Document Clustering

Anirban Chatterjee
Abhinav Chakraborty
Rohan Hore

Indian Statistical Institute,Kolkata

December 13, 2018

Contents

Overview

- Clustering is an automatic learning technique aimed at grouping a set of objects into subsets or clusters. The goal is to create clusters that are coherent internally, but substantially different from each other.
- Clustering is one of the well known unsupervised learning methods that can be used on the text data.Some possible directions of application of clustering to document data are
  - Finding Similar Documents
  - Organizing Large Document Collection
  - Duplicate Content Detection
  - Recommendation System
  - Search Optimization

Goals

- The goal of a document clustering scheme is to minimize intra-cluster distances between documents, while maximizing inter-cluster distances (using an appropriate distance measure between documents)
- Since,A distance measure (or, dually, similarity measure) lies at the heart of document clustering we need to mathematically formalize it.

Challenges

To extract information from the textual data is not a easy job.We list out a few of many limitations of working with textual data

- One of the main problems is ambiguity of the language *i.e.* same word can be interpreted in two or more possible ways.(*Polysemy*)
- In the other way,there can be multiple phrases associated with same meaning.(*Synonymy*)
- Sometimes in textual data occurrence of words like "btw","ppl" which don't exist in dictionary affect the results.
- Since,to work with textual data we break it up to word level,the textual data corresponds to a huge sparse data matrix,thus turning our algorithms computationally expensive.

Tokenization

- To work with text data,the first step is parsing the 'huge' document into smaller units(tokens) such as phrases or words.We will discuss the most commonly used tokenization method below.
- **Bag Of Words Model**:
  - Frequency or occurrence of a word is used as a feature to train.
  - Combining all the words appeared in all the documents,we form the **Vocabulary** to work with,say of size N.
  - Then i^{th} document in our collection is represented by a N-vector $\boldsymbol{w}$,whose i^{th} co-ordinate

$$\boldsymbol{w}_i = \#\{i^{th} \text{ word appears in the document}\}$$

  - The vector(or list) representation completely ignores the order of occurrence of the words.This is the main highlight of "BOW" model.

Corpus Cleaning

By **Corpus**,we refer to a large and structured collection of M documents denoted by $D = \{\boldsymbol{w}_1, \ldots, \boldsymbol{w}_M\}$

- **Stop words removal**: Usually we refer *most common words* as "Stop-words",which are filtered/removed before computing.In most of the cases,short function words such as "the","is","an","who" are referred as the stop words.
- With a purpose of cleaning corpus,we remove all the punctuation,numbers from the document too.
- We also turn all the characters to lower case so that,same word in different cases are not counted different.

Stemming and Lemmatization

- For grammatical reasons, documents are going to use different forms of a word, such as organize, organizes, and organizing. Additionally, there are families of derivation-ally related words with similar meanings, such as democracy, democratic, and democratization.
- **Stemming** usually refers to a crude heuristic process that chops off the ends of words in the hope of achieving this goal correctly most of the time, and often includes the removal of derivational affixes.
- **Lemmatization** usually refers to doing things properly with the use of a vocabulary and morphological analysis of words, normally aiming to remove inflectional endings only and to return the base or dictionary form of a word, which is known as the *lemma*.
- Examples:am,are,is $\rightarrow$ be ; car,cars,car's$\rightarrow$ car

Term Frequencies

- As noted in the BOW model,each document is represented by a N-vector containing the word frequencies.Now,combining the N-vector of frequencies corresponding to all the M documents,we can form matrices
  - **Document Term Matrix**:rows correspond to documents in the collection and columns correspond to terms,$(i, j)^{th}$ entry contains frequency of j^{th} term in i^{th} document.
  - **Term Document Matrix**:rows correspond to terms and columns correspond to the documents.Rest is clear form context.
- Using term frequency is just one of the choices,we will explore some of the other popular choices in coming slides.

Tf-Idf Weighting

- tf-idf or **term frequency-inverse document frequency** is a numerical statistic,intended to reflect how important a word is to a document in a corpus.
- The tf-idf value increases *proportionally* to the number of times a word appears in the document and is offset by the number of documents in the corpus that contain the word, which helps to adjust for the fact that some words appear more frequently in general.
- Some other weighting choices include:
  - Boolean Frequencies
  - Term frequency,adjusted for document length
  - Logarithmic-ally scaled frequency

Clustering methods

We will use

- Once,we have a notion of similarity or distance between documents,we can very well apply the well-known clustering algorithms.We here,will use
  - K-means clustering method
  - Hierarchical clustering method
- In next few slides,we are going to formalize the similarity between documents

Semantic Similarity

At this point,we want to find out semantic similarity between documents.With a goal of clustering documents,we would like to group similar documents.We start with some basic approaches and develop the theoretically involved ones.

- Since,we have a vector representation of documents,we can try to define suitable distance measures between those vectors.
  - **Cosine Similarity**:Cosine Similarity is probably the most common metric used in this purpose.Theoretically,it's just the cosine of angle between the vectors and defined by

$$Cosine(u, v) := \frac{\sum_i u_i v_i}{\sqrt{\sum_i u_i^2}\sqrt{\sum_i v_i^2}}$$

  - **Levenshtein Distance**:Also referred as *edit distance*, is the minimum number of single-character edits (insertions, deletions or substitutions) required to change one word into the other.

Topic Models

- Topic modeling is a frequently used text-mining tool for discovery of hidden semantic structures in a text body.
- Intuitively, given that a document is about a particular topic, one would expect particular words to appear in the document more or less frequently.A document can possibly concern multiple topics in different proportions.
- Here,for clustering purpose we are only worried about the semantic distance between the documents.

LSA

Introduction

- Latent semantic analysis (LSA) is a technique in natural language processing, of analyzing relationships between a set of documents and the terms they contain by producing a set of concepts related to the documents and terms.
- LSA uses *Singular Value Decomposition* to reduce the term-document matrix to a lower dimension, while preserving the inherent semantic relationship.

