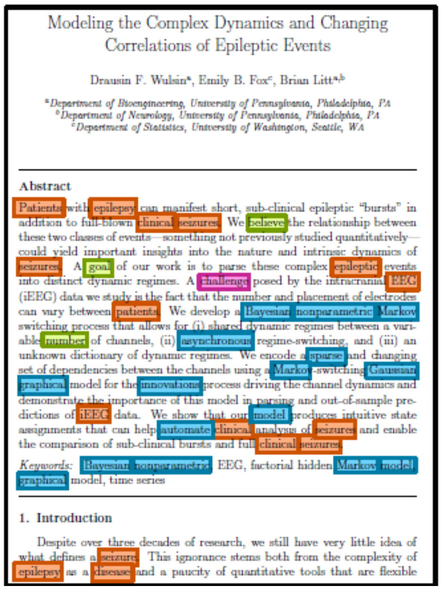
Random sample #10000

TOPIC 1	
experiment	0.1
test	0.08
discover	0.05
hypothesize	0.03
climate	0.01
...	...

TOPIC 2	
develop	0.18
computer	0.09
processor	0.032
user	0.027
internet	0.02
...	...

TOPIC 3	
player	0.15
score	0.07
team	0.06
goal	0.03
injury	0.01
...	...

⋮



Modeling the Complex Dynamics and Changing Correlations of Epileptic Events

Drausin F. Wulsin[a], Emily B. Fox[c], Brian Litt[a,b]

[a]*Department of Bioengineering, University of Pennsylvania, Philadelphia, PA*
[b]*Department of Neurology, University of Pennsylvania, Philadelphia, PA*
[c]*Department of Statistics, University of Washington, Seattle, WA*

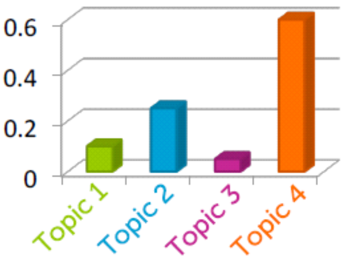
Abstract

Patients with epilepsy can manifest short, sub-clinical epileptic "bursts" in addition to full-blown clinical seizures. We believe the relationship between these two classes of events—something not previously studied quantitatively—could yield important insights into the nature and intrinsic dynamics of seizures. A goal of our work is to parse these complex epileptic events into distinct dynamic regimes. A challenge posed by the intracranial EEG (iEEG) data we study is the fact that the number and placement of electrodes can vary between patients. We develop a Bayesian nonparametric Markov switching process that allows for (i) shared dynamic regimes between a variable number of channels, (ii) asynchronous regime-switching, and (iii) an unknown dictionary of dynamic regimes. We encode a sparse and changing set of dependencies between the channels using a Markov-switching Gaussian graphical model for the innovations process driving the channel dynamics and demonstrate the importance of this model in parsing and out-of-sample predictions of iEEG data. We show that our model produces intuitive state assignments that can help automate clinical analysis of seizures and enable the comparison of sub-clinical bursts and full clinical seizures.

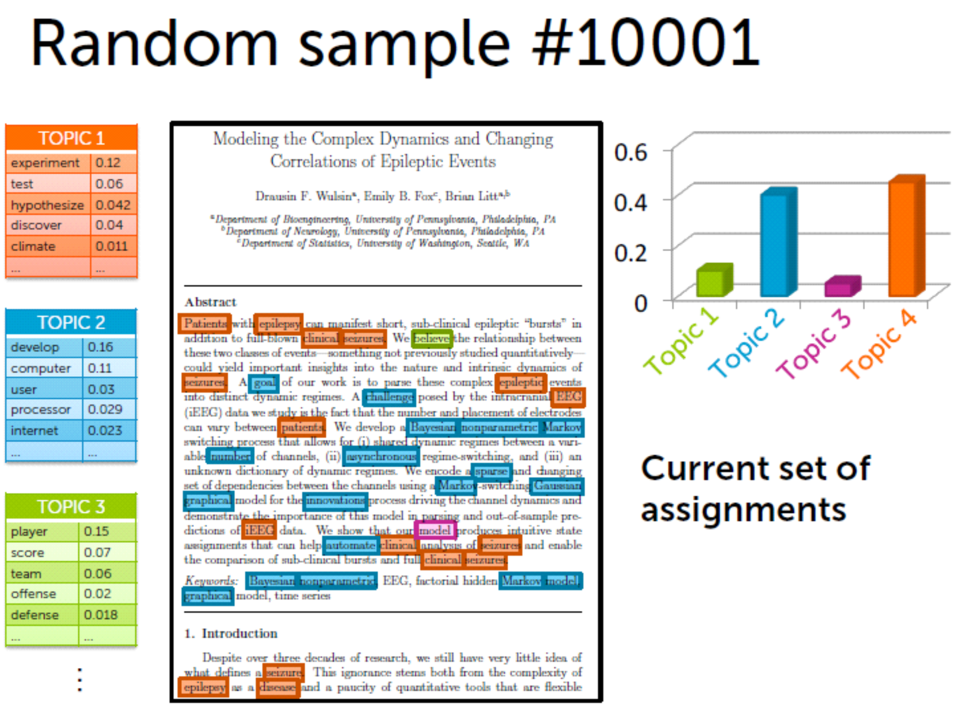
Keywords: Bayesian nonparametric, EEG, factorial hidden Markov model, graphical model, time series

1. Introduction

Despite over three decades of research, we still have very little idea of what defines a seizure. This ignorance stems both from the complexity of epilepsy as a disease and a paucity of quantitative tools that are flexible



Current set of assignments



Random sample #10001

TOPIC 1	
experiment	0.12
test	0.06
hypothesize	0.042
discover	0.04
climate	0.011
...	...

TOPIC 2	
develop	0.16
computer	0.11
user	0.03
processor	0.029
internet	0.023
...	...

TOPIC 3	
player	0.15
score	0.07
team	0.06
offense	0.02
defense	0.018
...	...

⋮

Modeling the Complex Dynamics and Changing Correlations of Epileptic Events

Drausin F. Wulsin[a], Emily B. Fox[c], Brian Litt[a,b]

[a]Department of Bioengineering, University of Pennsylvania, Philadelphia, PA
[b]Department of Neurology, University of Pennsylvania, Philadelphia, PA
[c]Department of Statistics, University of Washington, Seattle, WA

Abstract

Patients with epilepsy can manifest short, sub-clinical epileptic "bursts" in addition to full-blown clinical seizures. We believe the relationship between these two classes of events—something not previously studied quantitatively—could yield important insights into the nature and intrinsic dynamics of seizures. A goal of our work is to parse these complex epileptic events into distinct dynamic regimes. A challenge posed by the intracranial EEG (iEEG) data we study is the fact that the number and placement of electrodes can vary between patients. We develop a Bayesian nonparametric Markov switching process that allows for (i) shared dynamic regimes between a variable number of channels, (ii) asynchronous regime-switching, and (iii) an unknown dictionary of dynamic regimes. We encode a sparse and changing set of dependencies between the channels using a Markov-switching Gaussian graphical model for the innovations process driving the channel dynamics and demonstrate the importance of this model in parsing and out-of-sample predictions of iEEG data. We show that our model produces intuitive state assignments that can help automate clinical analysis of seizures and enable the comparison of sub-clinical bursts and full clinical seizures.

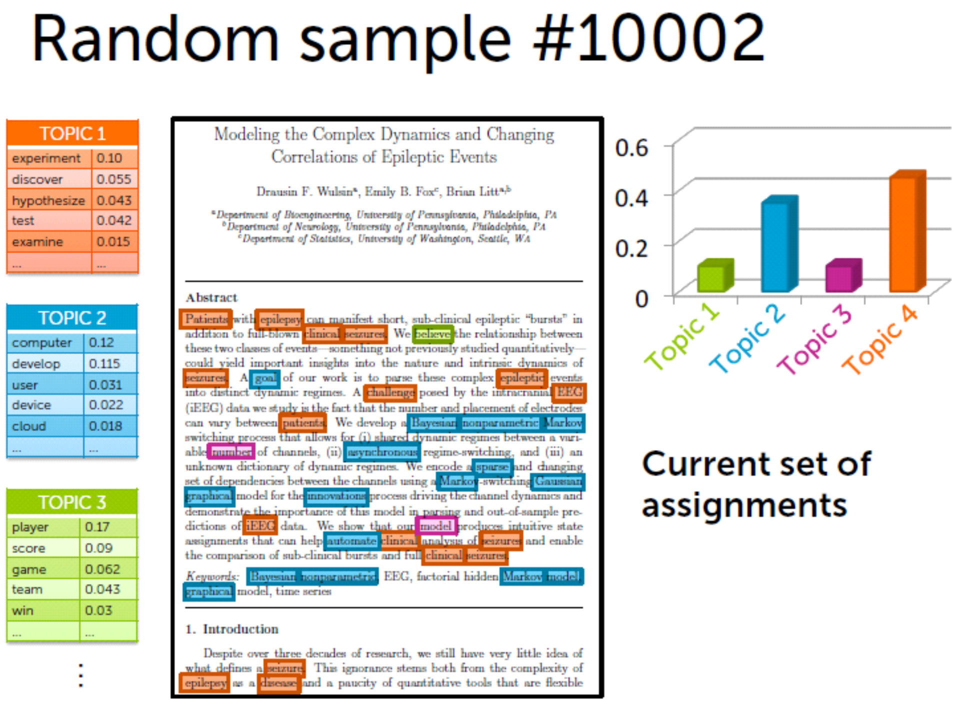
Keywords: Bayesian nonparametric, EEG, factorial hidden Markov model, graphical model, time series

1. Introduction

Despite over three decades of research, we still have very little idea of what defines a seizure. This ignorance stems both from the complexity of epilepsy as a disease and a paucity of quantitative tools that are flexible

Current set of assignments

Screen clipping taken: 2/3/2019 7:30 PM



Random sample #10002

TOPIC 1	
experiment	0.10
discover	0.055
hypothesize	0.043
test	0.042
examine	0.015
...	...

TOPIC 2	
computer	0.12
develop	0.115
user	0.031
device	0.022
cloud	0.018
...	...

TOPIC 3	
player	0.17
score	0.09
game	0.062
team	0.043
win	0.03
...	...

