CS5560 Knowledge Discovery and Management

Problem Set 5

July 3 (T), 2017

Name: Sai Sriharsha S

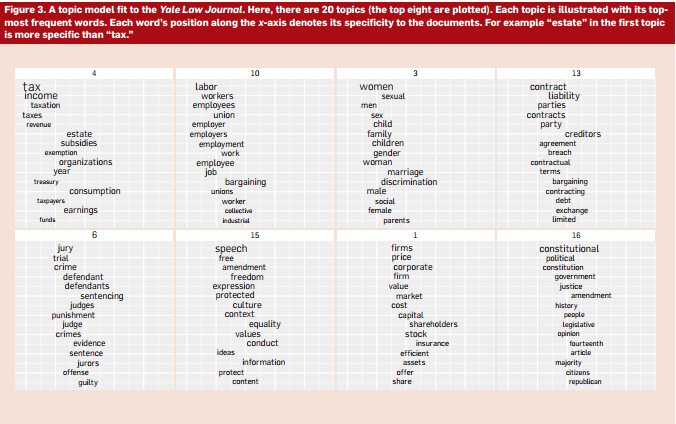
Class ID: 25

1. LDA

Read the following articles to learn more about LDA

* <https://algobeans.com/2015/06/21/laymans-explanation-of-topic-modeling-with-lda-2/>
* <http://engineering.intenthq.com/2015/02/automatic-topic-modelling-with-lda/>

Consider the topics discovered from Yale Law Journal. (Here the number of topics was set to be 20.) Topics about subjects like about discrimination and contract law.



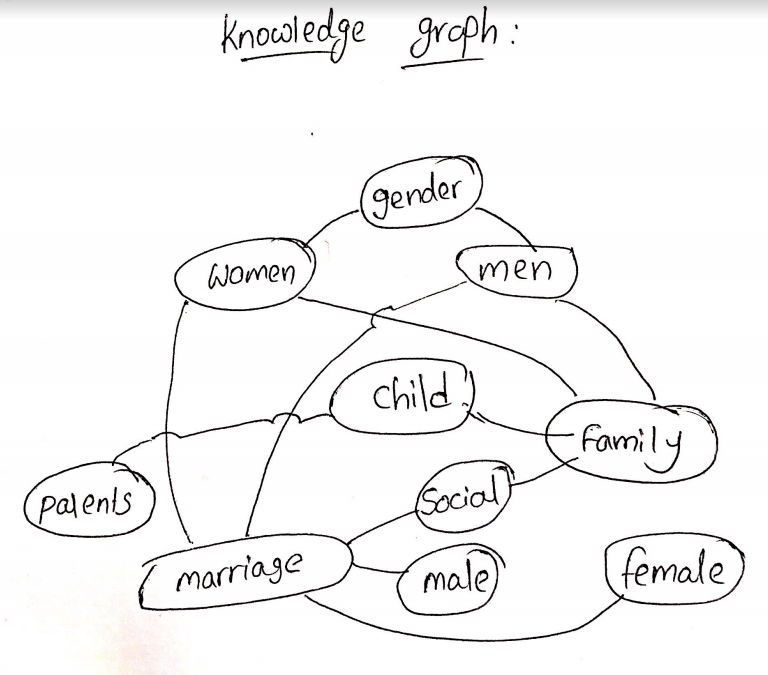
1. Describe the overall process to generate such topics from the corpus.

LDA is a statistical model of document collections that tries to capture this intuition. It is most easily described by its generative process, the imaginary random process by which the model assumes the documents arose. (The interpretation of LDA as a probabilistic model is fleshed out later.)

We formally define a *topic* to be a distribution over a fixed vocabulary. For example, the *genetics* topic has words about genetics with high probability and the *evolutionary biology* topic has words about evolutionary biology with high probability. We assume that these topics are specified before any data has been generated.[a](https://cacm.acm.org/magazines/2012/4/147361-probabilistic-topic-models/fulltext#FNA)Now for each document in the collection, we generate the words in a two-stage process.

* Randomly choose a distribution over topics.
* For each word in the document
  1. Randomly choose a topic from the distribution over topics in step #1.
  2. Randomly choose a word from the corresponding distribution over the vocabulary.

