Anchor Free Correlated Topic Modelling

Project Report: Spring, 2019
CS 519: Matrix Analysis and Signal Processing

Rogen George
Oregon State University, Corvallis OR, USA

Every industry in today's day and age generates an unprecedented amount of data, it has become increasingly imperative to obtain "Information" from the data. With the growing amount of data in recent years, most of which is unstructured, it's difficult to obtain the relevant and desired information. One technique used to mine through the data and fetch the information that we are looking for is Topic Modelling. Topic Models are very useful for the purpose for document clustering, organizing large blocks of textual data, information retrieval from unstructured text and feature selection. In this report, we go over the different methodologies of building text models, such as, LDA, LDA with Gibbs sampling, and Correlated Topic Modelling. We further describe and implement Anchor Free Correlated Topic modelling. We compare the results of Anchor Free Topic modelling with LDA which is a popular Topic modelling algorithm.

I. Introduction

In machine learning and natural language processing, a topic model is a type of statistical model for discovering the abstract "topics" that occur in a collection of documents. Originally developed as a text-mining tool, topic models have been used to detect instructive structures in data such as genetic information, images, and networks. They also have applications in other fields such as bioinformatics. Topic modeling is a frequently used data mining technique for discovery of hidden semantic structures in a text body. For example, when a document is about a particular topic, certain words would appear more frequently than others: "dog" and "cat" might appear more often in documents about animals, "car" and "motorcycle" will appear more often in documents about vehicles. There are many layers to extracting meaningful topics from a given document: a document that contains only one occurrence of the word "political" may still have the overarching theme of politics. A topic model captures these different methods of extracting topics in a mathematical framework, which allows examining a set of documents and discovering, based on the underlying statistics of the words in each, what the topics might be and what each document's balance of topics is.

Each document in a given corpus can be represented by a matrix containing the occurrence of words, where it is modelled by a distribution over a certain number of topics, each of which is a distribution over words in a given vocabulary.

$$D \approx CW$$

where D is the document corpus, C is the word-topic vertical and W is the weight associated with it.

By learning the distributions, a corresponding low-rank representation of the high-dimensional matrix can be obtained for each document. In doing so, topic modelling can largely be considered as a matrix factorisation problem. In the past decades, many methods have been devised to solve this problem: such as Latent Semantic Analysis (LSA) [3], Probabilistic Latent Semantic Analysis (PLSA) [2], Latent Dirichlet Allocation (LDA) [1], and Correlated Topic Modelling [4].

Identifiability or the discovery of unique topics across a document is an important problem space in topic modelling. Identifiability can be obtained when the topics within the documents are uncorrelated, and consequently also requires more assumptions. Employing models based purely on PMFs inherently makes the model less susceptible to the relations between topics in a given document, which we observe in approaches such as LDA. This doesn't pose so much of a problem in Correlated topic models(such as certain variations of NMF) since they are built on the foundation of topic-topic, word-topic correlation matrices.

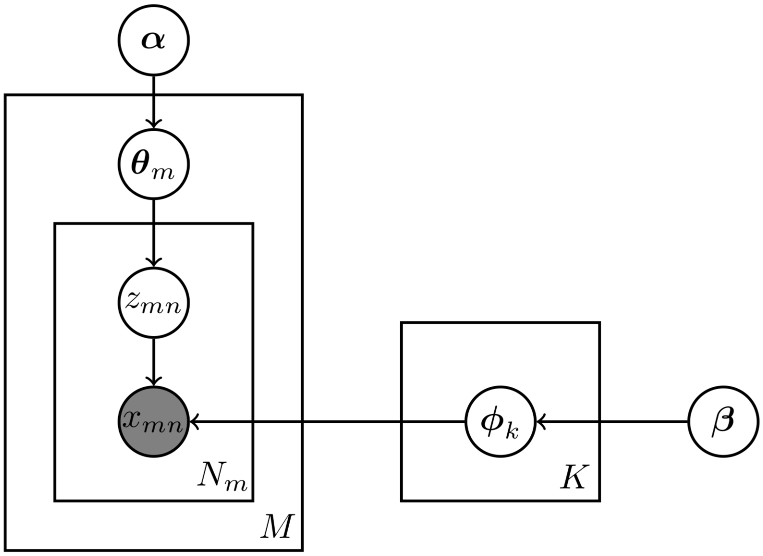

Fig. 1 Latent Dirichlet Allocation

A. Latent Dirichlet Allocation

The foundation for topic modelling, as we see it today, was laid down largely with the ascent of Latent Dirichlet Allocation [1]. It is an extension of Probabilistic Latent Semantic Analysis (PLSA) [2] with a very minute difference in terms of how they treat per-document distribution. There exist three primary components to LDA:

1) **Latent:** This refers to everything that we don't know apriori and is 'latent' or 'hidden' in the data. Here, the topics that the document consists of are unknown, but they are believed to be present as the text is generated based on those topics.
2) **Dirichlet:** LDA model uses Dirichelt priors for the topic word distributions and document topic distributions. Essentially, Dirichlet refers to a 'distribution of distributions'. Using an example to illustrate this concept, let's suppose there is a machine that produces dice and we can control whether the machine will always produce a dice with equal weight to all sides, or will there be any bias for some sides. Basically, the machine producing dice is a distribution as it is producing dice of different types. Also, we know that the dice itself is a distribution as we get multiple values when we roll a dice: this is what it means to be a distribution of distributions. In context of topic modeling, the Dirichlet is the distribution of topics in documents and distribution of words in the topic.
3) **Allocation:** Allocation, in this context, refers to the allocation of topics to the documents and words of the document to topics, given the Dirichlet.