⋮

Modeling the Complex Dynamics and Changing Correlations of Epileptic Events

Drausin F. Wulsin[a], Emily B. Fox[c], Brian Litt[a,b]

[a]Department of Bioengineering, University of Pennsylvania, Philadelphia, PA
[b]Department of Neurology, University of Pennsylvania, Philadelphia, PA
[c]Department of Statistics, University of Washington, Seattle, WA

Abstract

Patients with epilepsy can manifest short, sub-clinical epileptic "bursts" in addition to full-blown clinical seizures. We believe the relationship between these two classes of events—something not previously studied quantitatively—could yield important insights into the nature and intrinsic dynamics of seizures. A goal of our work is to parse these complex epileptic events into distinct dynamic regimes. A challenge posed by the intracranial EEG (iEEG) data we study is the fact that the number and placement of electrodes can vary between patients. We develop a Bayesian nonparametric Markov switching process that allows for (i) shared dynamic regimes between a variable number of channels, (ii) asynchronous regime-switching, and (iii) an unknown dictionary of dynamic regimes. We encode a sparse and changing set of dependencies between the channels using a Markov-switching Gaussian graphical model for the innovations process driving the channel dynamics and demonstrate the importance of this model in parsing and out-of-sample predictions of iEEG data. We show that our model produces intuitive state assignments that can help automate clinical analysis of seizures and enable the comparison of sub-clinical bursts and full clinical seizures.

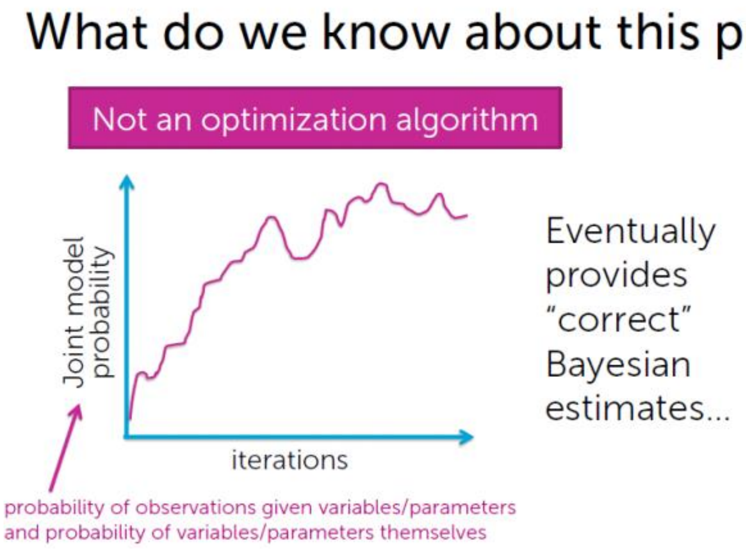
Keywords: Bayesian nonparametric, EEG, factorial hidden Markov model, graphical model, time series

1. Introduction

Despite over three decades of research, we still have very little idea of what defines a seizure. This ignorance stems both from the complexity of epilepsy as a disease and a paucity of quantitative tools that are flexible

Current set of assignments

Screen clipping taken: 2/3/2019 7:30 PM



What do we know about this process?

Not an optimization algorithm

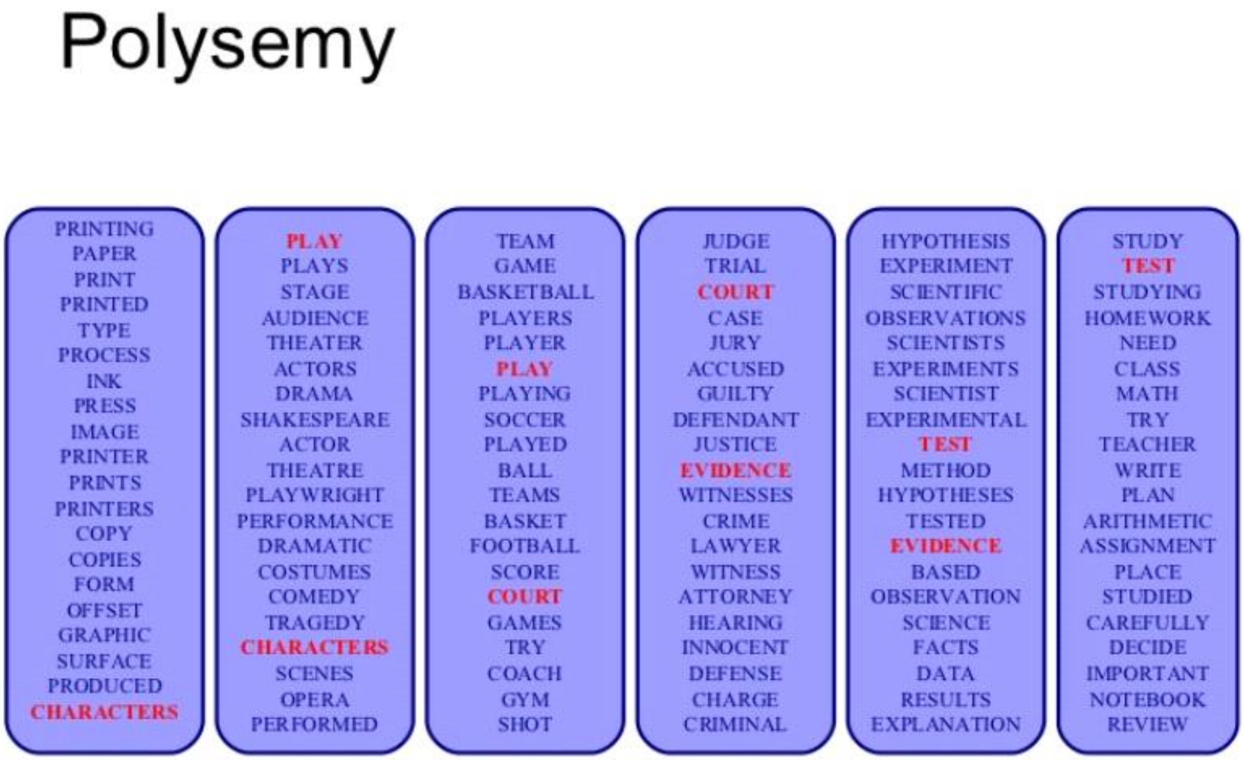
Eventually provides "correct" Bayesian estimates...

probability of observations given variables/parameters and probability of variables/parameters themselves

-- We will see in detail in the next section

How is LDA better than PLSI
https://www.quora.com/What-are-the-reasons-to-choose-LDA-over-pLSA-or-vice-versa

i) PLSI - CANNOT scale as good as LDA. Number of parameters increase with size of corpus
ii) LDA can handle polysemy. Look at the example 'Court' - tennis court or judicial court



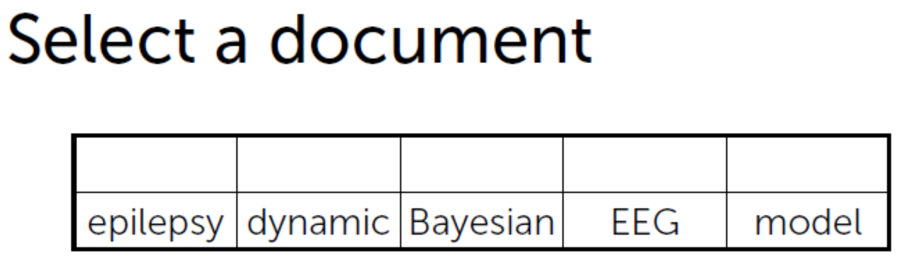
Polysemy

PRINTING	PLAY	TEAM	JUDGE	HYPOTHESIS	STUDY
PAPER	PLAYS	GAME	TRIAL	EXPERIMENT	TEST
PRINT	STAGE	BASKETBALL	COURT	SCIENTIFIC	STUDYING
PRINTED	AUDIENCE	PLAYERS	CASE	OBSERVATIONS	HOMEWORK
TYPE	THEATER	PLAYER	JURY	SCIENTISTS	NEED
PROCESS	ACTORS	PLAY	ACCUSED	EXPERIMENTS	CLASS
INK	DRAMA	PLAYING	GUILTY	SCIENTIST	MATH
PRESS	SHAKESPEARE	SOCCER	DEFENDANT	EXPERIMENTAL	TRY
IMAGE	ACTOR	PLAYED	JUSTICE	TEST	TEACHER
PRINTER	THEATRE	BALL	EVIDENCE	METHOD	WRITE
PRINTS	PLAYWRIGHT	TEAMS	WITNESSES	HYPOTHESES	PLAN
PRINTERS	PERFORMANCE	BASKET	CRIME	TESTED	ARITHMETIC
COPY	DRAMATIC	FOOTBALL	LAWYER	EVIDENCE	ASSIGNMENT
COPIES	COSTUMES	SCORE	WITNESS	BASED	PLACE
FORM	COMEDY	COURT	ATTORNEY	OBSERVATION	STUDIED
OFFSET	TRAGEDY	GAMES	HEARING	SCIENCE	CAREFULLY
GRAPHIC	CHARACTERS	TRY	INNOCENT	FACTS	DECIDE
SURFACE	SCENES	COACH	DEFENSE	DATA	IMPORTANT
PRODUCED	OPERA	GYM	CHARGE	RESULTS	NOTEBOOK
CHARACTERS	PERFORMED	SHOT	CRIMINAL	EXPLANATION	REVIEW

3. Simplified Explanation for LDA

Saturday, February 02, 2019 9:37 PM

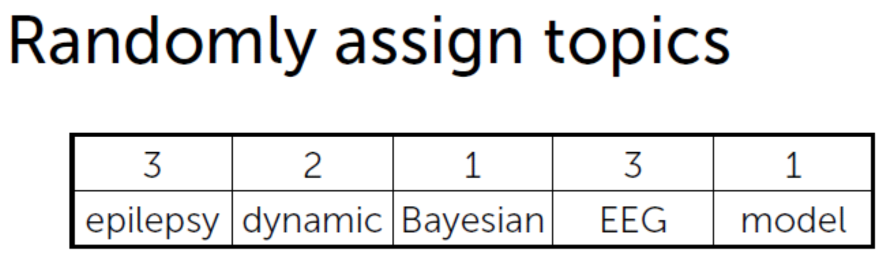
How LDA works - step by step:



