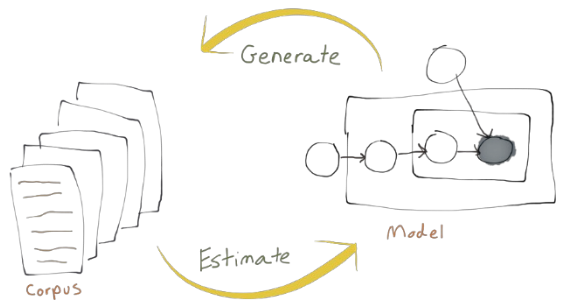
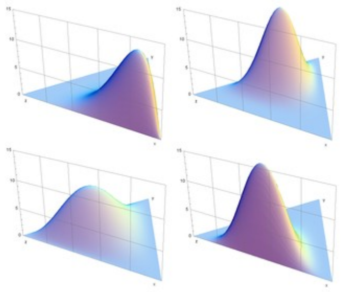
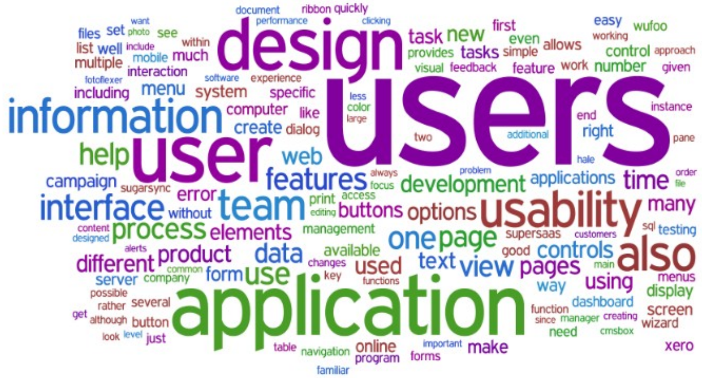
Latent Dirichlet Allocation (LDA)

A review of topic modeling and customer interactions application







Agenda

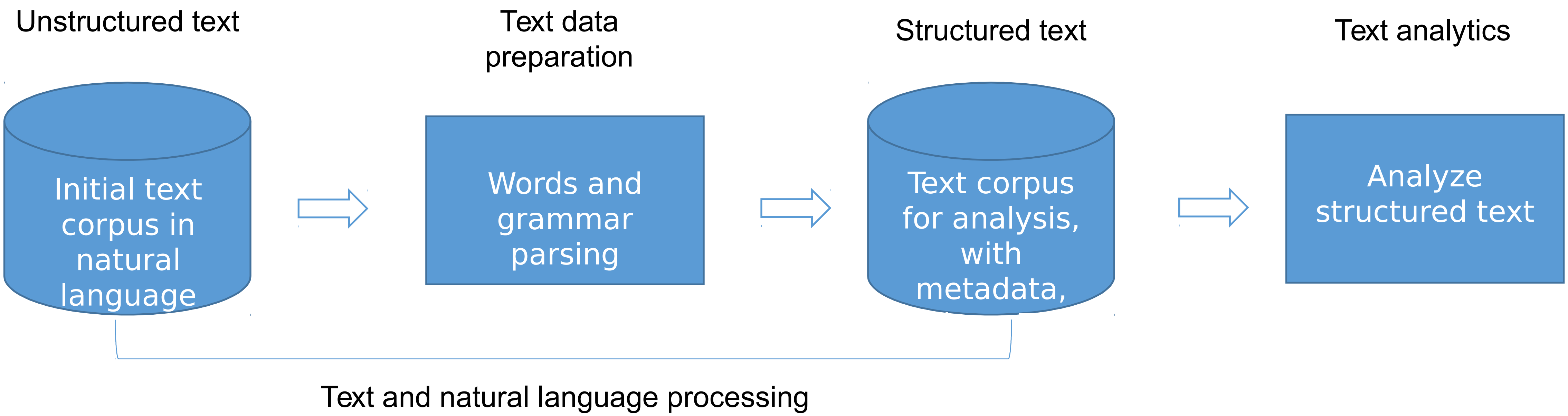
Agenda Items

1. What is topic modeling?
   - Intro Text Mining & Pre-Processing
   - Natural Language Processing & Topics
2. Introduction into Latent Dirichlet Allocation (LDA)
   - LDA Graphical Model
   - The Dirichlet Distribution
   - Generative Process
   - Gibbs Sampling
   - Maximum Likelihood Estimates
3. Application - Customer Incident Routing
4. Demo in R
5. Wrap up
6. Questions

Quick Text Mining Introduction

What is topic modeling?

Intro Text Mining & Pre-Processing



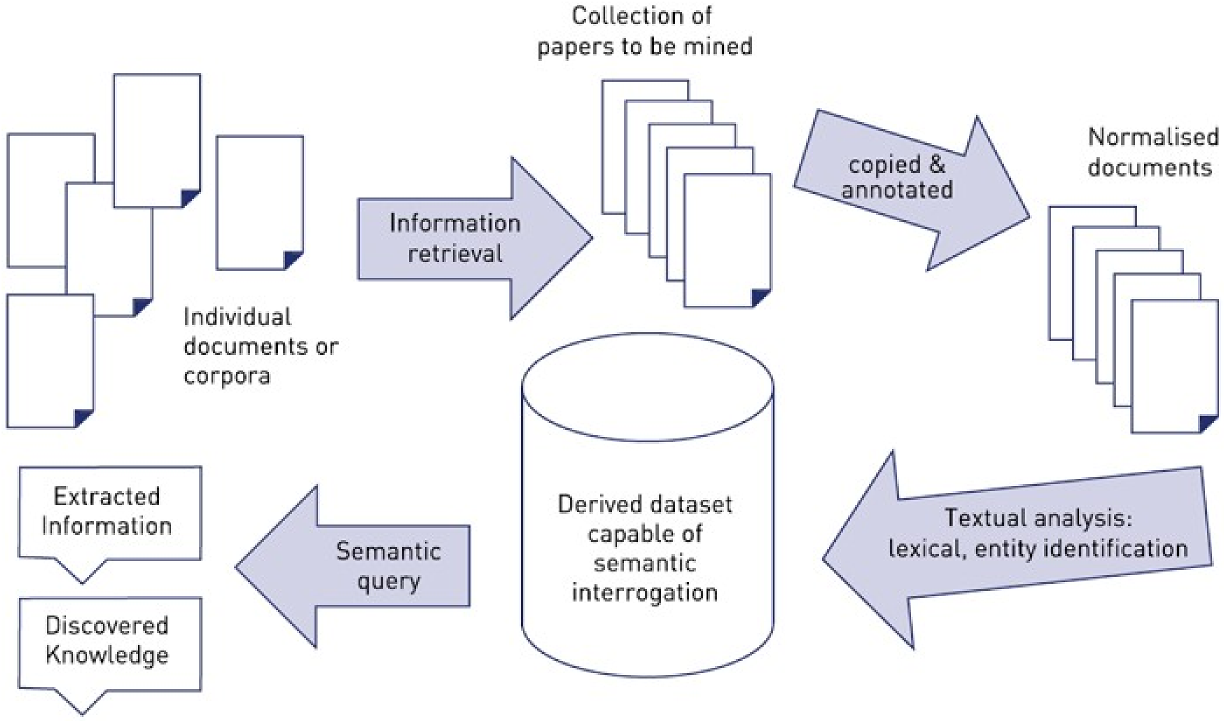
Source: Adaptive from Miller (2005)