This statistical model reflects the intuition that documents exhibit multiple topics. Each document exhibits the topics in different proportion (step #1); each word in each document is drawn from one of the topics (step #2b), where the selected topic is chosen from the per-document distribution over topics (step #2a).

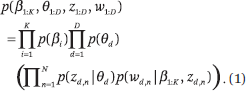
1. Draw a knowledge graph for Topic 3 in Yale Law Journal (The First Figure).
2. Each topic is illustrated with its topmost frequent words. Each word’s position along the x-axis denotes its specificity to the documents. For example “estate” in the first topic is more specific than “tax.” (the second figure). Describe how to determine the generality or specificity of the terms in a topic.

LDA and other topic models are part of the larger field of *probabilistic modeling*. In generative probabilistic modeling, we treat our data as arising from a generative process that includes *hidden variables*. This generative process defines a *joint probability distribution* over both the observed and hidden random variables. We perform data analysis by using that joint distribution to compute the *conditional distribution* of the hidden variables given the observed variables. This conditional distribution is also called the *posterior distribution*.

LDA falls precisely into this framework. The observed variables are the words of the documents; the hidden variables are the topic structure; and the generative process is as described here. The computational problem of inferring the hidden topic structure from the documents is the problem of computing the posterior distribution, the conditional distribution of the hidden variables given the documents.

We can describe LDA more formally with the following notation. The topics are *β*1*:K*, where each *βk* is a distribution over the vocabulary. The topic proportions for the *d*th document are *θd*, where *θd,k* is the topic proportion for topic *k* in document *d* .The topic assignments for the *d*th document are *zd*, where *zd,n* is the topic assignment for the *n*th word in document *d*. Finally, the observed words for document *d* are *wd*, where *wd,n* is the *n*th word in document *d*, which is an element from the fixed vocabulary.

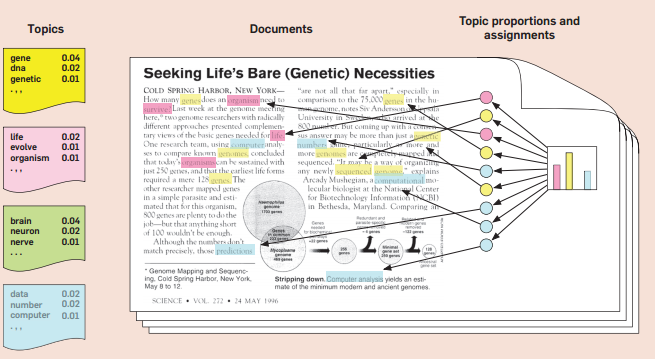
With this notation, the generative process for LDA corresponds to the following joint distribution of the hidden and observed variables,



Notice that this distribution specifies a number of dependencies. For example, the topic assignment *zd,n*depends on the per-document topic proportions *θd*. As another example, the observed word *wd,n* depends on the topic assignment *zd,n* and *all* of the topics *β*1*:K*. (Operationally, that term is defined by looking up as to which topic *zd,n* refers to and looking up the probability of the word *wd,n* within that topic.)

These dependencies define LDA. They are encoded in the statistical assumptions behind the generative process, in the particular mathematical form of the joint distribution, and—in a third way—in the *probabilistic graphical model* for LDA. Probabilistic graphical models provide a graphical language for describing families of probability distributions.

1. Describe the inference algorithm that was used in LDA.



We now turn to the computational problem, computing the conditional distribution of the topic structure given the observed documents. (As we mentioned, this is called the *posterior*.) Using our notation, the posterior is

eq02.gif

The numerator is the joint distribution of all the random variables, which can be easily computed for any setting of the hidden variables. The denominator is the *marginal probability* of the observations, which is the probability of seeing the observed corpus under any topic model. In theory, it can be computed by summing the joint distribution over every possible instantiation of the hidden topic structure.

That number of possible topic structures, however, is exponentially large; this sum is intractable to compute. As for many modern probabilistic models of interest and for much of modern Bayesian statistics. we cannot compute the posterior because of the denominator, which is known as the *evidence*. A central research goal of modern probabilistic modeling is to develop efficient methods for approximating it. Topic modeling algorithms like the algorithms used to create are often adaptations of general-purpose methods for approximating the posterior distribution.