Select a document

epilepsy	dynamic	Bayesian	EEG	model

5 word document

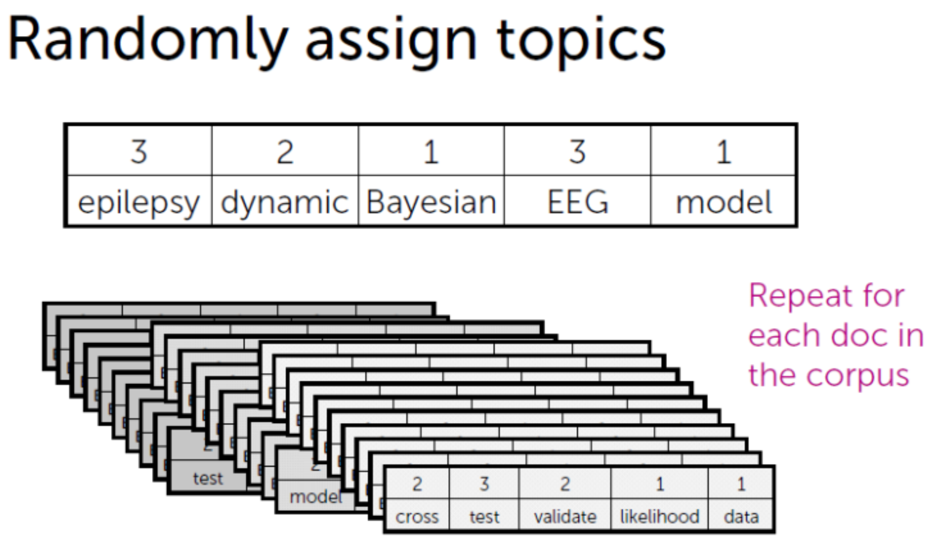


Randomly assign topics

3	2	1	3	1
epilepsy	dynamic	Bayesian	EEG	model

(one possible approach)

One approach is Collapsed Gibbs Sampling



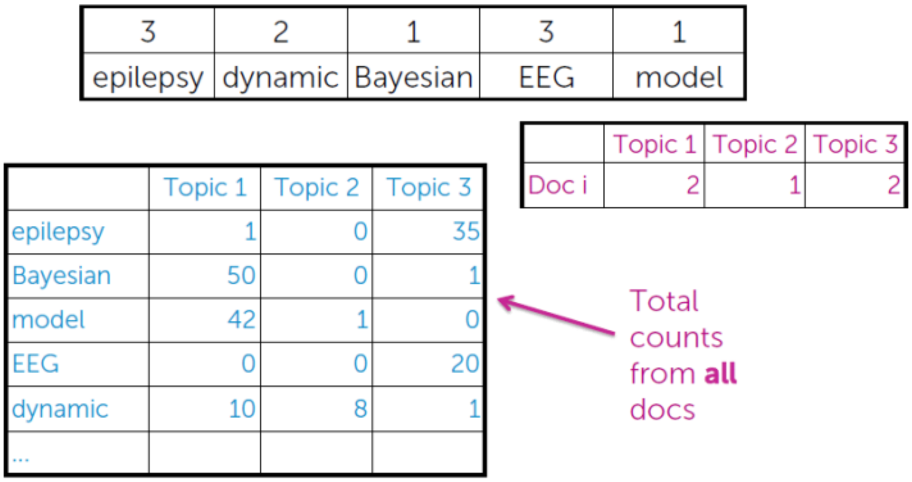
Randomly assign topics

3	2	1	3	1
epilepsy	dynamic	Bayesian	EEG	model

2	3	2	1	1
cross	test	validate	likelihood	data

Repeat for each doc in the corpus

Prepare Topic - Word Matrix and Doc - Topic Matrix



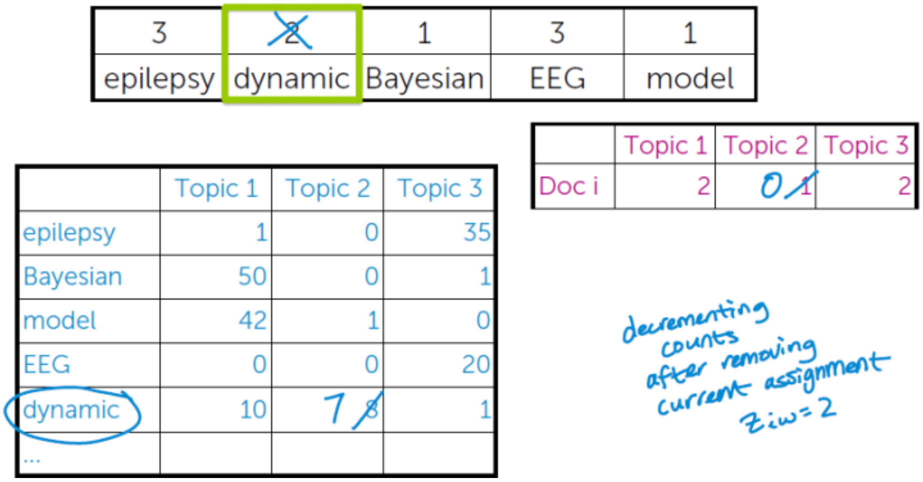
3	2	1	3	1
epilepsy	dynamic	Bayesian	EEG	model

	Topic 1	Topic 2	Topic 3
Doc i	2	1	2

	Topic 1	Topic 2	Topic 3
epilepsy	1	0	35
Bayesian	50	0	1
model	42	1	0
EEG	0	0	20
dynamic	10	8	1
…			

Total counts from **all** docs

Assume the topic assigned to the current word 'dynamic' is wrong (everything else is assumed correct)

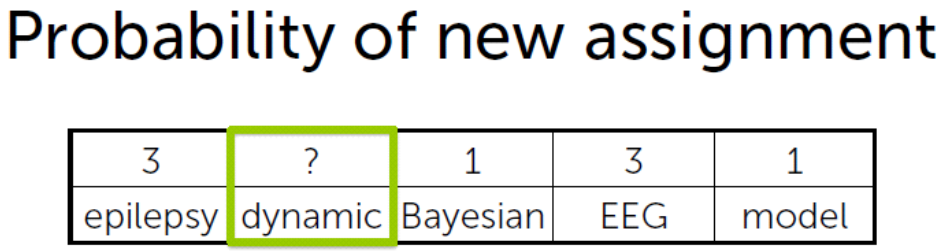


3	~~2~~	1	3	1
epilepsy	dynamic	Bayesian	EEG	model

	Topic 1	Topic 2	Topic 3
Doc i	2	0 ~~1~~	2

	Topic 1	Topic 2	Topic 3
epilepsy	1	0	35
Bayesian	50	0	1
model	42	1	0
EEG	0	0	20
dynamic	10	7 ~~8~~	1
…			

decrementing counts after removing current assignment $z_{iw}=2$



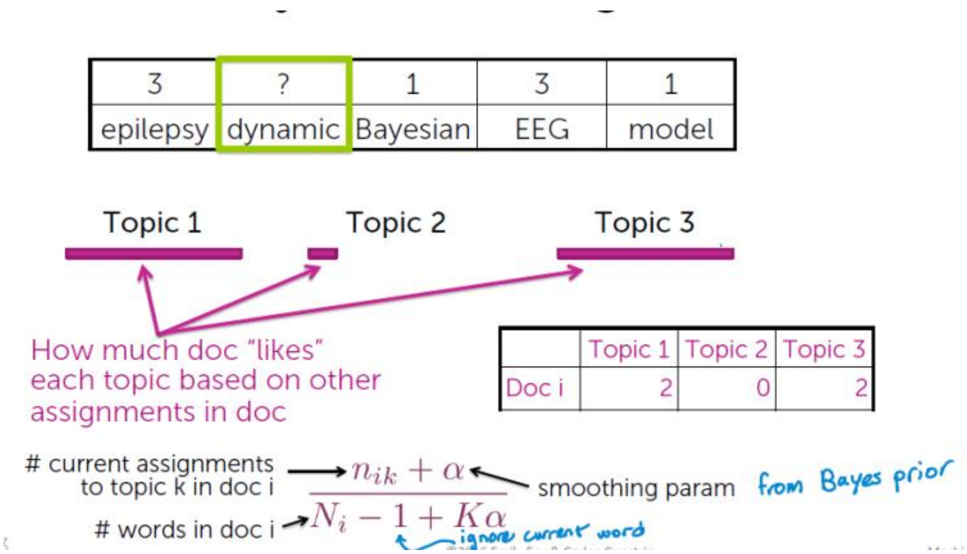
Probability of new assignment

3	?	1	3	1
epilepsy	dynamic	Bayesian	EEG	model

reassign with probability $p(z_{iw} \mid$ every other z_{jv} in corpus, words in corpus)

Calculate Prob (topic t | document d) for every topic 1, 2, 3 given doc i

- This tells us which topic is most liked by doc i



3	?	1	3	1
epilepsy	dynamic	Bayesian	EEG	model

How much doc "likes" each topic based on other assignments in doc

	Topic 1	Topic 2	Topic 3
Doc i	2	0	2

current assignments to topic k in doc i

$$\frac{n_{ik} + \alpha}{N_i - 1 + K\alpha}$$

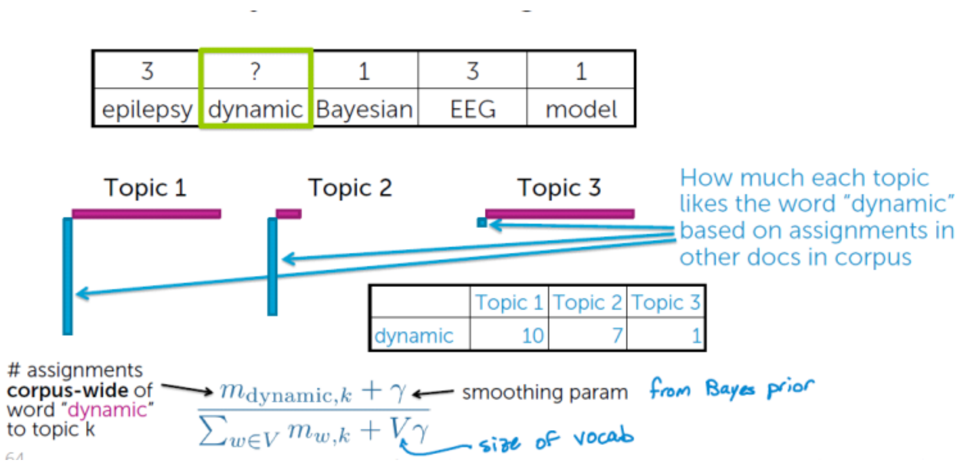
smoothing param from Bayes prior

words in doc i

ignore current word

Calculate prob (word w | topic t)

- How much each topic likes the word 'dynamic' is calculated



3	?	1	3	1
epilepsy	dynamic	Bayesian	EEG	model

How much each topic likes the word "dynamic" based on assignments in other docs in corpus