Text mining and other terms

- **Corpus:** is a large and structured set of texts
- **Stop words:** words which are filtered out before or after processing of natural language data (text)
- **Unstructured text:** information that either does not have a pre-defined data model or is not organized in a pre-defined manner.
- **Tokenizing:** process of breaking a stream of text up into words, phrases, symbols, or other meaningful elements called tokens (see also lexical analysis)
- **Natural language processing:** field of computer science, artificial intelligence, and linguistics concerned with the interactions between computers and human (natural) languages
- **Term document (or document term) matrix:** is a mathematical matrix that describes the frequency of terms that occur in a collection of documents
- **Supervised learning:** s the machine learning task of inferring a function from labeled training data
- **Unsupervised learning:** find hidden structure in unlabeled data
- **Stemming:** the process for reducing inflected (or sometimes derived) words to their word stem, base or root form—generally a written word form

Source: Wikipedia

Document & information retrieval



Source: http://www.jisc.ac.uk/reports/value-and-benefits-of-text-mining

The idea is how do we take this unstructured text, index it in such a way that allows us to integrate the structure analytics back to the core information to move, sort, search, process, categorize, etc. by document.

Common IR goals:

- Ad-hoc retrieval
- Filtering/Sorting
- Browsing

Pre-Processing for Topic Modeling

Base Corpus

Clean the Corpus

- Lower case
- Remove number, punctuation, etc.
- Steaming
- Remove stop words

Generate Term Doc Matrix

- Tokenize
- Word length
- Apply constraints (sparsity, etc.)

```
packages <- c('tm', 'NLP', 'SnowballC', 'openNLP',
'openNLPmodels.en', 'RWeka')
```

Pre-processing

- The input data for topic models is a <u>document-term matrix</u>. The rows in this matrix correspond to the documents and the columns to the terms.

- The number of rows is equal to the size of the corpus and the number of columns to the size of the vocabulary

- Mapping from the document to the term frequency vector involves tokenizing the document and then processing the tokens for example by converting them to lower-case, removing punctuation characters, removing numbers, stemming, removing stop words and omitting terms with a length below a certain minimum

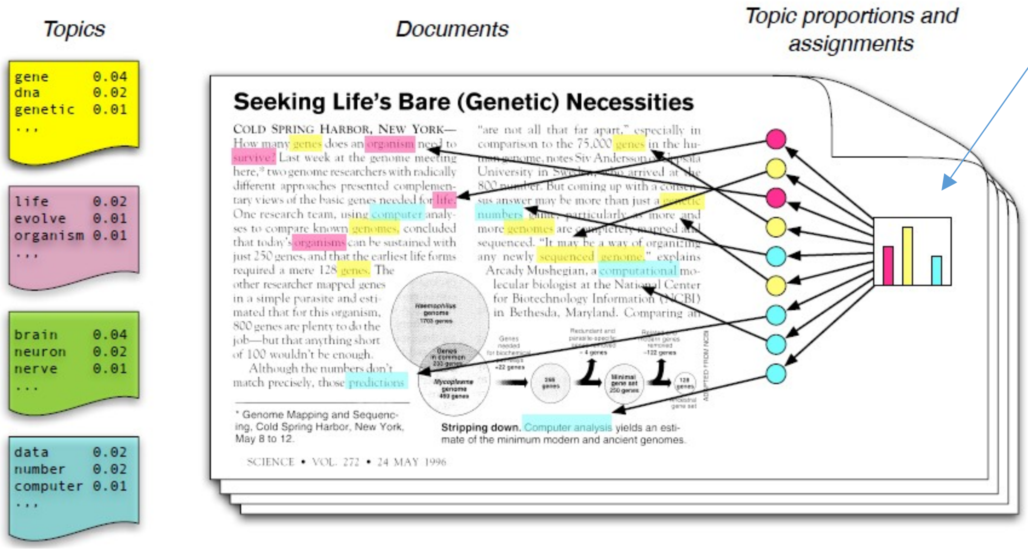
- Each term in a collection's vocabulary the index maps in which document the term was posted (inverted indices or lists)

Topic Modeling

Probabilistic modeling

1. Treat data as observations that arise from a generative probabilistic process that includes hidden variables:
   - For documents, the hidden variables reflect the thematic structure of the collection.
2. Infer the hidden structure using posterior inference:
   - What are the topics that describe this collection?
3. Situate new data into the estimated model:
   - How does this query or new document fit into the estimated topic structure?

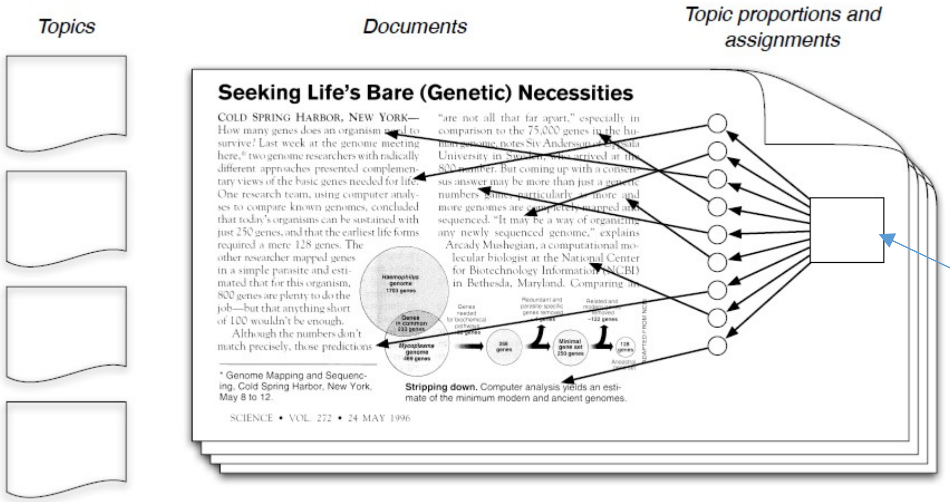
Generative Model & The Posterior Distribution



Each doc is a random mixture of corpus-wide topics and each word is drawn from one of those topics. This assumes topics exists outside of the doc collection. Each topic is a distribution over fixed vocabulary.

<u>Generative Process</u>:

- First, choose a distribution over topics (drawn from a Dirichlet distribution where yellow, pink, green, and blue have some probabilities)
- Then, repeatedly draw a word (color) from each distribution
- Next, lookup what each word topic it belongs to by the color
- Finally, choose the word from that distribution



<u>Posterior Distribution</u>: Conditional distribution of all latent variables given the observations which are in this case are each of the words of the documents. However, we actually only observe the docs and therefore must infer the underlying topic structure.