LSA

Overview

- LSA use a term-document matrix (usually weighted) as defined above.
- After the construction of the term-document matrix, LSA finds a low-rank approximation to the term-document matrix
- There could be various justifications for these approximations, one of the most important is the removal of synonymy. The rank lowering is expected to merge the dimensions associated with terms that have similar meanings.

LSA

Theoretical Justification

- Suppose C is matrix of rank r, and we want to construct the rank k($< r$) approximation of C.
- Now,if C_k is the rank k approximation obtained via SVD.

Theorem

$$\min_{Z|rank(Z)=k} ||C - Z||_F = ||C - C_k||_F$$

LSA

Derivation

- Suppose X=$\begin{bmatrix} x_{1,1} & \dots x_{1,j} & \dots x_{1,n} \\ \vdots & x_{i,j} & \vdots \\ x_{m,1} & x_{m,j} & x_{m,n} \end{bmatrix}$ denotes the term-document matrix (possibly weighted by tf-idf).
- Each row $t_i^\top = \begin{bmatrix} x_{i,1} \dots x_{i,j} \dots x_{i,n} \end{bmatrix}$ represents the term vector, giving its relation to each document.
- Each column $d_j^\top = \begin{bmatrix} x_{1,j} \dots x_{i,j} \dots x_{m,j} \end{bmatrix}$ represents the document vector, giving its relation to each term.

LSA

Derivation

- Consider the SVD of X, $X = U\Sigma V^\top$.
- For low-rank (k<r) approximation we consider the first k singular values. Then the k-rank approximation of X is

$$X_k = U_k \Sigma_k V_k^\top$$

- We consider the rows of U_k as the term vector and the columns of $V_k^\top$ as the document vector in the lower dimension.

LSA

Limitations

Although LSA is representationally simple,and easy to implement it has some serious limitations.

- LSA is unable to capture polysemy.
- There is no generative model
- Limitations with BOW model
- Loss of Interpretation

PLSA

Introduction

Probabilistic Latent Semantic Analysis is a novel statistical technique for the analysis of co-occurrence data, which has applications in information retrieval and filtering, natural language processing, machine learning from text, and in related areas. Compared to standard Latent Semantic Analysis which stems from linear algebra and performs a Singular Value Decomposition of co-occurrence tables, the proposed method is based on a mixture decomposition derived from a latent class model. This results. in a more principled approach which has a solid foundation in statistics.

PLSA

Aspect Model

The starting point for Probabilistic Latent Semantic Analysis is a statistical model which has been called aspect model. The aspect model is a latent variable model for co-occurrence data which associates an unobserved class variable $z \in \mathcal{Z} = \{z_1, \ldots, z_K\}$ with each observation. A joint probability model over $\mathcal{D} \times \mathcal{W}$ is defined by the mixture

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

PLSA

Aspect Model

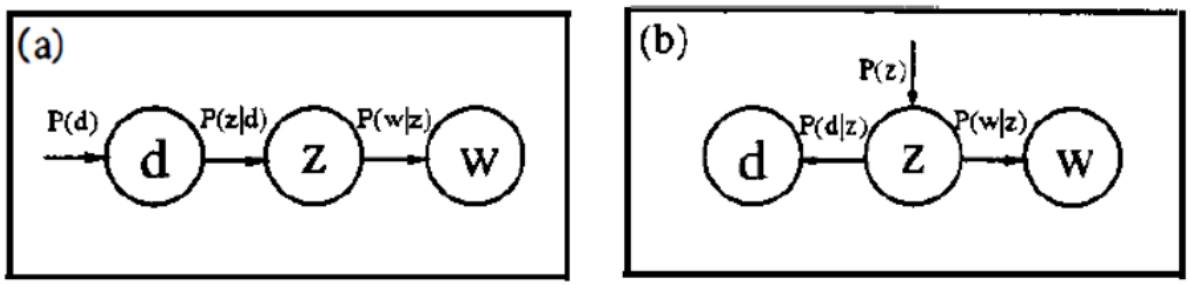
The aspect model introduces a conditional independence assumption, namely that d and w are independent conditioned on the state of the associated latent variable(the corresponding graphical model representation is depicted in Figure 1 (a)). Since the cardinality of z is smaller than the number of documents/words in the collection, z acts as a bottleneck variable in predicting words. It is worth noticing that the model can be equivalently parameterized by (see fig (b))

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

PLSA

Aspect Model

Graphical model representation for the equivalent parameterization



PLSA

Likelihood

- The likelihood of observed data can be written as

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

  where $n(d,w)$ measures the frequency of word w in document d.
- The log-likelihood can be now written as

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

PLSA

Model Fitting with EM

The standard procedure for maximum likelihood estimation in latent variable models is the Expectation Maximization (EM) algorithm. EM alternates two coupled steps:

- an expectation (E) step where posterior probabilities are computed for the latent variables
- an maximization (M) step, where parameters are updated

PLSA

Model Fitting with EM

Standard calculations yield the E-step equation as

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

as well as the following M-step formulae

$$P(w|z) \propto \sum_{d \in \mathcal{D}} n(d,w)P(z|d,w)$$

$$P(d|z) \propto \sum_{w \in \mathcal{W}} n(d,w)P(z|d,w)$$

$$P(z) \propto \sum_{d \in \mathcal{D}} \sum_{w \in \mathcal{W}} n(d,w)P(z|d,w)$$

PLSA

Relation with LSA

- Recall the model

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

- The joint probabilities can be interpreted as follows

$$P = U\Sigma V^{\top} \text{ With } U, V, \Sigma \text{ defined as}$$

$U_{d,z}$ contains the probabilities $P(d|z)$

$V_{w,z}$ contains the probabilities $P(w|z)$

Σ is a diagonal matrix of prior probabilities P(z)

PLSA

Limitations

- The number of parameters grows linearly with the size of training documents.
- Although PLSA is a generative model of the documents in the collection it is estimated on, it is not a generative model of new documents.