	Topic 1	Topic 2	Topic 3
dynamic	10	7	1

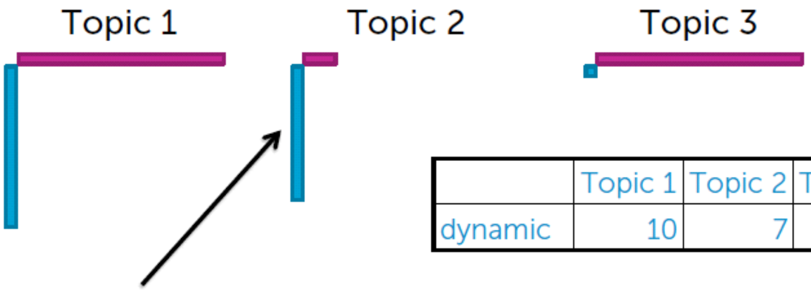
assignments **corpus-wide** of word "dynamic" to topic k

$$\frac{m_{\text{dynamic},k} + \gamma}{\sum_{w \in V} m_{w,k} + V\gamma}$$

smoothing param from Bayes prior

size of vocab

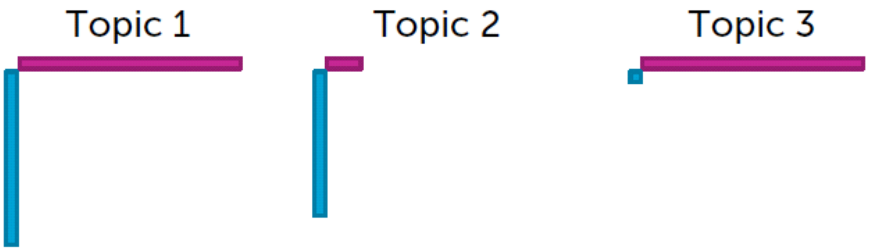
3	?	1	3	1
epilepsy	dynamic	Bayesian	EEG	model



	Topic 1	Topic 2	Topic 3
dynamic	10	7	1

Topic 2 also really likes “dynamic”,
but in a different context…
e.g., a topic on fluid dynamics

3	?	1	3	1
epilepsy	dynamic	Bayesian	EEG	model

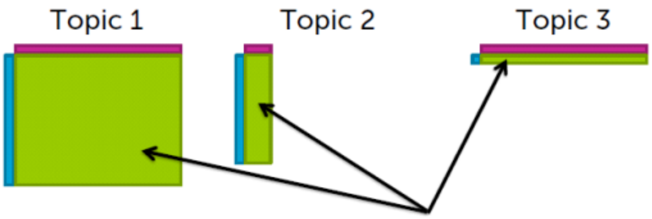


Topic fits word **and** document

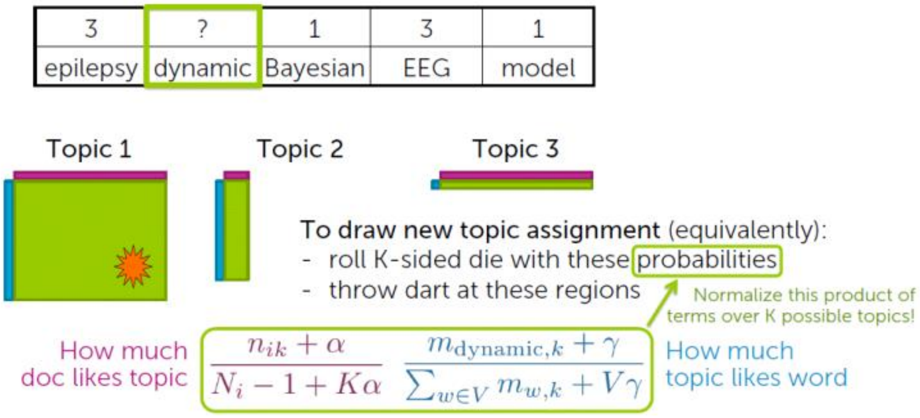
Topic fits word, but not doc

Topic fits doc, but not word

3	?	1	3	1
epilepsy	dynamic	Bayesian	EEG	model



How much doc likes topic $\frac{n_{ik}+\alpha}{N_i-1+K\alpha}$ $\frac{m_{\text{dynamic},k}+\gamma}{\sum_{w\in V} m_{w,k}+V\gamma}$ How much topic likes word



3	?	1	3	1
epilepsy	dynamic	Bayesian	EEG	model

To draw new topic assignment (equivalently):

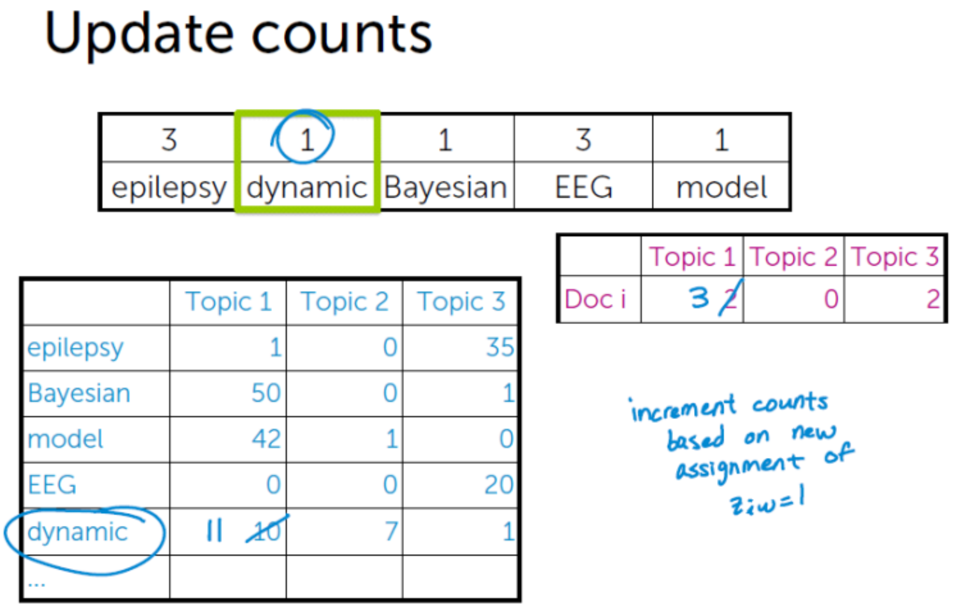
- roll K-sided die with these probabilities
- throw dart at these regions

Normalize this product of terms over K possible topics!

How much doc likes topic

$$\frac{n_{ik}+\alpha}{N_i-1+K\alpha}\;\frac{m_{\text{dynamic},k}+\gamma}{\sum_{w\in V} m_{w,k}+V\gamma}$$

How much topic likes word



Update counts

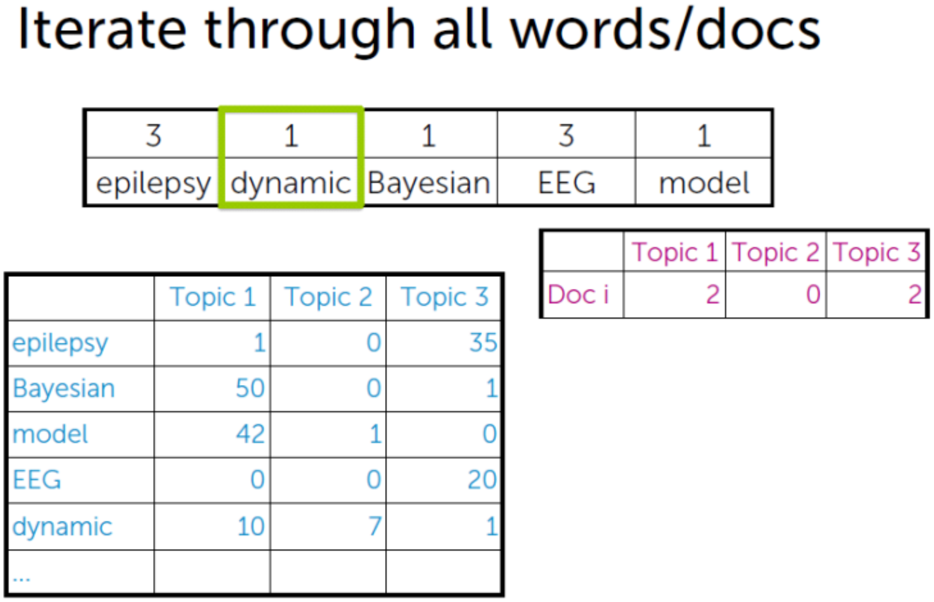
3	1	1	3	1
epilepsy	dynamic	Bayesian	EEG	model

	Topic 1	Topic 2	Topic 3
Doc i	3 ~~2~~	0	2

	Topic 1	Topic 2	Topic 3
epilepsy	1	0	35
Bayesian	50	0	1
model	42	1	0
EEG	0	0	20
dynamic	11 ~~10~~	7	1
...			

increment counts based on new assignment of $z_{iw}=1$

```
Repeat the above steps 'iterations' number of times
For each in doc d:
        For every in word w:
                For each_topic in topic t
```



Iterate through all words/docs

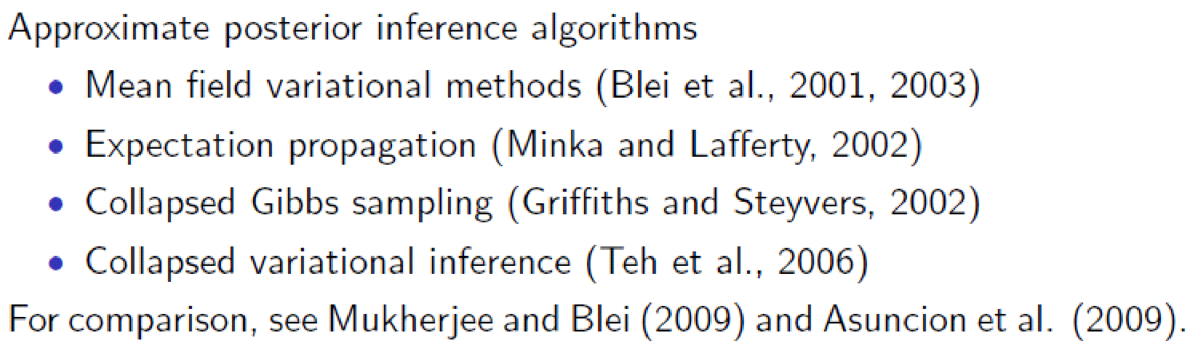
3	1	1	3	1
epilepsy	dynamic	Bayesian	EEG	model

	Topic 1	Topic 2	Topic 3
Doc i	2	0	2

	Topic 1	Topic 2	Topic 3
epilepsy	1	0	35
Bayesian	50	0	1
model	42	1	0
EEG	0	0	20
dynamic	10	7	1
...			