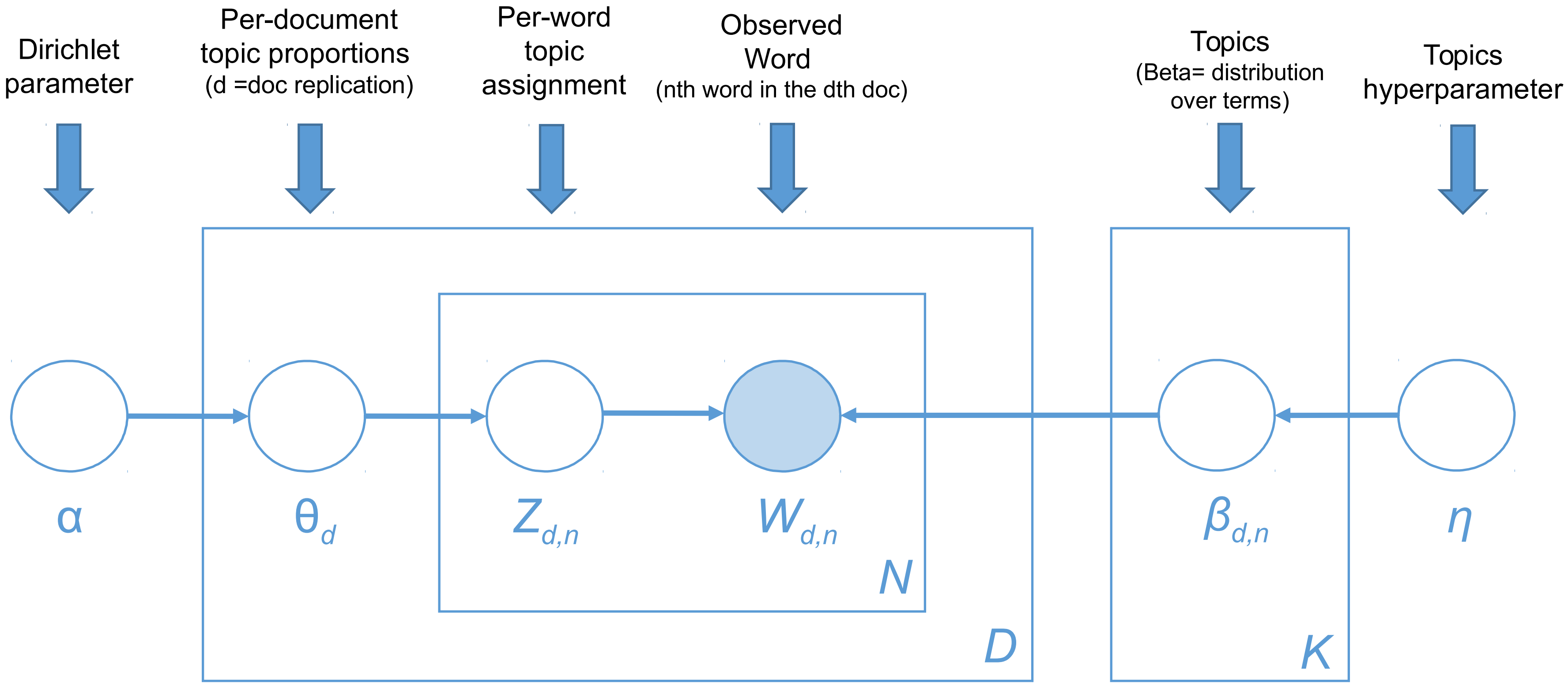
- Goal is to infer the underlying topic structure, given documents being considered/observed
- What are the topics generated under these assumptions?
- What are the distribution over terms that generated these topics?
- For each document, what is the distributions over topics associated with that document?
- For each word, which topic generated each word
- Conditional distribution of all of these latent variables given the observations which are the words in the documents

Intro to Latent Dirichlet Allocation (LDA)

What is Latent Dirichlet Allocation (LDA)?	A generative probabilistic model for collections of discrete data such as text corpora. LDA is a three-level hierarchical Bayesian model, in which each item of a collection is modeled as a finite mixture over an underlying set of latent topics. Each observed word originates from a topic that we do not directly observe. Each topic is, in turn, modeled as an infinite mixture over an underlying set of topic probabilities.
What is used for?	The fitted model can be used to estimate the similarity between documents as well as between a set of specified keywords using an additional layer of latent variables which are referred to as topics.
How is it related to text mining and other machine learning techniques?	Topic models can be seen as classical text mining or natural language processing tools. Fitting topic models based on data structures from the text mining usually done by considering the problem of modeling text corpora and other collections of discrete data. One of the advantages of LDA over related latent variable models is that it provides well-defined inference procedures for previously unseen documents (LSI uses a singular value decomposition)

Introduction into Latent Dirichlet Allocation (LDA)

LDA Graphical Model



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.

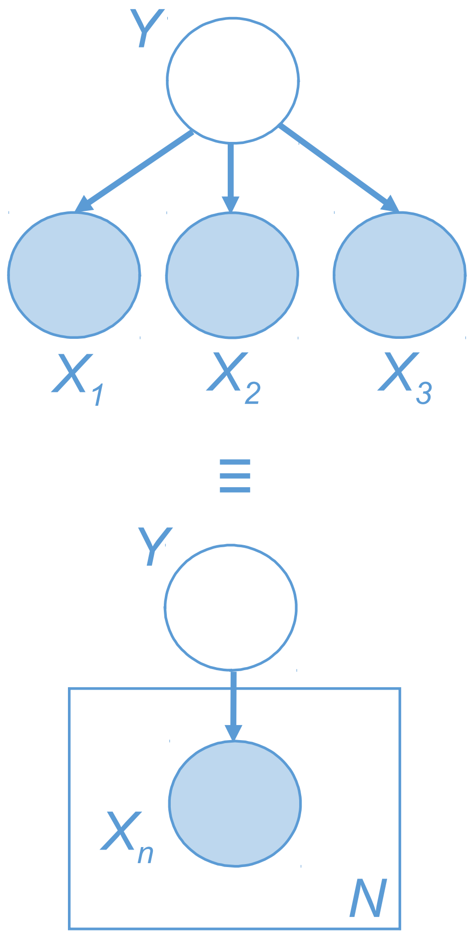
- Nodes are random variables
- Edges denote possible dependence
- Observed variables are shaded
- Plates denote replicated structure

K	specified number of topics	i	auxiliary index over words in a document
k	auxiliary index over topics	α	positive K-vector
V	number of words in vocabulary	β	positive V-vector
v	auxiliary index over topics	$Dir(\alpha)$	a K-dimensional Dirichlet
d	auxiliary index over documents	$Dir(\beta)$	a V-dimensional Dirichlet
N_d	document length (number of words)	z	Topic indices: $z_{d,i} = k$ means that the i-th word in the d-th document is assigned to topic k