Topic modeling algorithms form an approximation of Equation 2 by adapting an alternative distribution over the latent topic structure to be close to the true posterior. Topic modeling algorithms generally fall into two categories sampling-based algorithms and variational algorithms.

Sampling-based algorithms attempt to collect samples from the posterior to approximate it with an empirical distribution. The most commonly used sampling algorithm for topic modeling is *Gibbs sampling*, where we construct a *Markov chain*—a sequence of random variables, each dependent on the previous—whose limiting distribution is the posterior. The Markov chain is defined on the hidden topic variables for a particular corpus, and the algorithm is to run the chain for a long time, collect samples from the limiting distribution, and then approximate the distribution with the collected samples. (Often, just one sample is collected as an approximation of the topic structure with maximal probability.) See Steyvers and Griffiths for a good description of Gibbs sampling for LDA.

Variational methods are a deterministic alternative to sampling-based algorithms. Rather than approximating the posterior with samples, variational methods posit a parameterized family of distributions over the hidden structure and then find the member of that family that is closest to the posterior. Thus, the inference problem is transformed to an optimization problem. Variational methods open the door for innovations in optimization to have practical impact in probabilistic modeling

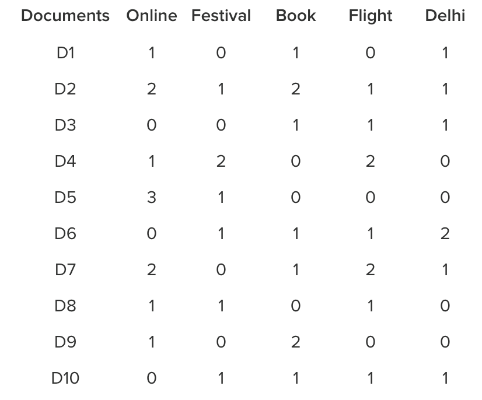
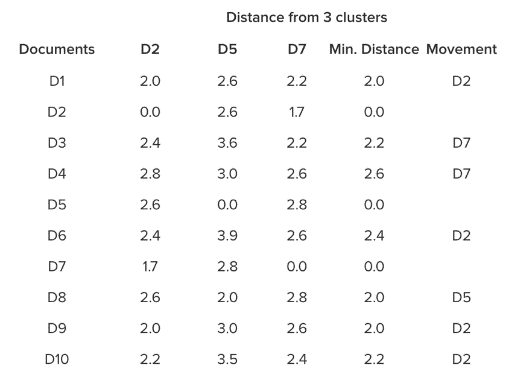
Loosely speaking, both types of algorithms perform a search over the topic structure. A collection of documents (the observed random variables in the model) are held fixed and serve as a guide toward where to search. Which approach is better depends on the particular topic model being used—we have so far focused on LDA, but see below for other topic models—and is a source of academic debate. For a good discussion of the merits and drawbacks of both, see Asuncion et al.

1. K-means clustering vs. LDA

Read the K-means clustering for text clustering from <https://www.experfy.com/blog/k-means-clustering-in-text-data>

1. Describe the steps how the following 10 documents have moved into 3 different clusters using clustered using k-means (K=3).

**Document/Term Matrix**

**Distance Matrix**

**Ans:**

Clustering/segmentation is one of the most important techniques used in Acquisition Analytics. K means clustering groups similar observations in clusters in order to be able to extract insights from vast amounts of unstructured data.

When you want  to analyze the Facebook/Twitter/Youtube comments of a particular event, it would be impossible to manually look at each and every mention and see where the sentiment regarding a particular brand/event/person lies.

* The basic idea of K Means clustering is to form K seeds first, and then group observations in K clusters on the basis of distance with each of K seeds. The observation will be included in the nthseed/cluster if the distance betweeen the observation and the nth seed is minimum when compared to other seeds.

Below is a brief overview of the methodology involved in performing a K Means Clustering Analysis.