Eventually we will reach steady state

What we saw was Collapsed Gibbs Sampling. Other methods:



Approximate posterior inference algorithms

- Mean field variational methods (Blei et al., 2001, 2003)
- Expectation propagation (Minka and Lafferty, 2002)
- Collapsed Gibbs sampling (Griffiths and Steyvers, 2002)
- Collapsed variational inference (Teh et al., 2006)

For comparison, see Mukherjee and Blei (2009) and Asuncion et al. (2009).

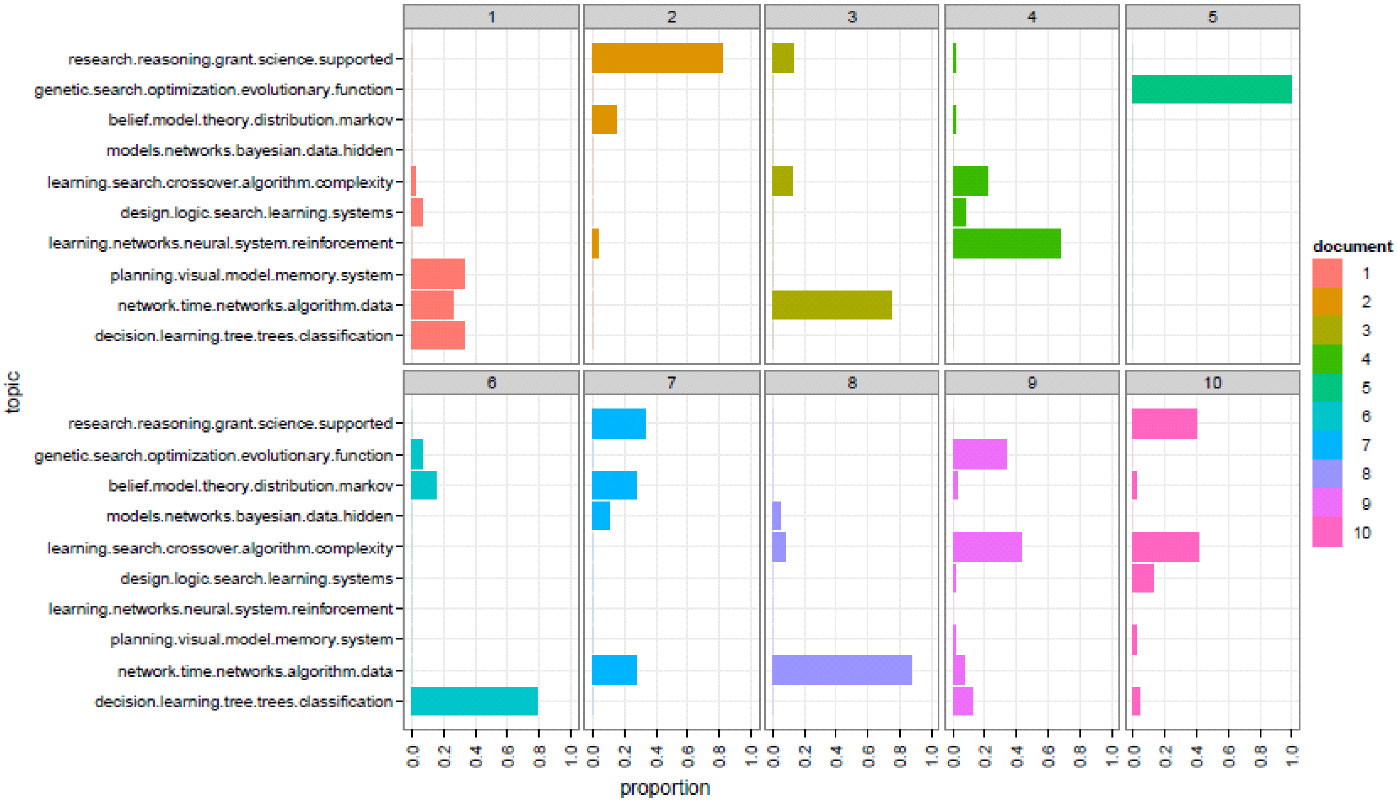
4. How do we implement LDA

Wednesday, February 06, 2019 12:43 PM

Before seeing what is LDA (the maths behind it), let us quickly see, what it can do

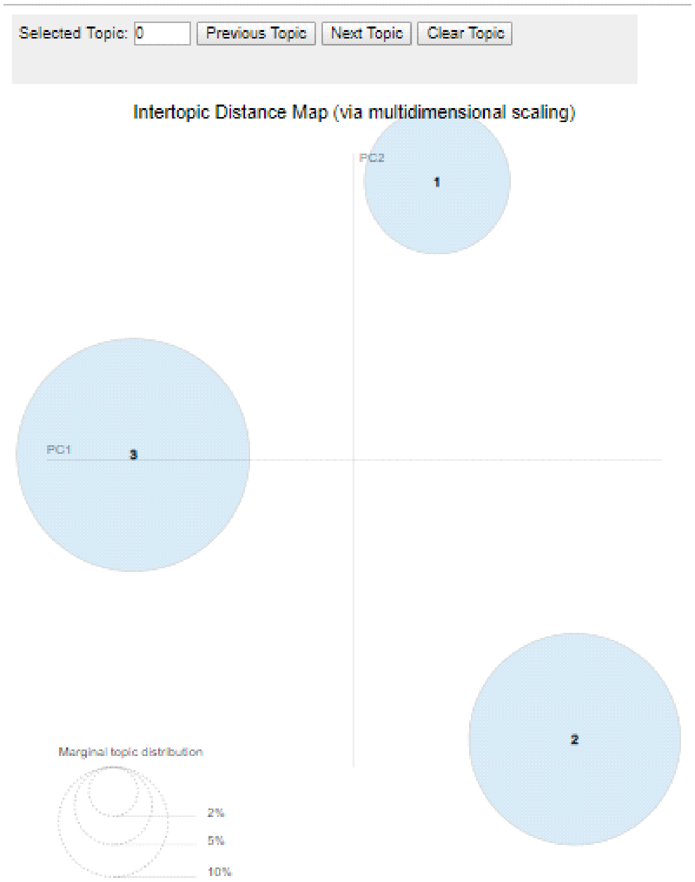
How LDA is implemented?

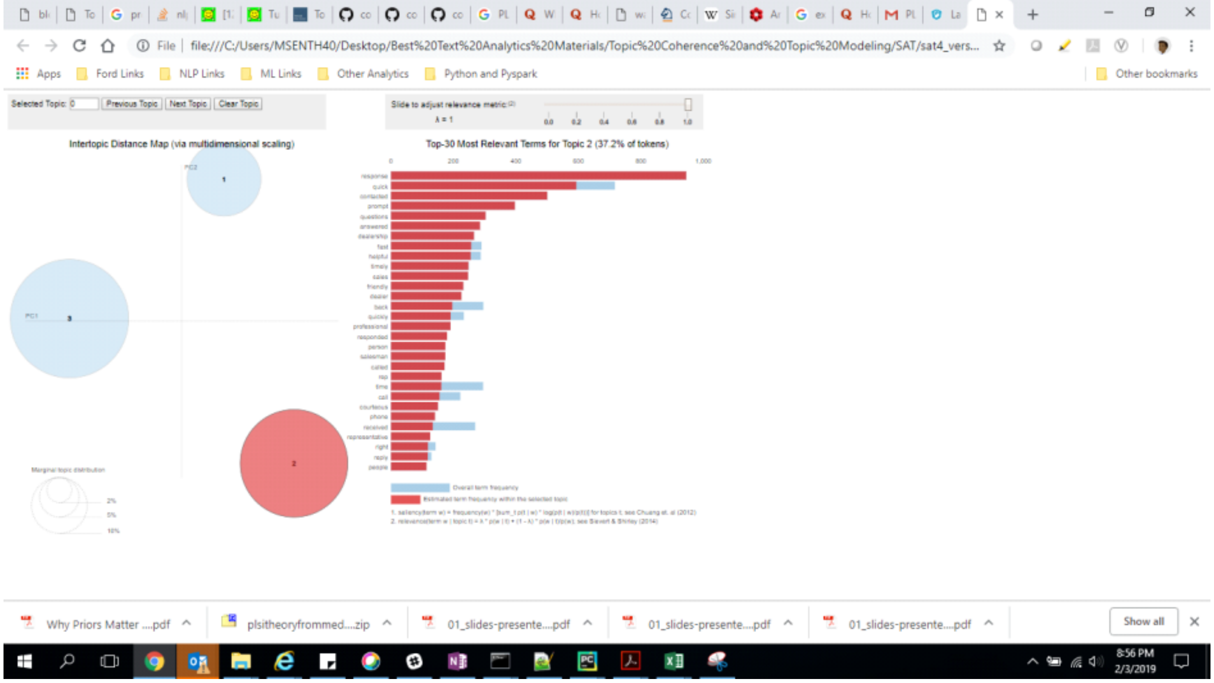
- **Preprocessing Data (stop-words removal, stemming)**
- **Prepare Term Document Matrix (identify**
- **Choose number of topics**
- **Run a topic model**
- **Observed Output:**



-

Topic_No	Topic_Name (with top words in that topic)	Count
3	good_ford_great_service_car_vehicle_experience_truck_excellent_price	7907
2	response_quick_contacted_prompt_questions_answered_dealership_fast_helpful_timely	7132
1	easy_information_quote_info_price_requested_contact_needed_online	3563


Selected Topic: 0
Previous Topic
Next Topic
Clear Topic
Intertopic Distance Map (via multidimensional scaling)
PC2
1
PC1
3
2
Marginal topic distribution
2%
5%
10%


file:///C:/Users/MSENTH40/Desktop/Best%20Text%20Analytics%20Materials/Topic%20Coherence%20and%20Topic%20Modeling/SAT/sat4_vers...
Apps
Ford Links
NLP Links
ML Links
Other Analytics
Python and Pyspark
Other bookmarks
Selected Topic: 0
Previous Topic
Next Topic
Clear Topic
Slide to adjust relevance metric:(2)
λ = 1
0.0
0.2
0.4
0.6
0.8
1.0
Intertopic Distance Map (via multidimensional scaling)
PC2
1
PC1
3
2
Marginal topic distribution
2%
5%
10%
Top-30 Most Relevant Terms for Topic 2 (37.2% of tokens)
0
200
400
600
800
1,000
response
quick
contacted
prompt
questions
answered
dealership
fast
helpful
timely
sales
friendly
dealer
back
quickly
professional
responded
person
salesman
called
rep
time
call
courteous
phone
received
representative
right
reply
people
Overall term frequency
Estimated term frequency within the selected topic
1. saliency(term w) = frequency(w) * [sum_t p(t | w) * log(p(t | w)/p(t))] for topics t; see Chuang et. al (2012)
2. relevance(term w | topic t) = λ * p(w | t) + (1 - λ) * p(w | t)/p(w); see Sievert & Shirley (2014)
Why Priors Matter ....pdf
plsitheoryfrommed....zip
01_slides-presente....pdf
01_slides-presente....pdf
01_slides-presente....pdf
Show all
8:56 PM
2/3/2019