Plates

- *D* = docs
- *N* = words
- *K* = topics

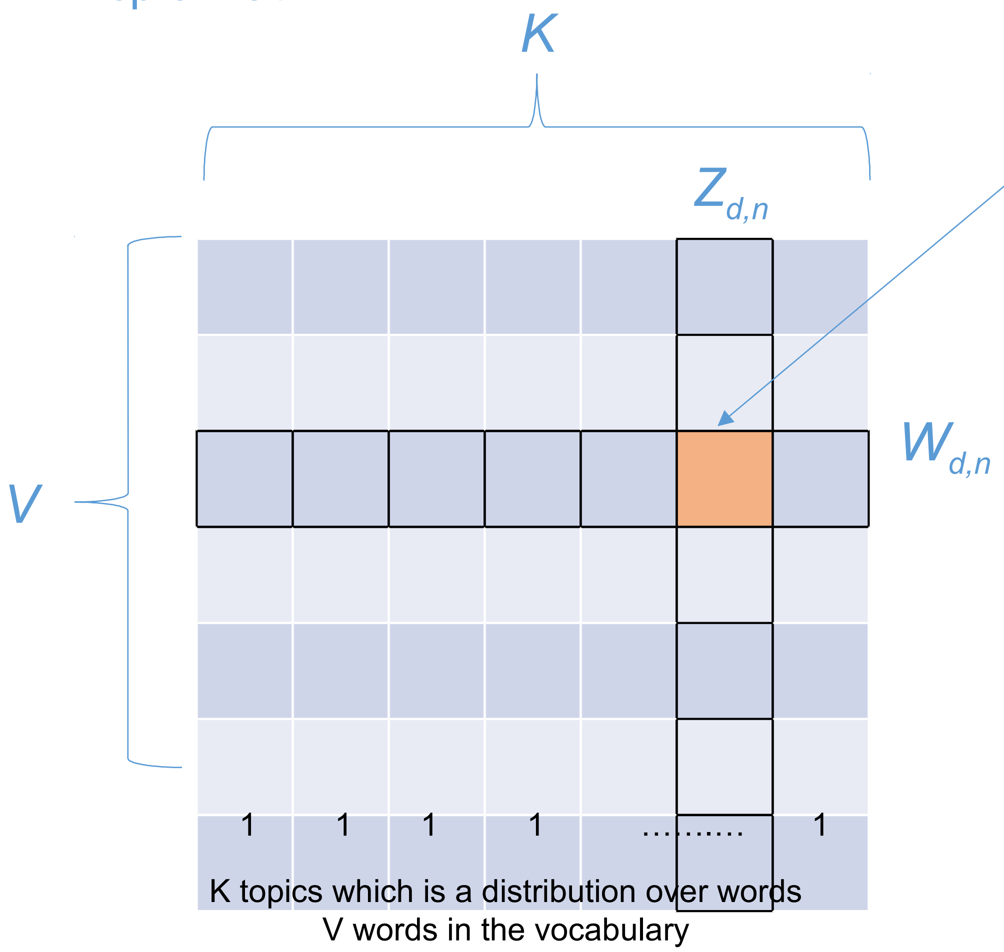
Graphical models



SMU. Bobby B. Lyle School of Engineering

Introduction into Latent Dirichlet Allocation (LDA)

Topic Matrix



1

- Once we select Z, we know what topic its coming from, then lookup the cell
- Lookup in beta in the Zdn column the Wdn word and get the words probability from there
- That is why we have the observed Wdn depend on all the Zdn and beta's

2

Probability of observing this word: $\rho(w_{d,n}|Z_{d,n},\beta_{d,k})$

WHERE Wd,n is the observed word
AND Zd,n is an index from 1 to k
AND Beta d,k are the topics

3

Joint probability of all the hidden and observed variables according to this model:

Comes from a Dirichlet

$$\left(\prod_{k=1}^{K} p(\beta_K|\eta)\right)\left(\prod_{d=1}^{D} p(\theta_d|\alpha)\left(\prod_{n=1}^{N} p(Z_{d,n}|\theta_d)\ \rho(w_{d,n}|Z_{d,n},\beta_{d,k})\right)\right)$$

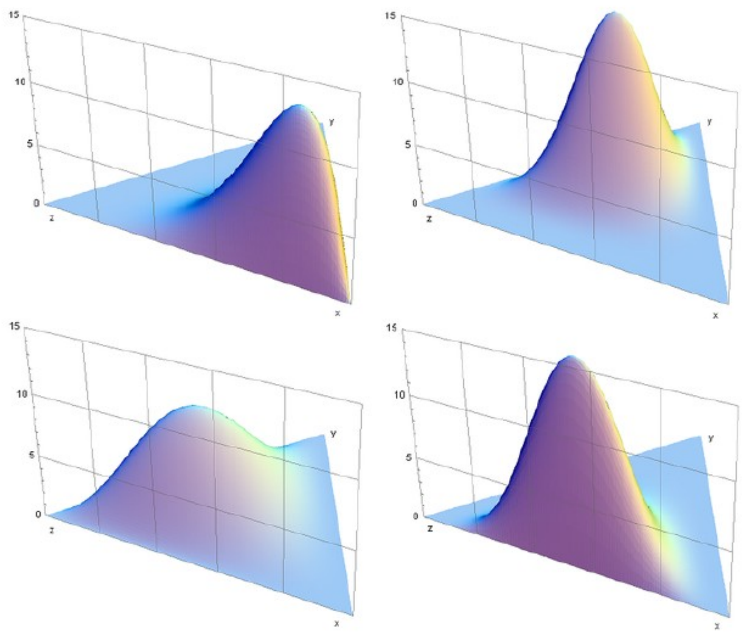
Each topic coming from some distribution that is appropriate over topics (Dirichlet) and is independent

In our documents, generate the topic proportion, using alpha

Within each doc, we have the words, drawn from the topic assignment from theta d

Probability of observing this word, conditioned on Zdn and the Beta's

The Dirichlet Distribution



(From Wikipedia)

1. The Dirichlet distribution is an exponential family distribution over the simplex, i.e., positive vectors that sum to one