LDA

Motivation

Below follows some sentences and topics assigned by LDA,when asked for 2 topics

- I like to eat broccoli and bananas. (100% topic A)
- I ate a banana and spinach smoothie for breakfast.(100% topic A)
- Chinchillas and kittens are cute.(100% topic B)
- My sister adopted a kitten yesterday.(100% topic B)
- Look at this cute hamster munching on a piece of broccoli.(60% topic A,40% topic B)

LDA

Motivation

Interpretations of the topics can be given as follows

- *Topic A:* 30% broccoli, 15% bananas, 10% breakfast, 10% munching,. . .(**Vegetable**)
- *Topic B*: 20% chinchillas, 20% kittens, 20% cute, 15% hamster,. . .(**Animals**)

LDA

Introduction

- In natural language processing, latent Dirichlet allocation (LDA) is a generative statistical model that allows sets of observations to be explained by unobserved groups that explain why some parts of the data are similar.
- For example, if observations are words collected into documents, it posits that each document is a mixture of a small number of topics and that each word's presence is attributable to one of the document's topics.

LDA

Overview

- Here each document is considered to have a set of topics that are assigned to it via LDA.
- This is identical to probabilistic latent semantic analysis (PLSA), except that in LDA the topic distribution is assumed to have a Dirichlet prior.

LDA

Notation and Terminology

- We represent words by unit-basis vectors that have a single component equal to one and all others equal to zero. So the i^{th} word in vocabulary V would be a vector w such that $w^i = 1$ and $w^j = 0 \quad \forall j \neq i$.
- A document is a sequence of N words denoted by $\boldsymbol{w} = \{w_1, w_2 \ldots, w_N\}$, where w_n is the n^{th} word in the sequence.
- A corpus is a collection of M documents denoted by $D = \{\boldsymbol{w_1}, \boldsymbol{w_2} \ldots, \boldsymbol{w_M}\}$

LDA

Model

LDA assumes the following generative process for each document $\boldsymbol{w}$ in a corpus D.

- Choose $N \sim \text{Poisson}(\xi)$
- Choose $\theta \sim \text{Dir}(\alpha)$
- For each of the N words w_n:
  - Choose a topic $z_n \sim \text{Multinomial}(\theta)$
  - Choose a word w_n from $p(w_n|z_n, \beta)$, a multinomial probability conditioned on topic z_n.
- The word probabilities are parametrized by a kxV matrix β, where $\beta_{ij} = p(w^j = 1|z^i = 1)$.

LDA

Model

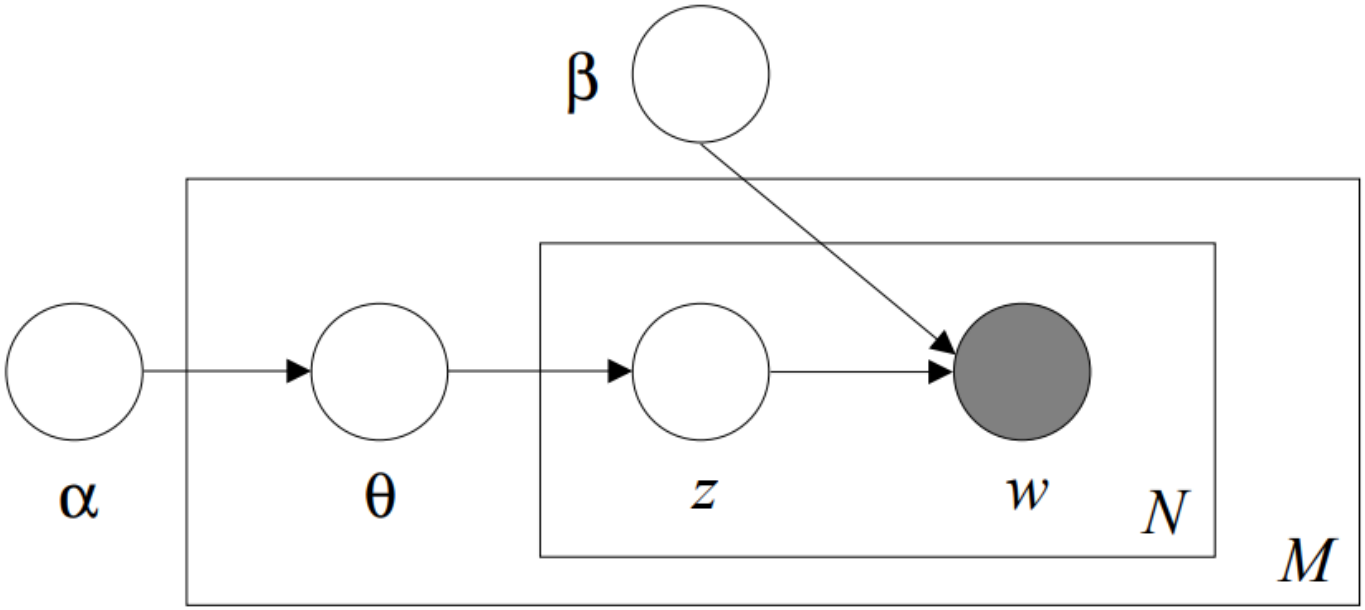


Figure: Graphical Model Representation of LDA. The outer plate represents documents, while the inner plate represents choice of topics and words within a document

LDA

Probability of a Corpus

We would use the above generative model to find the probability of observing a corpus.