LDA is a generative topic model for documents, and inherently follows the Bayesian statistical model. Mathematically, the word probabilities are parameterized by a K X M matrix β. A topic k ($1 \leq$ k $\leq$ K) is a discrete distribution over words with probability vector β_k. Each document d_j maintains a separated distribution θ_j that describes the contribution of each topic. Implementations of LDA inference algorithms typically use symmetric Dirichlet prior over $\theta = \theta_1$, . . . , θ_D , in which the concentration parameter α is fixed. A topic distribution of a document d_j and a word w_i is associate in a distribution variable $z_{j,i}$. Figure 1 represents the dynamics of LDA, as has been described here.

LDA can be interpreted to be similar to matrix factorization [7] or SVD, where we decompose the probability distribution matrix of word in document in two matrices consisting of distribution of topic in a document and distribution of words in a topic. Non-negative Matrix Factorization in the context of LDA can be applied with two different objective functions: the Frobenius norm, and the generalized Kullback-Leibler (KL) divergence. The latter is equivalent to Probabilistic Latent Semantic Indexing.

The value of variational parameters are chosen by a optimization procedure that attempts to minimizing the KL-divergence between the variational distribution and the true posterior p(θ, z, w|α, β). Since it is not possible to minimize the KL-divergence directly, we use a bound over the log likelihood of a document, p(w|α, β). Further, using Jensen's inequality, it is possible to show that minimizing the KL-divergence between the variational distribution and

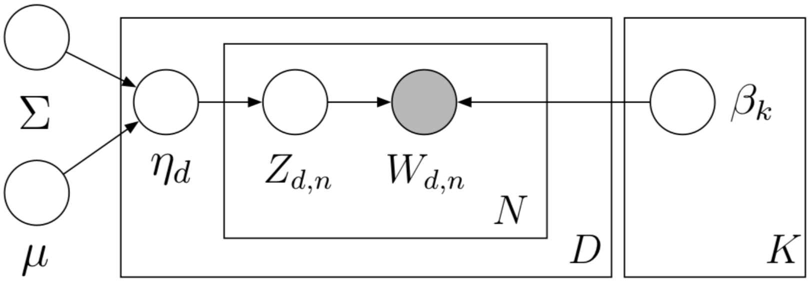

Fig. 2 Correlated Topic Modelling

the true posterior distribution is equivalent to maximizing the difference between the variational expectation of real posterior distribution and the variational distribution, which forms our primary goal.

Taking into account another perspective, where we need to estimate the weights (W) corresponding the the topic-words, topic-document and word-documents, poses the important question of how to learn the weights of the C and W two matrices. To start with, we can randomly assign weights to both the matrices. To identify the correct weights, we propose using an algorithm called Gibbs sampling. Let's now understand what Gibbs sampling is and how does it work in LDA.

B. Gibbs Sampling

Gibbs sampling is an algorithm for successively sampling conditional distributions of variables, whose distribution over states converges to the true distribution in the long run [5]. Essentially, we're finding the conditional probability distribution of a single word's topic assignment conditioned on the rest of the topic assignments, that is, the conditional probability equation for a single word w in document d that belongs to topic k:

$$p(z_{d,n} = k | z_{d,n}, w, \alpha, \lambda) = \frac{n_{d,k} + \alpha_k}{\sum_i^K n_{d,i} + \alpha_i} * \frac{v_{k,w_{d,n}} + \lambda_{w_{d,n}}}{\sum_i^K v_{k,i} + \lambda_i}$$

where n(d,k): Number of times document d use topic k, v(k,w): Number of times topic k uses the given word, αk: Dirichlet parameter for document to topic distribution, λw: Dirichlet parameter for topic to word distribution.

There are two parts two this equation. The first part tells us how much each topic is present in a document and the second part tells how much each topic likes a word. Note that for each word, we will get a vector of probabilities that will explain how likely this word belongs to each of the topics. In the above equation, it can be seen that the Dirichlet parameters also act as smoothing parameters when n(d,k) or v(k,w) is zero which means that there will still be some chance that the word will choose a topic going forward.