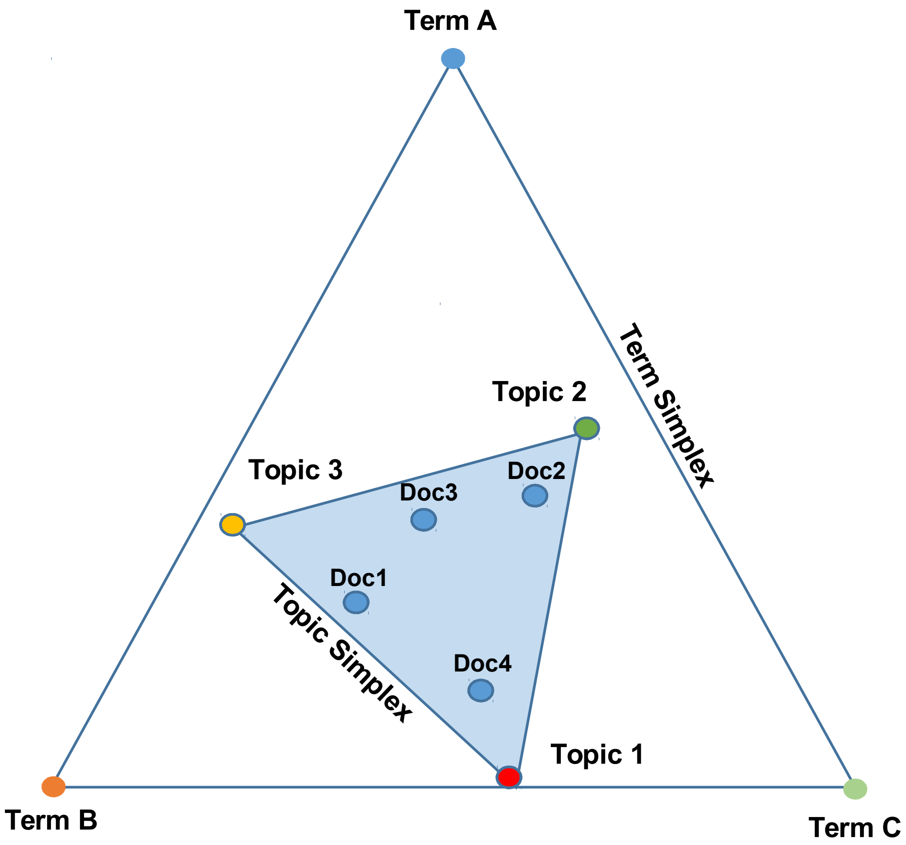
$$p(\theta \mid \vec{\alpha}) = \frac{\Gamma\left(\sum_i \alpha_i\right)}{\prod_i \Gamma(\alpha_i)} \prod_i \theta_i^{\alpha_i - 1}$$

2. The Dirichlet is conjugate to the multinomial. Given a multinomial observation, the posterior distribution of θ is a Dirichlet.

3. The parameter α controls the mean shape and sparsity of θ. Parameter α is a k-vector with components αi >0

4. The topic proportions are a K dimensional Dirichlet. The topics are a V dimensional Dirichlet.

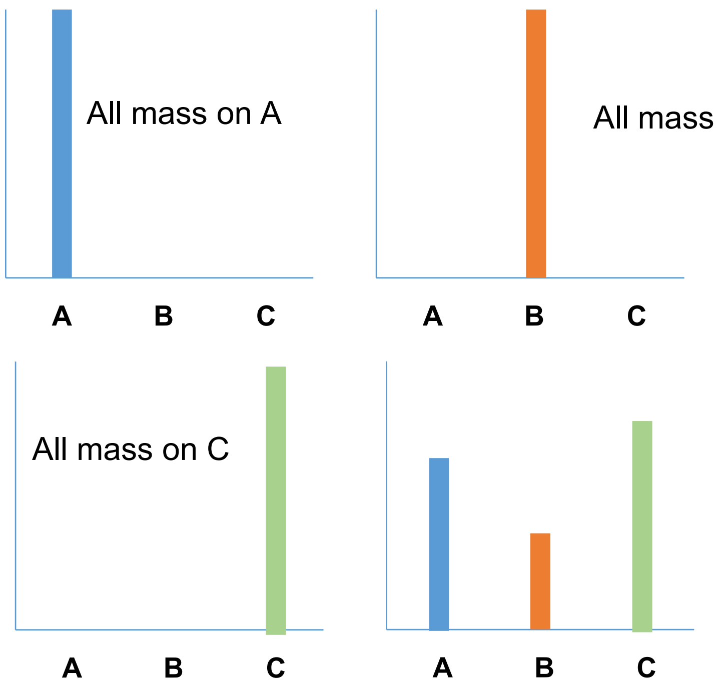
Geometric Interpretation of LDA



3 Dimension Dirichlet

as we draw random variables from theta, I'm going to get distributions over 3 elements.

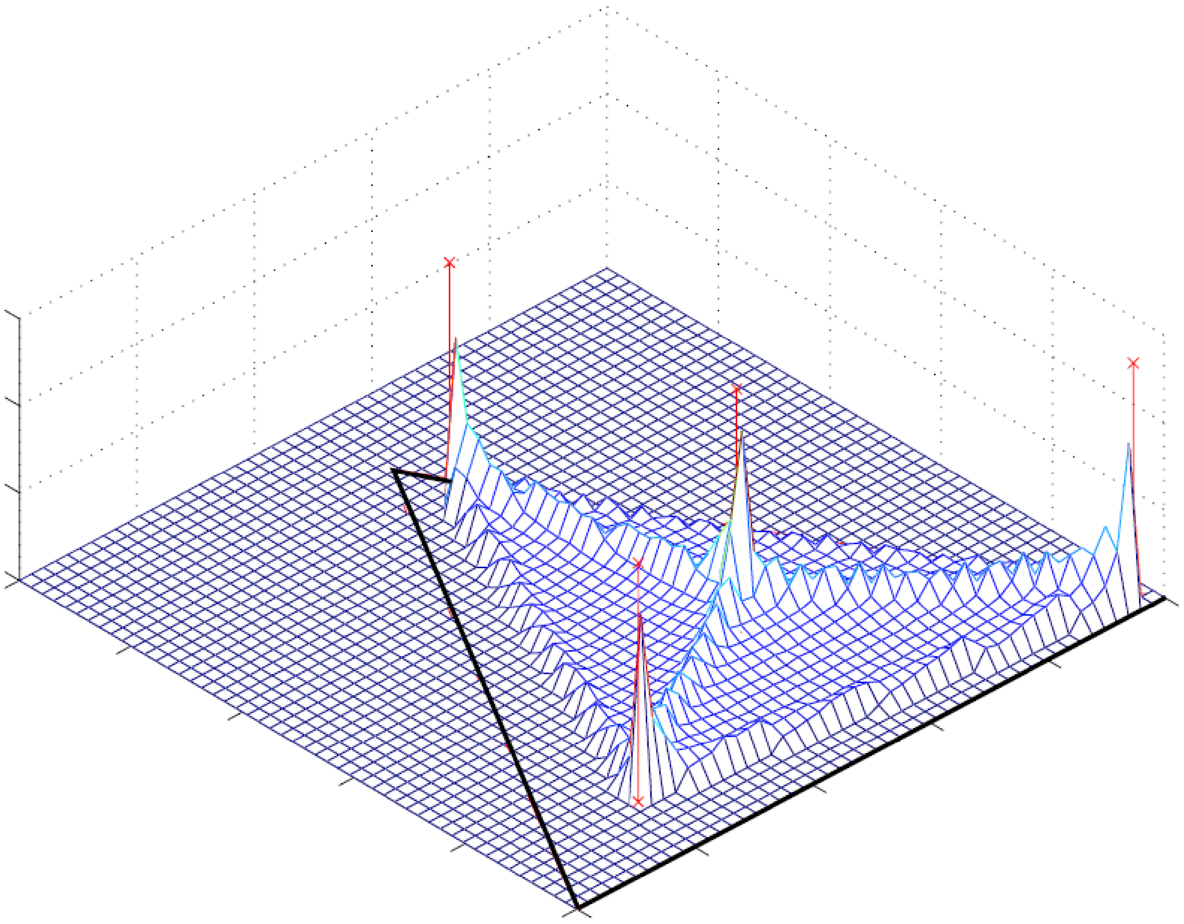
θ ~ Dirichlet(1,1,1) = α1 = α2 = α3 = 1, uniform distribution as an example



Some point within the space over the distribution of 3 items. The Dirichlet places a distribution over this space.