### The Process of building K clusters on Social Media text data:

* The first step is to pull the social media mentions for a particular timeframe using social media listening tools (Radian 6, Sysmos, Synthesio etc.).  You would need to build query/add keywords to pull the data from social Media Listening tools.
* The next step is data cleansing. This is the most important part as social media comments do not have any specific format. People use locals/slangs etc. on social media to express their emotions, so it's important to be able to see through them and understand the underlying sentiment.
* Remove punctuations, numbers, stopwords (R has specific stopword library but you can also create your own list of stopwords). Also, remove duplicate rows or URLs from the social media mentions.
* The next step is to create corpus vector of all the words.
* Once you have created the corpus vector of words, the next step is to create a document term matrix.

Let’s visualize the problem with one example. Let’s assume that there are 10 documents/mentions and 5 unique words post data cleansing. Below is the document term matrix for this dataset. It shows for how many times one word has appeared in the document. For example, in document 1 (D1), the words online, book and *Delhi* have each been mentioned once.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Document Term Matrix** | |  |  |  |  |
|  |  |  |  |  |  |
| **Documents** | **Online** | **Festival** | **Book** | **Flight** | **Delhi** |
| D1 | 1 | 0 | 1 | 0 | 1 |
| D2 | 2 | 1 | 2 | 1 | 1 |
| D3 | 0 | 0 | 1 | 1 | 1 |
| D4 | 1 | 2 | 0 | 2 | 0 |
| D5 | 3 | 1 | 0 | 0 | 0 |
| D6 | 0 | 1 | 1 | 1 | 2 |
| D7 | 2 | 0 | 1 | 2 | 1 |
| D8 | 1 | 1 | 0 | 1 | 0 |
| D9 | 1 | 0 | 2 | 0 | 0 |
| D10 | 0 | 1 | 1 | 1 | 1 |

* Let’s assume that we want to create K=3 clusters. First, three seeds should be chosen. Suppose, D2, D5 & D7 are chosen as initial three seeds.
* The next step is to calculate the Euclidean distance of other documents from D2, D5 & D7.
* Assuming: U=Online, V= Festival, X=Book, Y=Flight, Z=Delhi. Then the Euclidean distance between D1 & D2 would be:

((U1-U2)^2 + (W1-W2)^2+(X1-X2)^2+ (Y1-Y2)^2+(Z1-Z2)^2  )^0.5

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Distance Matrix** |  |  |  |  |  |
|  | **Distance from 3 clusters** | | | | |
| **Documents** | **D2** | **D5** | **D7** | **Min. Distance** | **Movement** |
| D1 | 2.0 | 2.6 | 2.2 | 2.0 | D2 |
| D2 | 0.0 | 2.6 | 1.7 | 0.0 |  |
| D3 | 2.4 | 3.6 | 2.2 | 2.2 | D7 |
| D4 | 2.8 | 3.0 | 2.6 | 2.6 | D7 |
| D5 | 2.6 | 0.0 | 2.8 | 0.0 |  |
| D6 | 2.4 | 3.9 | 2.6 | 2.4 | D2 |
| D7 | 1.7 | 2.8 | 0.0 | 0.0 |  |
| D8 | 2.6 | 2.0 | 2.8 | 2.0 | D5 |
| D9 | 2.0 | 3.0 | 2.6 | 2.0 | D2 |
| D10 | 2.2 | 3.5 | 2.4 | 2.2 | D2 |

|  |  |
| --- | --- |
| **Clusters** | **# of Observations** |
| D2 | 5 |
| D5 | 2 |
| D7 | 3 |

* Hence, 10 documents have moved into 3 different clusters. Instead of Centroids, Medoids are formed and again distances are re-calculated to ensure that the documents who are closer to a medoid is assigned to the same cluster.

1. Describe the difference (pro and con) of k-means clustering and the LDA topic discovery model.

Both K-means and Latent Dirichlet Allocation (LDA) are unsupervised learningalgorithms, where the user needs to decide a priori the parameter K, respectively the number of clusters and the number of topics.  
  
 If both are applied to assign K topics to a set of N documents, the most evident difference is that K-means is going to partition the N documents in K disjoint clusters (i.e. topics in this case). On the other hand, LDA assigns a document to a mixture of topics. Therefore each document is characterized by one or more topics (e.g. Document D belongs for 60% to Topic A, 30% to topic B and 10% to topic E). Hence, LDA can give more realistic results than k-means for topic assignment.