- $$p(\theta|\alpha) = \frac{\Gamma(\sum_{i=1}^{k}\alpha_i)}{\prod_{i=1}^{k}\Gamma(\alpha_i)}\theta_1^{\alpha_1-1}\ldots\theta_k^{\alpha_k-1}$$

  . where θ is a k-dimensional Dirichlet Random Variable.
- Given parameters α and β, the joint distribution is given by

$$p(\theta, \boldsymbol{z}, \boldsymbol{w}|\alpha, \beta) = p(\theta|\alpha)\prod_{n=1}^{N} p(z_n|\theta)p(w_n|z_n, \beta)$$

LDA

Probability of a Corpus

- Then the marginal distribution of a document becomes

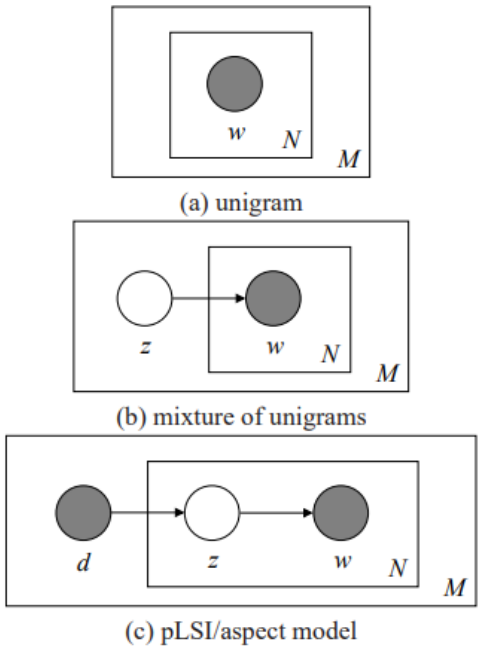
$$p(\boldsymbol{w}|\alpha,\beta) = \int p(\theta|\alpha)\left(\prod_{n=1}^{N}\sum_{z_n} p(z_n|\theta)p(w_n|z_n,\beta)\right)d\theta$$

- The probability of a corpus is given by

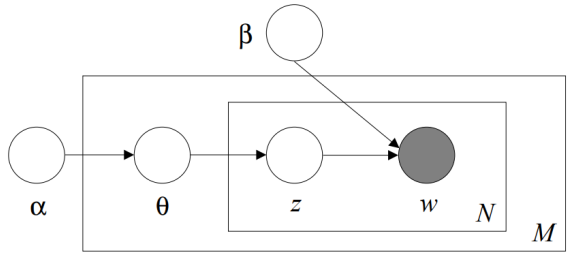
$$p(D|\alpha,\beta) = \prod_{d=1}^{M}\int p(\theta_d|\alpha) \left(\prod_{n=1}^{N_d}\sum_{z_{dn}} p(z_{dn}|\theta_d)p(w_{dn}|z_{dn},\beta)\right)d\theta_d$$

LDA

Comparison with other models



(a) Graphical representation of different models



(b) Graphical representation of LDA

LDA

Comparison with other models

- **Unigram Model:** The words in every document is drawn from a single multinomial distribution.

$$p(\boldsymbol{w}) = \prod_{i=1}^{N} p(w_n)$$

- **Mixture of Unigrams:** Here each document is generated by first choosing a topic and generating N words from the conditional multinomial $p(w|z)$

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

- **Probabilistic Latent Semantic Ananlysis**

$$p(d, w_n) = p(d) \sum_{z} p(w_n|z)p(z|d)$$

LDA

Estimation

- The main inferential problem is to compute

$$p(\theta, \boldsymbol{z}|\boldsymbol{w}, \alpha, \beta) = \frac{p(\theta, \boldsymbol{z}, \boldsymbol{w}|\alpha, \beta)}{p(\boldsymbol{w}|\alpha, \beta)}$$

- This distribution is intractable to compute in general because of the coupling between θ and β.

$$p(\boldsymbol{w}|\alpha, \beta) = \frac{\Gamma(\sum_i \alpha_i)}{\prod_i \Gamma(\alpha_i)} \int \left(\prod_{j=1}^{k} \theta_i^{\alpha_i - 1} \right) \left(\prod_{n=1}^{N} \sum_{i=1}^{k} \prod_{j=1}^{V} (\theta_i \beta_{ij})^{w_n^j} \right) d\theta$$

LDA

Variational Inference

- Although the posterior distribution in intractable for inference,we can use a wide variety of approximate inference algorithms.
- The basic idea of variational inference is to make use of Jensen's Inequality to obtain an adjustable tight lower bound to the log-likelihood.
- A way to obtain tractable family of lower bounds is to consider simple modifications of the original graphical model in which some of the edges and nodes are removed.

LDA

Variational Inference

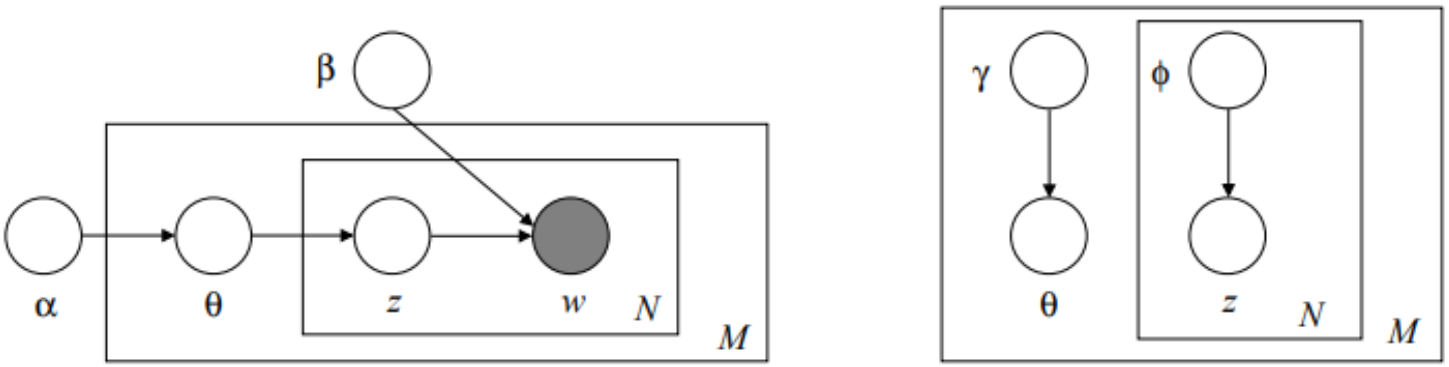


Figure: (Left) Graphical Model Representation of LDA. (Right) Graphical Model Representation of the variational distribution