Dirichlet is parameterized by α, so as α increases the chart gets more peaky.

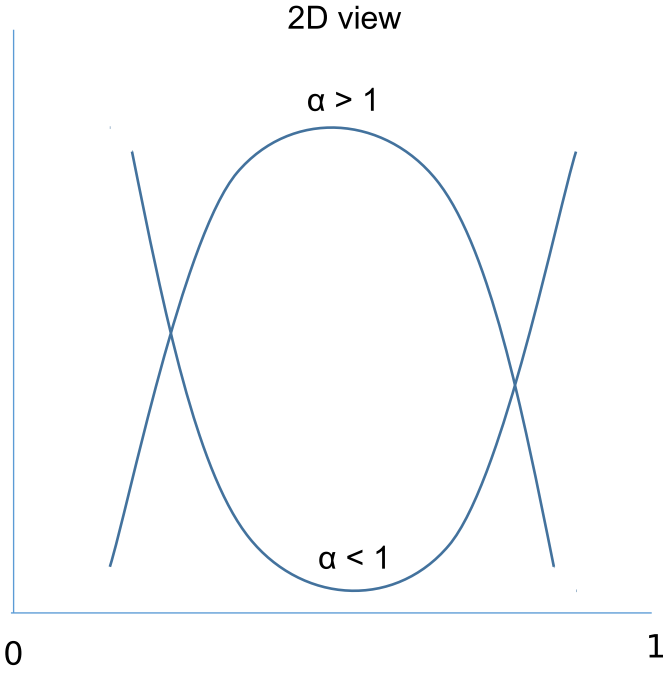
Density Example



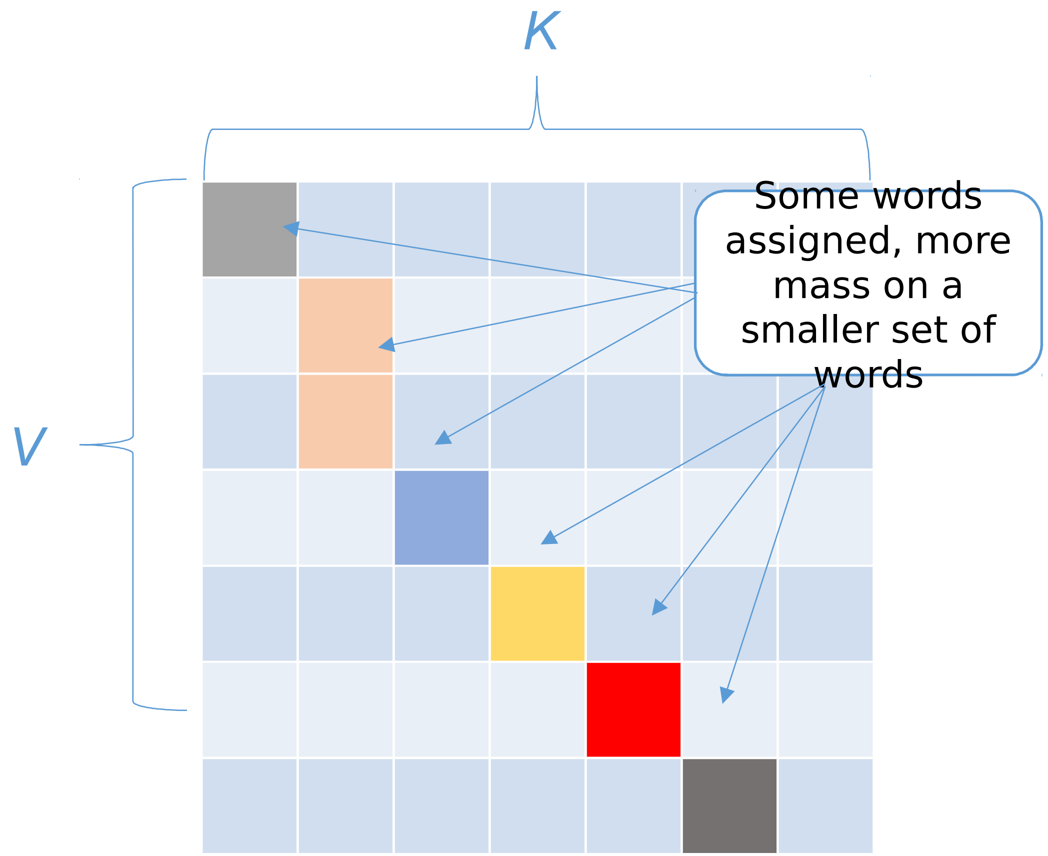
When α < 1 (s < k), you get sparsity and on the 3 simplex you get a figure with increased probability at the corners.

Important piece of info:

1) The expectations of the posterior (sometimes called M for mean)
2) The sum of the alphas, which determines the peaky-ness of the Dirichlet
   - If this sum is small, the Dirichlet will be more spread out
   - If large, the Dirichlet will have more peaks at its expectation (sometimes called S for scaling)



LDA Inferences



V words in the vocabulary

LDA puts posterior topical words together by:

1. Maximizing the word probabilities by dividing the words among the topics.
    1. Joint distribution:
2. In a mixture model, finds cluster of co-occurring words (in the same topic)

In LDA, a document will be penalized for having too many topics (hyperparameter)

Loosely, this can be thought of as softening the strict definition of “co-occurrence" in a mixture model

This flexibility leads to sets of terms that more tightly co-occur

Likelihood term

$$\left(\prod_{d=1}^{D} p(\theta_d|\,\alpha)\left(\prod_{n=1}^{N} p(Z_{d,n}|\,\theta_d)\;\underline{p(w_{d,n}|Z_{d,n},\beta_{d,k})}\right)\right)$$

Posterior distribution & model estimation for LDA

Approximate posterior inference methods

1 Gibbs sampling

- The Gibbs sampling algorithm is a typical Markov Chain Monte Carlo (Mcmc) method and was originally proposed for image restoration
- Define a Markov chain whose stationary distribution is the posterior of interest
- Collect independent samples from that stationary distribution; approximate the posterior with them
- The chain is run by iteratively sampling from the conditional distribution of each hidden variable given observations and the current state of the other hidden variables
- Once a chain has “burned in," collect samples at a lag to approximate the posterior.