5. The Maths behind LDA

Sunday, February 03, 2019 7:20 PM

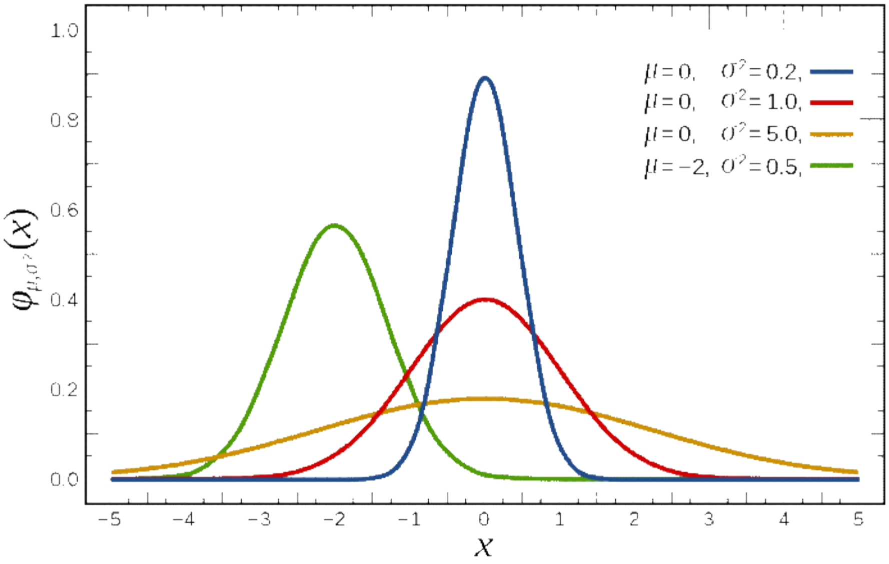
Latent: **Topic structures** in a document are latent meaning they are **hidden** structures in the text.
Dirichlet: The **Dirichlet distribution** determines the mixture proportions of

i) the topics in the documents and
ii) the words in each topic.

Allocation: Allocation of words to a given topic and allocation of topics to a document

Intuitive understanding of Dirichlet Allocation:
What is **Normal or Gaussian** Distribution:

- The normal distribution is a probability distribution over all the real numbers.
- It is described by a **mean** and a **variance**.
- The mean is the expected value of this distribution, and the variance tells us how much we can expect samples to deviate from the mean.
- If the variance is very high, then you’re going to see values that are both much smaller than the mean and much larger than the mean. If the variance is small, then the samples will be very close to the mean. If the variance goes close to zero, all samples will be almost exactly at the mean.



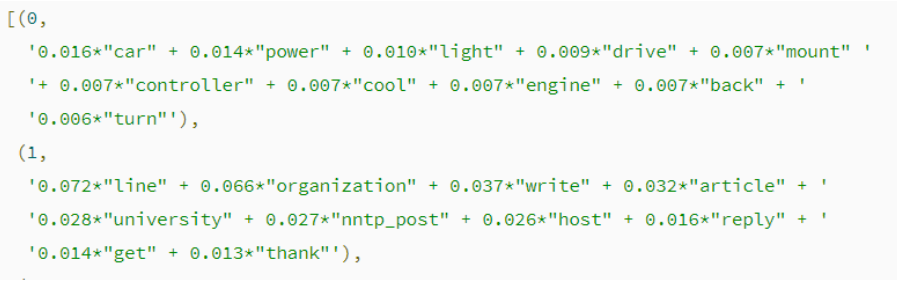
The **dirichlet distribution** is a probability distribution as well - but it is not sampling from the space of real numbers. Instead it is sampling over a probability simplex.

And what is a probability simplex? It’s a bunch of numbers that add up to 1. For example:

(0.6, 0.4)
(0.1, 0.1, 0.8)
(0.05, 0.2, 0.15, 0.1, 0.3, 0.2)

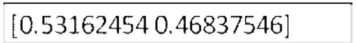
Both per-topic word distribution (almost infinite number of words) and per-document topic distribution (finite mixture of topics)

Per-topic Word Distribution is a simplex



```
[(0,
  '0.016*"car" + 0.014*"power" + 0.010*"light" + 0.009*"drive" + 0.007*"mount" '
  '+ 0.007*"controller" + 0.007*"cool" + 0.007*"engine" + 0.007*"back" + '
  '0.006*"turn"'),
 (1,
  '0.072*"line" + 0.066*"organization" + 0.037*"write" + 0.032*"article" + '
  '0.028*"university" + 0.027*"nntp_post" + 0.026*"host" + 0.016*"reply" + '
  '0.014*"get" + 0.013*"thank"'),
```

Per-document Topic Distribution is a simplex



```
[0.53162454 0.46837546]
```

Multinomial Distribution: Example:
If you roll a six-sided die, there is a chance for one out of 6 numbers to come up.

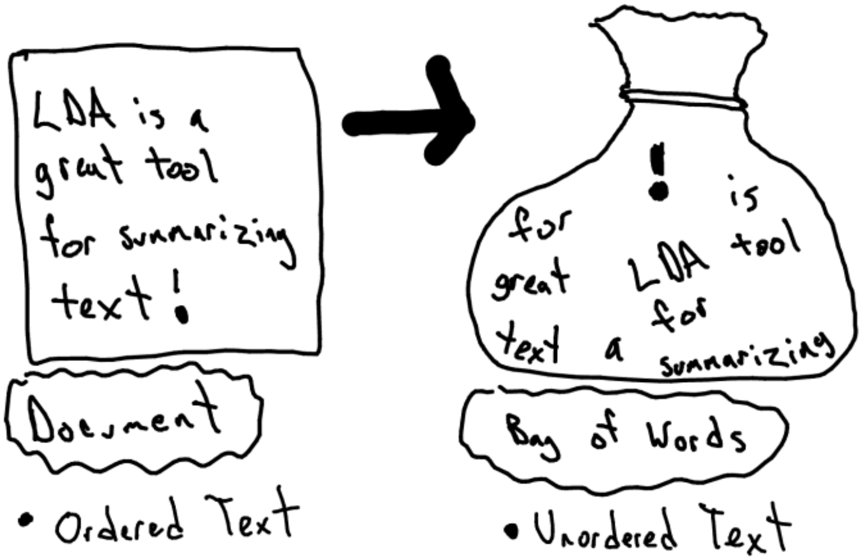
consider a chess match in which we have a certain probability of player A winning, players B winning, or the game ending in a draw.

Poisson Distribution:

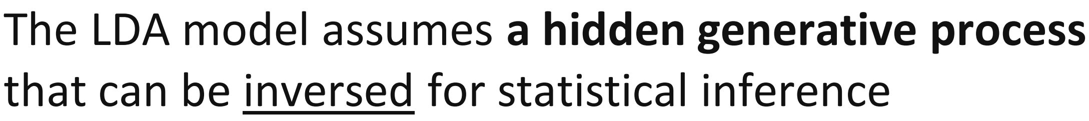
is a discrete probability distribution that expresses the probability of a given number of events occurring in a fixed interval

E.g.: The number of patients arriving in an emergency room between 10 and 11 pm

LDA processes documents as 'bag of words' -- ordering of words is not important


LDA is a great tool for summarizing text!
Document
• Ordered Text
! is for tool great LDA for text a summarizing
Bag of Words
• Unordered Text

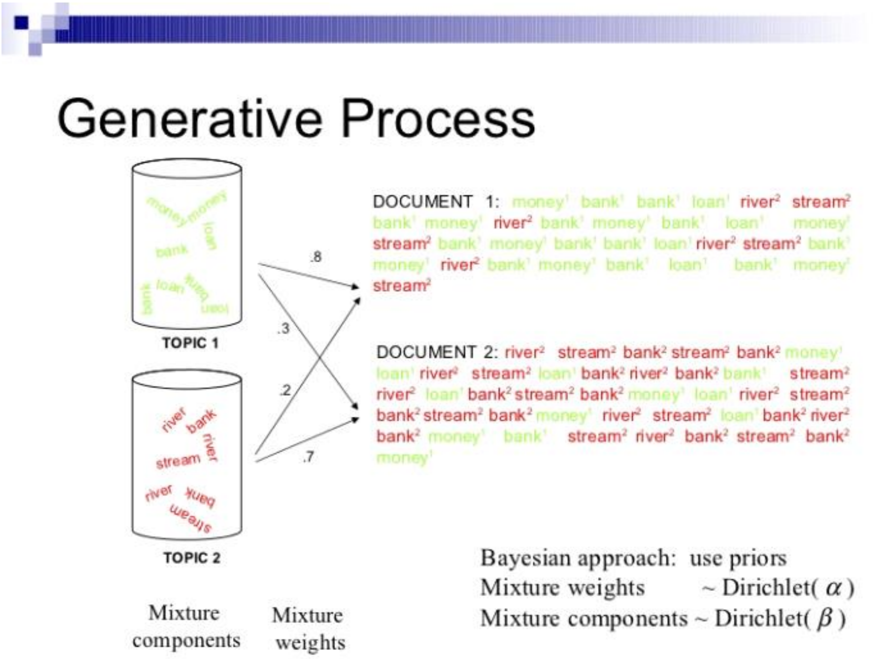
LDA is a generative model



The LDA model assumes **a hidden generative process** that can be inversed for statistical inference

From <https://www.researchgate.net/figure/The-LDA-model-assumes-a-hidden-generative-process-that-can-be-inversed-for-statistical_fig6_51109937>

In principle, LDA generates a document based on dirichlet distribution of topics over documents and words over topics