LDA

Variational Inference

- From the resulting simplified graphical model,with free variational parameters,we obtain a family of distribution on the latent variables

$$q(\theta, z|\gamma, \phi) = q(\theta|\gamma) \prod_{n=1}^{N} q(z_n|\phi_n)$$

where,the Dirichlet parameters γ and multinomial parameters $(\phi_1, \ldots, \phi_N)$ are free varaitional parameters.

LDA

Variational Inference

- One can show,finding a tight lower bound on the log-likelihood translates directly into the following optimization problem:

$$(\gamma^*, \phi^*) = arg \min_{\gamma,\phi} D(q(\theta, \boldsymbol{w}|\gamma, \phi)||p(\theta, \boldsymbol{z}|\boldsymbol{w}, \alpha, \beta)$$

where D is KUllback-Leibler Divergence between variational distribution and true posterior.

LDA

Parameter Estimation

We use a alternating variational EM procedure,as explained below

- **E-step**:For each document,find the optimizing values of the variational parameters $\{\gamma_d^*, \phi_d^* : d \in \mathcal{D}\}$This is done as discussed before
- **M Step**: Maximize the resulting lower bound on the log- likelihood w.r.t the model parameters α and β.

The two steps are repeated until the lower bound on the log-likelihood converges.

LDA

Clustering

- As long as we get hold of the estimate of posterior,we have the probabilities $p(\theta|\boldsymbol{w})$.This probability vector(document-topic vector) can be used as a representation for the document for normal clustering methods
- We can use **Jensen-Shannon Divergence** as a notion of dissimilarity between the document-topic vectors.

$$JSD(P||Q) = \frac{1}{2}D(P||M) + \frac{1}{2}D(Q||M)$$

with $M = \frac{P+Q}{2}$

LDA

Clustering

- We can also possibly use,euclidean distance between the one-coordinate removed document-topic vector
- Another suggestion is if we choose the topic with maximum posterior probability $p(\theta|\boldsymbol{w})$ and report simply that as our cluster id.Here,some prior belief on the no of topics will possibly be helpful.

Evaluation Measure

- Measuring clustering accuracy corresponds to measuring how much "internally consistent" a cluster is.In presence of ground truth measures,we can compute measures of discrepancy/similarity between estimated clustering and ground-truth partition.
  - Rand Index
  - Normalized Mutual Information(NMI)

Rand Index

- Given a set $S = \{\sigma_1, \sigma_2, \ldots, \sigma_n\}$,say we have two partitions of S,namely $X = \{X_1, X_2, \ldots, X_r\}$,partition into r subsets and $Y = \{Y_1, Y_2, \ldots, Y_s\}$,partition into s subsets.If We define,a as the number of pairs of elements in S that are in the same subset in X and in the same subset in Y, b as the number of pairs of elements in S that are in the different subset in X and in the different subset in Y
- The **Rand Index**,R is defined as

$$\frac{a+b}{\binom{n}{2}}$$

- It lies within 0 and 1,while 0 indicating the clusters don't agree on any pairs of points and 1 indicates the clusters match exactly

Normalized Mutual Information

- X= the Class Labels
- Y=the Cluster Labels
- **Normalized Mutual Information**(NMI) can be defined as

$$NMI(X, Y) = \frac{I(X; Yf)}{min(H(x), H(Y))}$$

Data-set

- All of the methods will be now applied on a data,collected from "MetroLyrics Data Link"
- There are around 3,80,000+ lyrics in the data set from a lot of different artists from a lot of different genres arranged by year. Every artist folder has a genre.txt file that tells what is the genre of the musician.

Data-set

1	then-tell-me	2009	beyonce-knowles	Pop	playin' everything so easy, it's like you seem so sure. still your ways, you dont see i'm not sure if they're for me. then things come right along our way, though we didn't truly ask. it seems as if they're gonna linger with every delight they bring, just like what you have truly seemed. i'm trying to think of what you really want to say, even through my darkest day. you might want to leave me, feeling strange about you like you're gonna let me know, when words then slipped out of you. when words dont come so easy to say you just leave me feeling, come what may though i want things coming from your way. i say to you, you bore me all the time when you seem to hold back all in you, all that you want to let me know. why dont you have the courage? speak up and i'll listen, if you truly want me to know, then tell me. is there something wrong with you and you seem fastened there. it sounds as if there'll be a melody if things in you are let out and then i will feel alright. when you sleep, do you feel the same, exactly as i do? i really want to hear things from you, though i've felt something new eversince you acted that way. if i go, would you still mind telling me? if i stay, you seem to let the days go by. if you truly want to let me know, then tell me.

Figure: One typical lyrics

Work Done

- We took input as the whole dataset and used stratified sampling to select our working dataset.
- Used inbuilt corpus cleaning to remove punctuations and stopwords.
- Used the above stated tf-idf weighting scheme for term-document matrix.
- Looked at a wordcloud to identify few important words.