2 Variational methods (Bayesian inference & Collapsed variational Bayesian inference)

- Variational methods are a deterministic alternative to MCMC.
- For many interesting distributions, the marginal likelihood of the observations is difficult to efficiently compute
- The goal is to optimize the variational parameters to make tight as possible

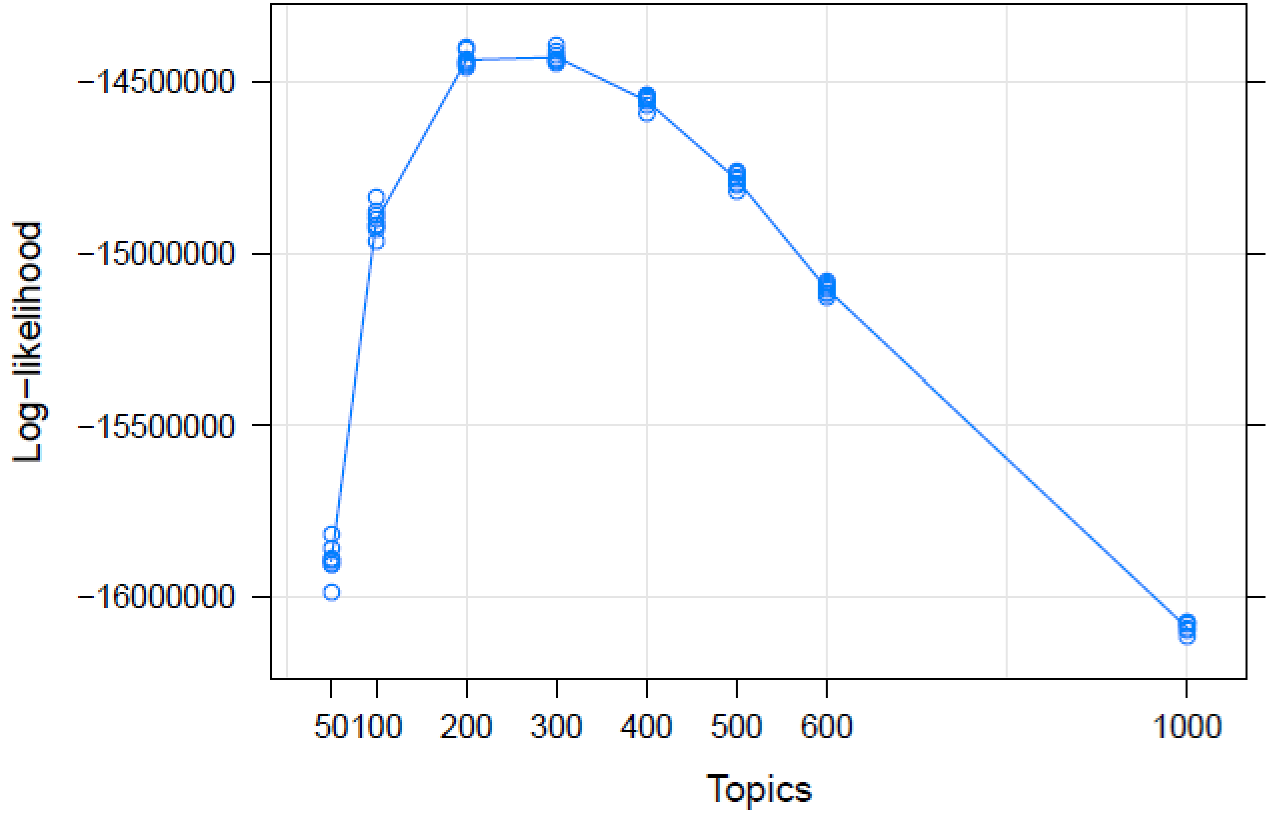
3 Particle filtering

<u>Summary of learning algorithm for Gibbs</u>:

- Initialize the topic to word assignments z randomly from {1,. . . ,K}
- For each Gibbs sample:
  - “For each word token, the count matrices n^-(a,b) are first decremented by one for the entries that correspond to the current topic assignment.“
  - The count matrices are updated by incrementing by one at the new topic assignment.
- Discard samples during the initial burn-in period
- After the Markov chain has reached a stationary distribution, i.e., the posterior distribution over topic assignments, samples can be taken at a fixed lag (averaging over Gibbs samples is recommended for statistics that are invariant to the ordering of topics)

Maximum likelihood (Ml) estimation

Example: Estimated marginal log-likelihoods per number of topics (circles), average likelihoods are connected by lines



Empirical Bayes method for parameter estimation:

- Given a corpus of docs we want to find parameters α and β that maximize the (marginal) log likelihood of the data

Using R & Demo

R implementations of Latent Dirichlet Allocation (LDA)

Available packages through CRAN

Topic models

- Provides an interface to the C code for Latent Dirichlet Allocation (LDA) models and Correlated Topics Models (CTM) by David M. Blei and co-authors and the C++ code for fitting LDA models using Gibbs sampling by Xuan-Hieu Phan and co-authors

lda

- This package implements latent Dirichlet allocation (LDA) and related models. This includes (but is not limited to) sLDA, corrLDA, and the mixed-membership stochastic blockmodel. Inference for all of these models is implemented via a fast collapsed Gibbs sampler writtten in C. Utility functions for reading/writing data typically used in topic models, as well as tools for examining posterior distributions are also included

These functions use a collapsed Gibbs sampler to fit three different models: latent Dirichlet allocation (LDA), the mixed-membership stochastic blockmodel (MMSB), and supervised LDA (sLDA).
These functions take sparsely represented input documents, perform inference, and return point estimates of the latent parameters using the state at the last iteration of Gibbs sampling.