Generative Process

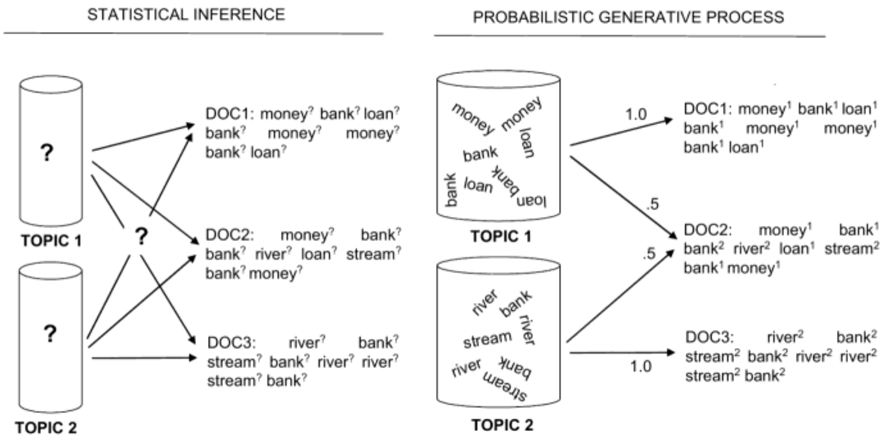
DOCUMENT 1: money^{1} bank^{1} bank^{1} loan^{1} river^{2} stream^{2} bank^{1} money^{1} river^{2} bank^{1} money^{1} bank^{1} loan^{1} money^{1} stream^{2} bank^{1} money^{1} bank^{1} bank^{1} loan^{1} river^{2} stream^{2} bank^{1} money^{1} river^{2} bank^{1} money^{1} bank^{1} loan^{1} bank^{1} money^{1} stream^{2}

DOCUMENT 2: river^{2} stream^{2} bank^{2} stream^{2} bank^{2} money^{1} loan^{1} river^{2} stream^{2} loan^{1} bank^{2} river^{2} bank^{2} bank^{1} stream^{2} river^{2} loan^{1} bank^{2} stream^{2} bank^{2} money^{1} loan^{1} river^{2} stream^{2} bank^{2} stream^{2} bank^{2} money^{1} river^{2} stream^{2} loan^{1} bank^{2} river^{2} bank^{2} money^{1} bank^{1} stream^{2} river^{2} bank^{2} stream^{2} bank^{2} money^{1}

Bayesian approach: use priors
Mixture weights ~ Dirichlet(α)
Mixture components ~ Dirichlet(β)

LDA is generative which means that they model texts as if they were generated from a certain probability distribution

What really happens, we inverse the generative process for statistical inference.


STATISTICAL INFERENCE
?
TOPIC 1
?
?
TOPIC 2
DOC1: money? bank? loan? bank? money? money? bank? loan?
DOC2: money? bank? bank? river? loan? stream? bank? money?
DOC3: river? bank? stream? bank? river? river? stream? bank?
PROBABILISTIC GENERATIVE PROCESS
money
money
loan
bank
loan
bank
bank
loan
TOPIC 1
river
bank
river
stream
river
bank
stream
TOPIC 2
1.0
.5
.5
1.0
DOC1: money¹ bank¹ loan¹ bank¹ money¹ money¹ bank¹ loan¹
DOC2: money¹ bank¹ bank² river² loan¹ stream² bank¹ money¹
DOC3: river² bank² stream² bank² river² river² stream² bank²

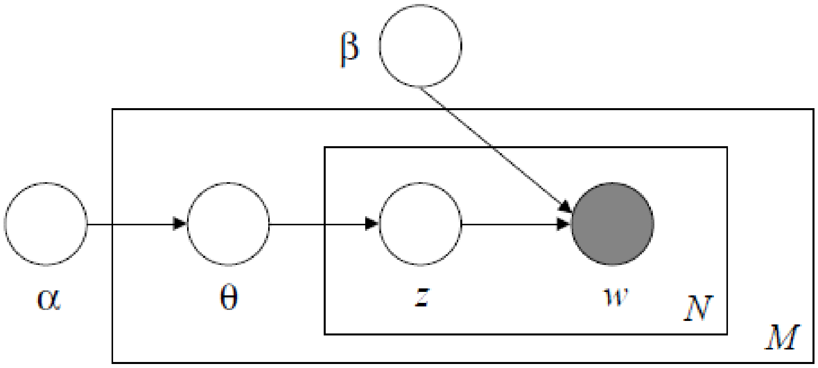
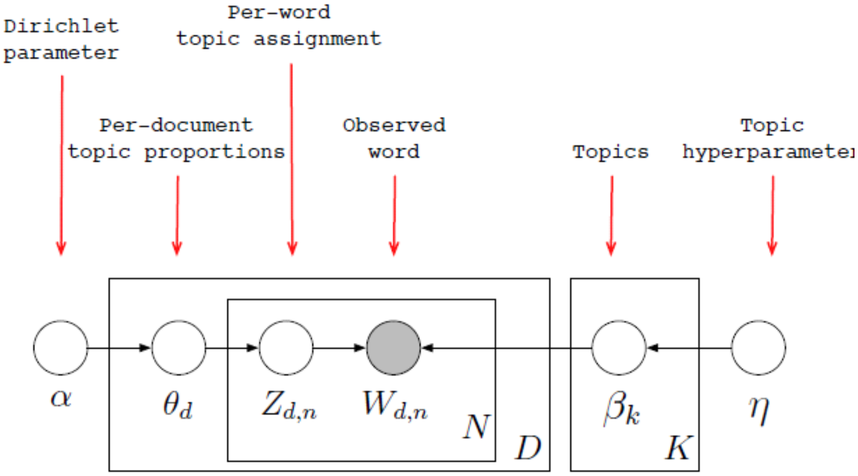


Figure 1: Graphical model representation of LDA. The boxes are “plates” representing replicates. The outer plate represents documents, while the inner plate represents the repeated choice of topics and words within a document.

A more understandable plate notation:



D = Total No. of Documents
N = No. of Words = Vocab Size
K = No. of Topics

θd = Topic Distribution for a particular document d

$$\theta = [topic\ a = 0.5,\ \ topic\ b = 0.5]$$

Φt= Word Distribution for a topic t. Here for topic 1 and 2.

ϕ_1 = [🟥 = 0.8, 🟦 = 0.2, 🟩 = 0.0]

ϕ_2 = [🟥 = 0.2, 🟦 = 0.1, 🟩 = 0.7]

(colored books represent words/tokens)

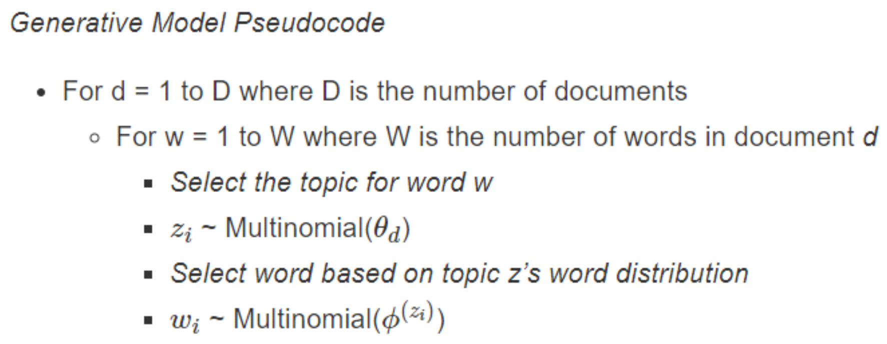
Zd,n = Topic Distribution for n th word in document d

Wd,n = nth word in dth document

α= parameter that sets the dircihlet prior on the per-document topic distribution (θ)
= parameter that represents the doc-topic density
= determines the no. of topics in each doc
= (Default) = 1/num_of_topics (in sklearn and gensim)
decreasing alpha results in less number of topics per document

β= parameter that sets the dirichlet prior on the per-topic word distribution (ϕ)
= parameter that represents the topic-word density
= determines the no. of words per each topic
= (Default) = 1/num_of_topics (in sklearn and gensim)
decreasing beta results in mutually exclusive words in topics

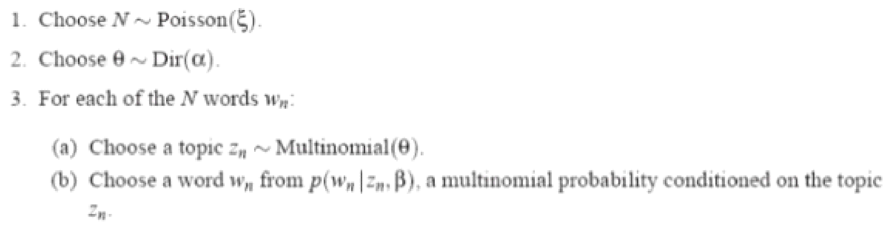
η= a hyper parameter to determine the number of topics (for ex. Topic Cohernce Score, Perplexity)



Generative Model Pseudocode

```
• For d = 1 to D where D is the number of documents
    ◦ For w = 1 to W where W is the number of words in document d
        ▪ Select the topic for word w
        ▪ $z_i$ ~ Multinomial($\theta_d$)
        ▪ Select word based on topic z's word distribution
        ▪ $w_i$ ~ Multinomial($\phi^{(z_i)}$)
```

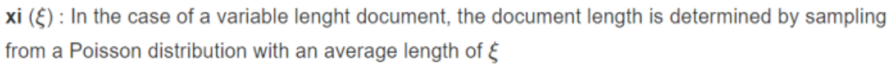
A more mathematical Logic flow:



1. Choose $N \sim \text{Poisson}(\xi)$.
2. Choose $\theta \sim \text{Dir}(\alpha)$.
3. For each of the N words w_n:

    (a) Choose a topic $z_n \sim \text{Multinomial}(\theta)$.

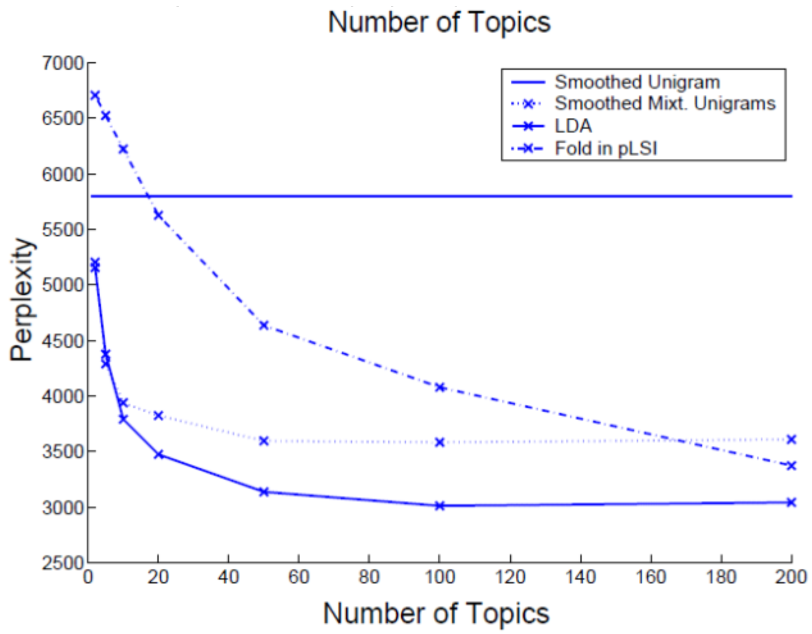
    (b) Choose a word w_n from $p(w_n | z_n, \beta)$, a multinomial probability conditioned on the topic z_n.