Genre representation

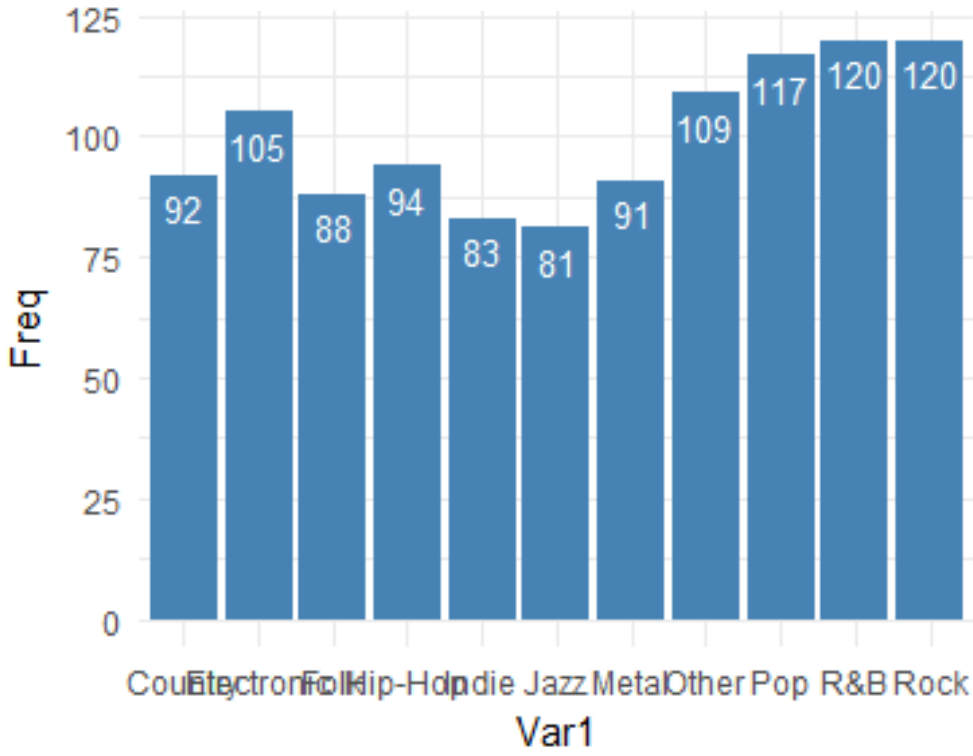


Figure: Representation of Genres

Work Done

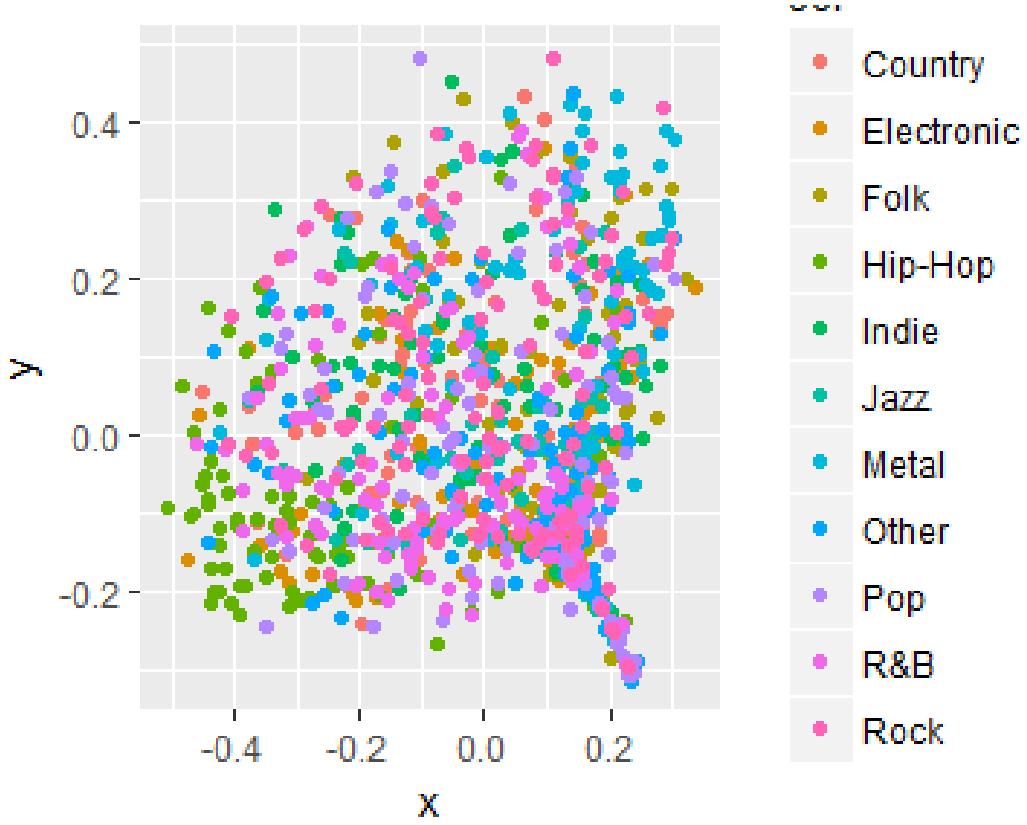


Figure: Actual Data

Work Done

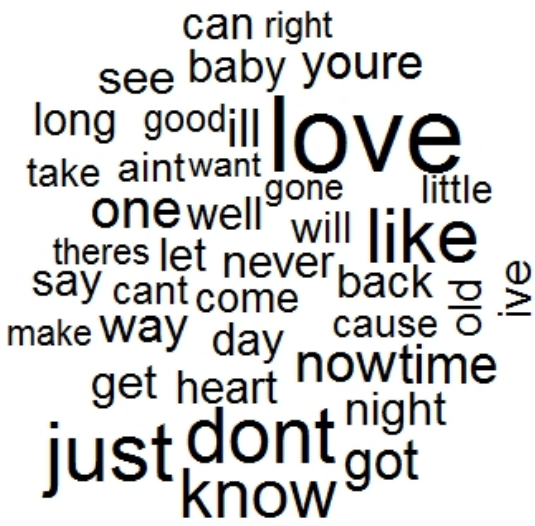


Figure: Most Important Words in Country Genre

Work Done

Figure: Most Important Words in Hip-Hop Genre