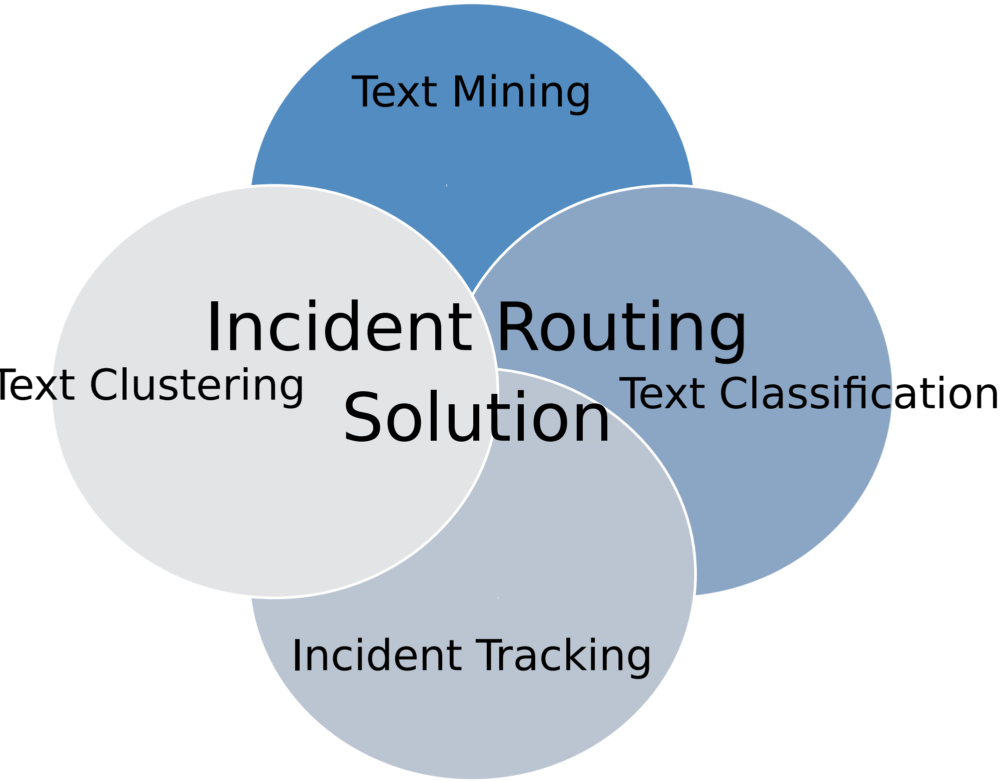
```
lda.collapsed.gibbs.sampler(documents, K, vocab, num.iterations, alpha,
eta, initial = NULL, burnin = NULL, compute.log.likelihood = FALSE,
  trace = 0L, freeze.topics = FALSE)

slda.em(documents, K, vocab, num.e.iterations, num.m.iterations, alpha,
eta, annotations, params, variance, logistic = FALSE, lambda = 10,
regularise = FALSE, method = "sLDA", trace = 0L)

mmsb.collapsed.gibbs.sampler(network, K, num.iterations, alpha,
beta.prior, initial = NULL, burnin = NULL, trace = 0L)
```

interesting is the post “Finding structure in xkcd comics with Latent Dirichlet Allocation”: http://cpsievert.github.io/projects/615/xkcd/

Application - Customer Incident Routing



What are the important drivers of incidents?

Text mining of incident data to find keywords and themes

What are the categories of incidents?

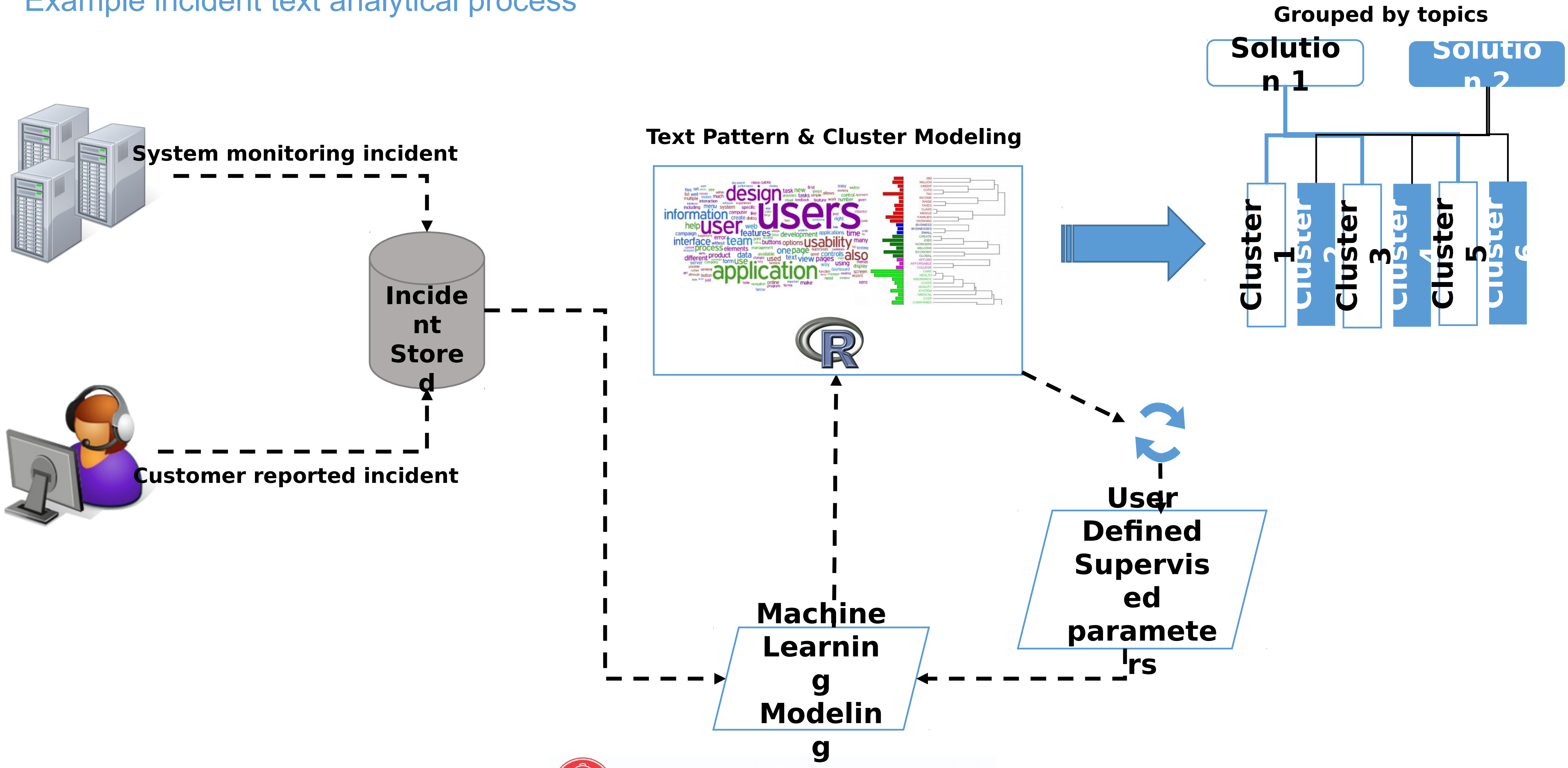
Text clustering to identify incident groups

How to derive insights and map to known solutions?

Text classification to tag each incident to theme clusters

Incident Text Analytics Process

Example incident text analytical process



References

- "Latent Dirichlet Allocation" David M. Blei, Andrew Y. Ng, Michael I. Jordan - Journal of Machine Learning Research 3 (2003) 993-1022
- "Topic Models" lecture David M. Blei, September 1, 2009 found at http://videolectures.net/mlss09uk_blei_tm/
- "Latent Dirichlet Allocation in R" Martin Ponweiser, Institute for Statistics and Mathematics http://statmath.wu.ac.at/, Thesis 2, May 2012
- "topicmodels: An R Package for Fitting Topic Models", Bettina Grun & Kurt Hornik
- "Text mining" Ian H. Witten, Computer Science, University of Waikato, Hamilton, New Zealand

Wrap up & Questions