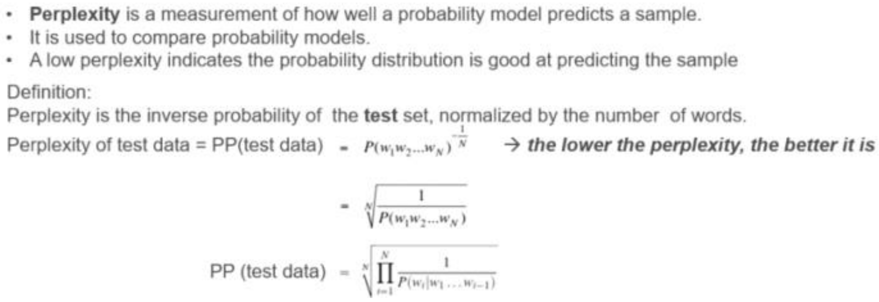
xi (ξ) : In the case of a variable lenght document, the document length is determined by sampling from a Poisson distribution with an average length of ξ

Ways to determine number of Topics:

As number of topics increase, perplexity decreases:



What is perplexity?



- **Perplexity** is a measurement of how well a probability model predicts a sample.
- It is used to compare probability models.
- A low perplexity indicates the probability distribution is good at predicting the sample

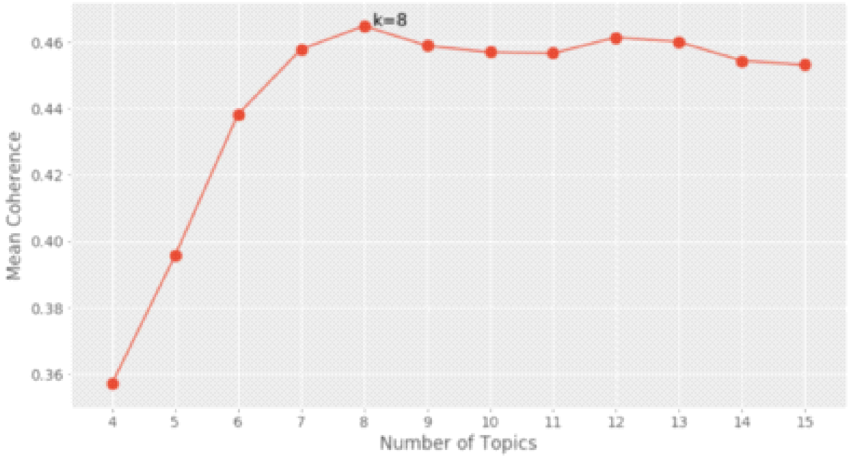
Definition:
Perplexity is the inverse probability of the **test** set, normalized by the number of words.

Perplexity of test data = PP(test data) $= P(w_1 w_2 \ldots w_N)^{-\frac{1}{N}}$ → ***the lower the perplexity, the better it is***

$$= \sqrt[N]{\frac{1}{P(w_1 w_2 \ldots w_N)}}$$

PP (test data) $= \sqrt[N]{\prod_{i=1}^{N} \frac{1}{P(w_i | w_1 \ldots w_{i-1})}}$

As number of topics increase, coherence score increases



5. Applications of LDA

Wednesday, February 06, 2019 11:43 AM

There are many applications. As shown in the paper:

1. **Document Modeling** - (same as topic modeling but defined here from the perspective of documents) -

   Reducing the large number of features (words) of a document into a manageable list of topics. Then comparing what are the related documents, which category of documents (a.k.a category of documents= topic was spoken most)

   You train LDA on a training data and test it on a test/held out list of documents. You can choose that LDA model which has the lowest perplexity score for the same number of topics

$$perplexity(D_{\text{test}}) = \exp\left\{ -\frac{\sum_{d=1}^{M} \log p(\mathbf{w}_d)}{\sum_{d=1}^{M} N_d} \right\}$$

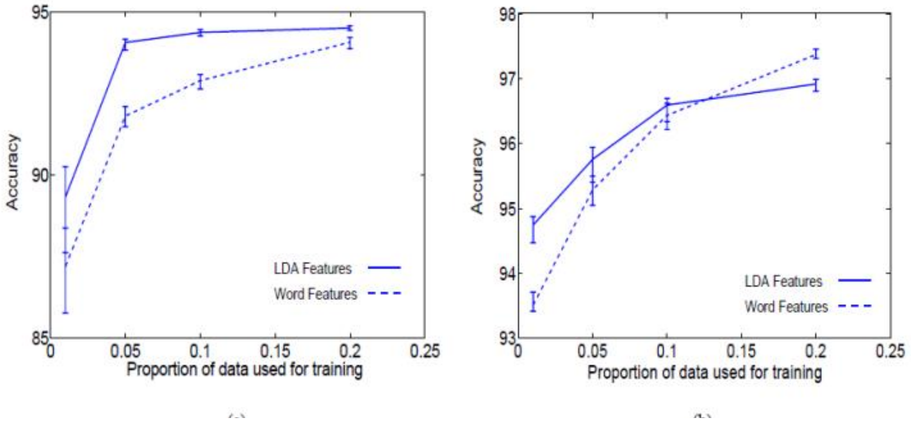
2. **Document Classification**:
   Reduce the high-dimensional word_features for a text data into a more fixed set of real-valued topic_features

   d1 = [78 92 98 8 2 8 …. 9] (a document d1 is represented as a V x 1 array where V represents the length of the vocabulary

   d1 = [0.4 0.3 0.2 0.1] (the same document d1 is represented as a K x 1 array where K represents the number of topics of the LDA model

   We can similarity between two documents effectively if d1 is represented like the latter than the former.

   LDA Features model = SVM over the low-dimensional topic features
   Word Features model = SVM over the high-dimensional word features



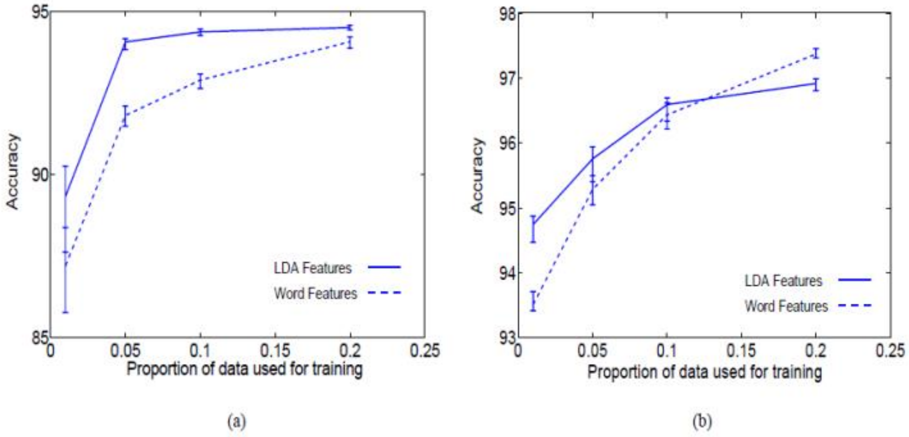
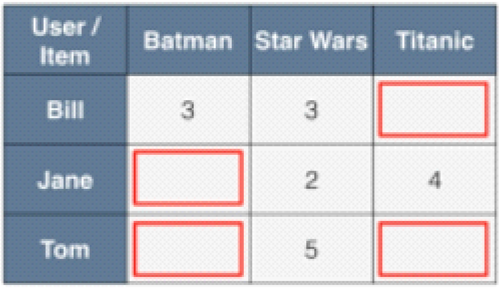


Figure 10: Classification results on two binary classification problems from the Reuters-21578 dataset for different proportions of training data. Graph (a) is EARN VS. NOT EARN. Graph (b) is GRAIN VS. NOT GRAIN.

3. **Collaborative Filtering**

In a movie recommendation dataset, a collection of users indicate ratings for a list of movies



User / Item	Batman	Star Wars	Titanic
Bill	3	3	
Jane		2	4
Tom		5	

User x Movies == Doc vs Words

$$predictive\text{-}perplexity(D_{\text{test}}) = \exp\left\{-\frac{\sum_{d=1}^{M} \log p(w_{d,N_d} \mid \mathbf{w}_{d,1:N_d-1})}{M})\right\}.$$

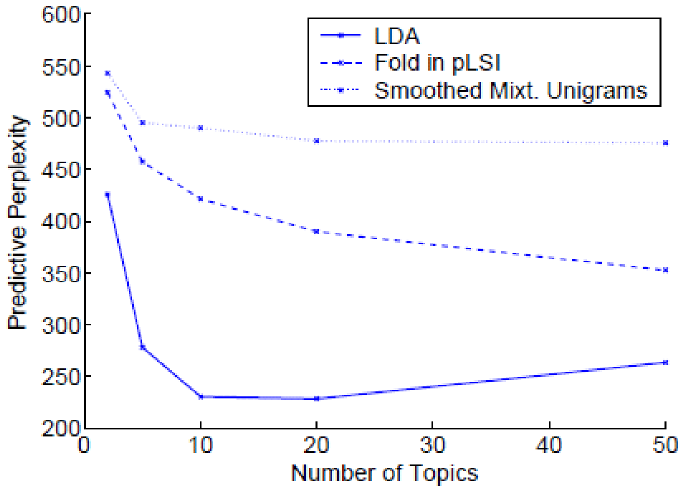


Figure 11: Results for collaborative filtering on the EachMovie data.

6. Summary

Wednesday, February 06, 2019 3:21 PM

- A flexible generative probabilistic model for collections of discrete data
- LDA is based on an 'Exchangeability' assumption - Exchange Words for Topics
- LDA can be viewed as a dimensionality reduction technique.

APPENDIX Good Links

Wednesday, February 06, 2019 12:20 PM

Apart from the original paper,

https://www.slideshare.net/hustwj/nicolas-loeff-lda
http://videolectures.net/mlss09uk_blei_tm/
http://www.cs.columbia.edu/~blei/talks/Blei_ICML_2012.pdf
https://www.quora.com/What-is-an-intuitive-explanation-of-the-Dirichlet-distribution (it compares normal distribution with Dirichlet Distribution)
https://ldabook.com/lda-as-a-generative-model.html
https://ldabook.com/index.html