Work Done

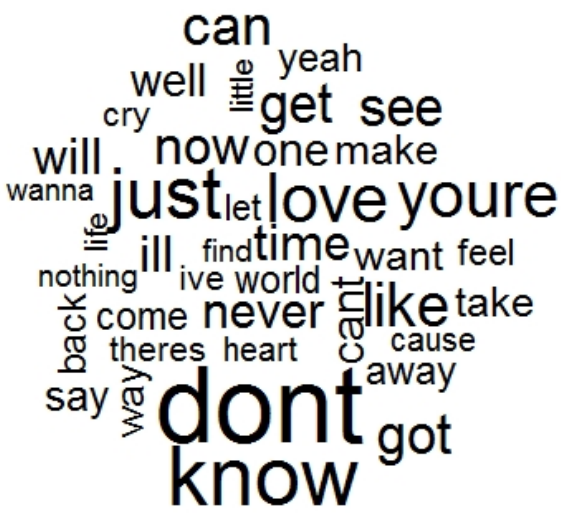


Figure: Most Important Words in Rock Genre

Work Done

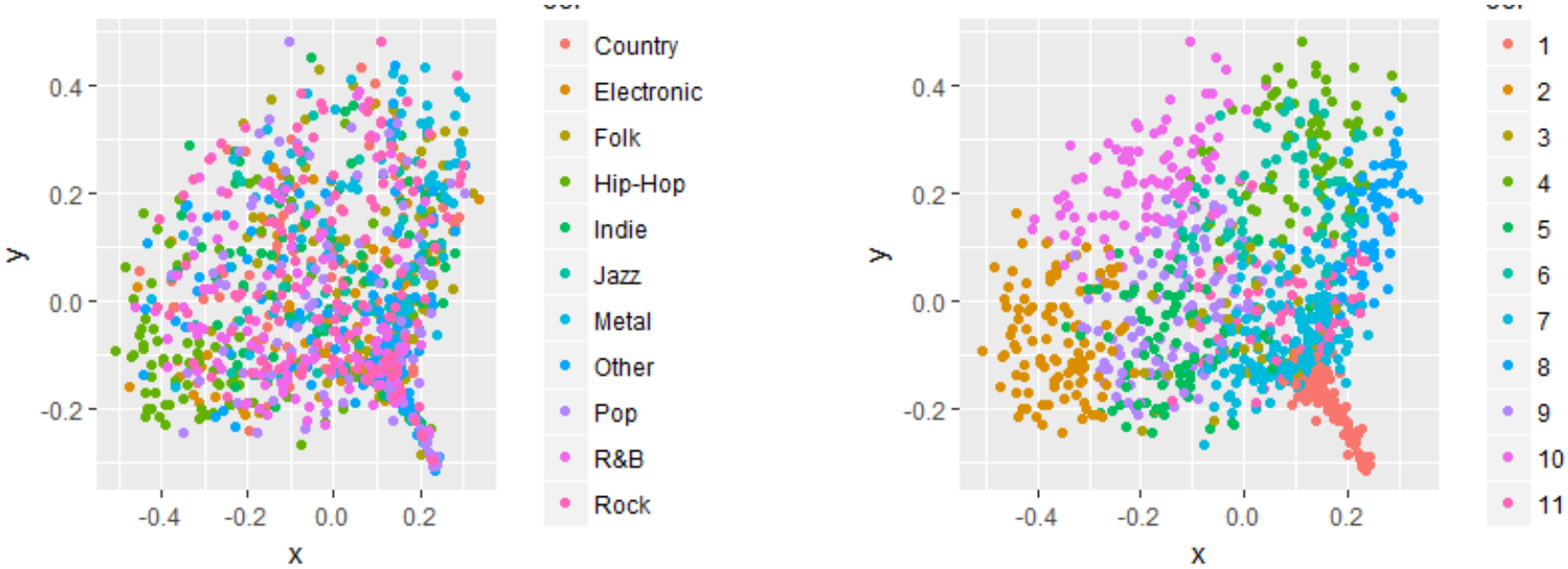
Figure: Most Important Words

Work Done

LSA

- Using inbuilt lsa function in R we implemented LSA.
- Used k-means clustering on the lower dimensional approximations.

LSA K -means



(a) Actual Clusters

(b) k-means Cluster

Figure: Comparisons

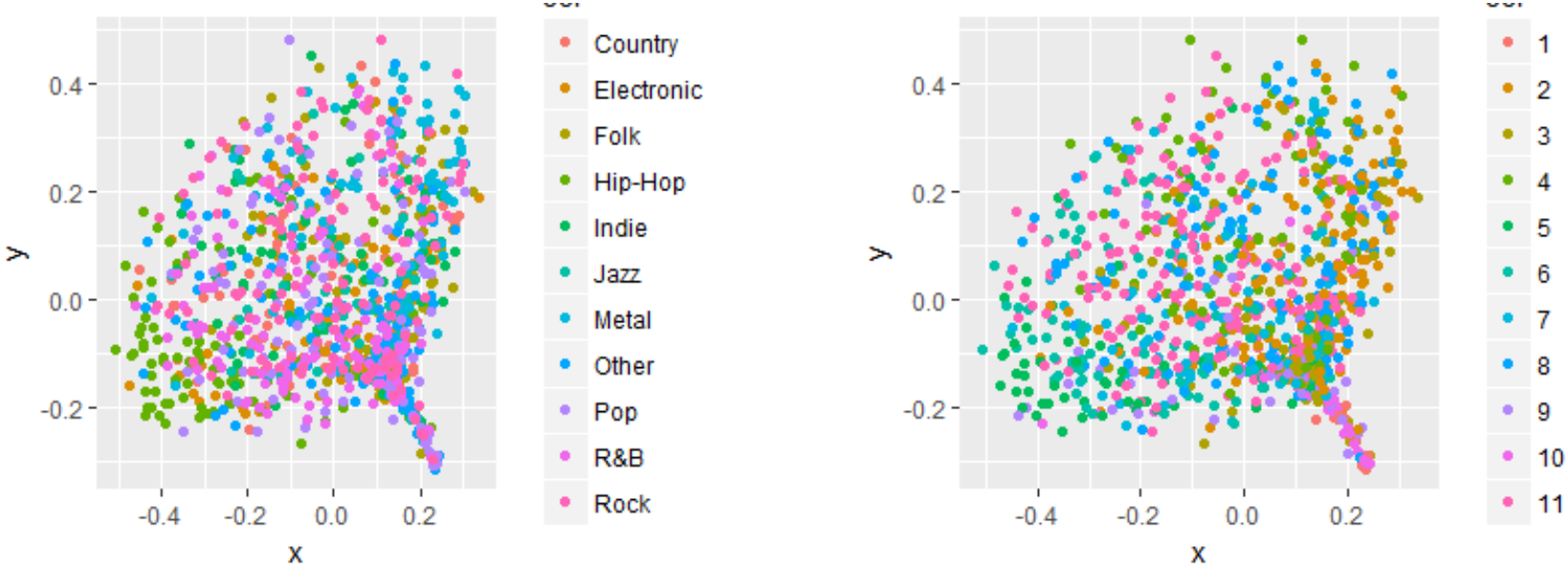
Work Done

LDA

- We used LDA function in topicmodels package to implement LDA in R
- As an output we received the posterior probability vector $P(\theta|\boldsymbol{w})$ for each document.
- Then used JS divergence as a similarity measure for clustering
- We also used the maximum posterior probability for clustering and compared both methods.

Work Done

LDA

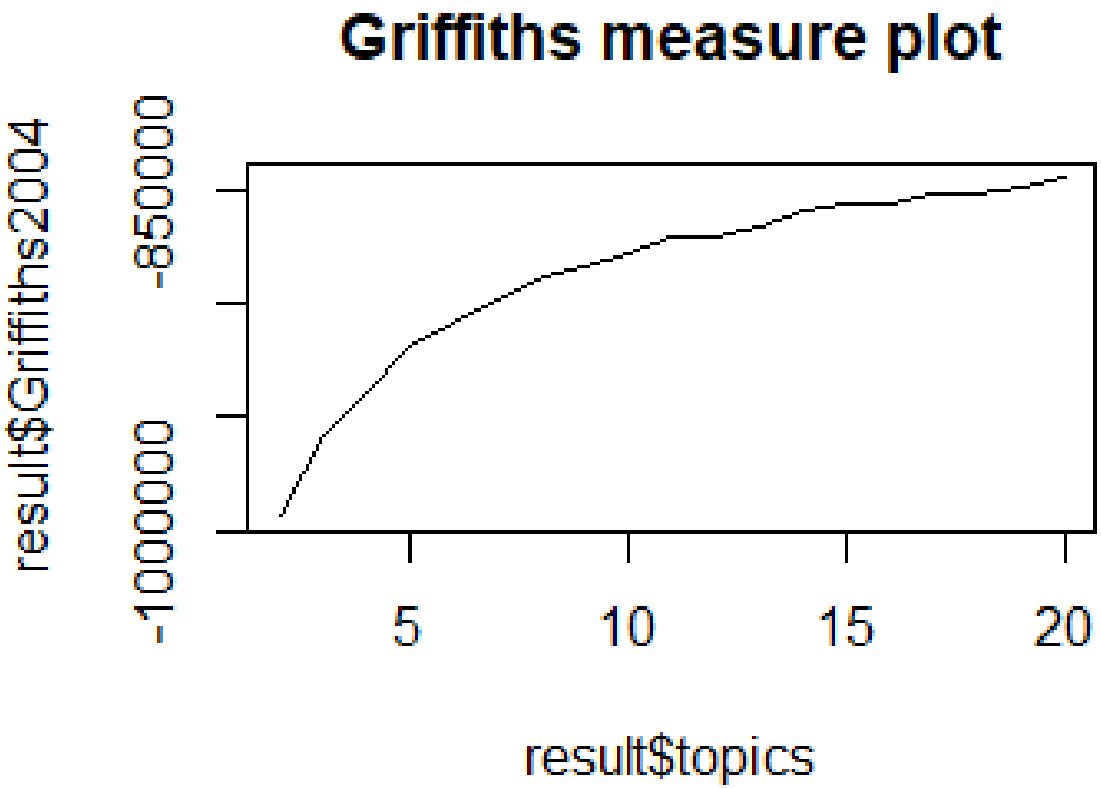


(a) Actual Clusters

(b) LDA Clusters

Figure: Comparison

Choosing no of topics



Results

Method	Rand Index
LSA K-Means	0.8274845
LDA k-means(JS)	0.81754
LDA k-means(Euclidean)	0.81536
LDA MAP predictor	0.8294488

Table: Comparison between the methods

Comments

- As a future work,we can possibly implement "non-negative matrix factorization",recent topic models
- Genre is not identified only by lyrics,but also is identified by its background score,instruments used etc.

References I

David M Blei, Andrew Y Ng, and Michael I Jordan. "Latent dirichlet allocation". In: *Journal of machine Learning research* 3.Jan (2003), pp. 993–1022.

Thomas Hofmann. "Probabilistic latent semantic analysis". In: *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*. Morgan Kaufmann Publishers Inc. 1999, pp. 289–296.

The End