C. Correlated Topic Modelling

Correlated Topic Model (CTM) is a kind of statistical model, and is essentially an extension of LDA [4]. The key to CTM is in its logistic normal distribution. It uses a logistic normal prior over topic assignments. The Dirichlet is a distribution on the simplex, where positive vectors sum up to 1. It assumes that components are nearly independent. This allows topic occurrences to exhibit correlation. CTM also provides a "map" of topics and how they are related. It also provides for a better fit to text data, although computation is more complex. CTM models the fact that in real data, an article about fossil fuels is more likely to also be about geology than about genetics. We use the logistic normal to model the dependence between components. The log of the parameters of the multinomial are drawn from a multivariate Gaussian distribution,

$$X \approx \mathbf{N}_K(\mu, \sigma)$$

$$\theta_i = expx_i$$

Figure 2 illustrates correlated topic modelling, where μ and σ depict the logistic normal prior.

Other extensions of LDA include, but are not limited to, Dynamic topic model, that are models that learn topic changes over time, and Polylingual topic models, that learn the correlations between topics that are impossible to capture using a single Dirichlet across multiple languages [4].

II. Anchor Free Topic Modelling

A. Anchor Word Assumption

The anchor word assumption refers to the separability of topics present in a document. There exists V = v_1, ..., v_F such that C(V,:) = Diagonal(c), where c belongs to RF. We can reframe our original problem statement $D \approx CW$ as, $D \approx C(V, :)W$. This implies that our task is to find v_1, ..., v_F, which we can call "anchor" words for the document, following which C can be estimated using an optimisation strategy.

B. Anchor Free Approach

Anchor Free Topic Modelling [7] starts off with the topic correlation concept. First, we make the use of an anchor-free topic identification criterion. This criterion aims at factoring the word-word correlation matrix using a word-topic Probability Mass Function (PMF) matrix and a topic-topic correlation matrix via minimizing the determinant of the topic-topic correlation matrix. We use the sufficiently scattered condition, which is much milder than the anchor-word assumption. The two matrices can be uniquely identified by the proposed criterion. Moreover, the proposed approach does not need to resort to higher-order statistics tensors to ensure topic identifiability, and can naturally deal with correlated topics. Second, a simple procedure for handling the proposed criterion that only involves eigen-decomposition of a large but sparse matrix, with linear programming.

We will now go over the different parts in detail, complete with its assumptions, problem formulation and final implementation.

C. Problem Formulation

The basis of our problem formulation lies in $D \approx CW$. The matrix D can tend to be very noisy, primarily due to the lack of representing relationships between words, which eventually leads to repetitive and redundant information. This led to formulating the problem space using second order statistics, where we can map correlations between words and topics:

$$P = E(DD^T) = CEC^T$$

where $E = E(WW^T)$ is essentially a topic-topic correlation matrix, and P is word-word correlation matrix.

This is a neat idea, given that we have P, formed using the PMF of words in the topics that they are found in. However the matrix decomposition of such a matrix is non-identifiable, which is a problem. Moreover, trying to solve the above problem directly is computationally heavy and not viable. The alternative approach is to use the 'self expressiveness' of P: by assuming stochasticity of D, we can rethink our problem from the perspective of sparse block optimisation.

$$min|detE|$$

where $E \in R^{FXF}, and, C \in R^{VXF}$, such that $P = CEC^T, C^T 1 = 1, C \geq 0$.

The optimal solutions of the aforementioned problem are the ground truth C and E, provided that the anchor-word assumption is satisfied. What is interesting, however, is that the optimal solutions for the given problem are valid and identifiable even when the anchor-word assumption is not satisfied.

D. Sufficiently Scattered Condition

The "Sufficiently Scattered Condition" is based on the geometric interpretation of NMF, and works on the principle of the solutions being nearly, but not entirely, separable. The condition can be summarised as below: Let $cone(CT)$ denote the polyhedral cone $x : Cx \geq 0$ and K denote the second-order cone $x : ||x||_2 < 1^T x$. Matrix C is called sufficiently scattered if it satisfies:

$$cone(CT) \in K, andcone(CT) \cap bdK = \lambda e_f : \lambda \geq 0, f = 1, ..., F$$

where bdK denotes the boundary of K, i.e. $bdK = x : ||x||_2 = 1^T x$.

The consequences of this condition leads to the observations:

1) Any feasible solution of our problem matrix is full rank, which implies that |det E| > 0, and ensures non triviality of our solution space.
2) An arbitrary square matrix E, given $P = CEC^T$ where C and E can be identified up to permutation via solving our original optimisation problem.

For formulating our problem space, the perspective of dual cones is quite useful. The notation of our polyhedral cone comes from the fact that it is the dual cone of the conic hull of the row vectors of C. A useful property of dual cone is that for two convex cones K_1 and K_2, if $K_1 \in K_2$, then $K_{1*} \in K_{2*}$ which means the first requirement of our Sufficiently Scattered Condition is equivalent to $K \in cone(CT)$. Geometrically, this significantly reduces the restrictions on our problem space.

E. Anchor Free Algorithm

The problem space we obtained in (Problem Formulation) involves minimising the determinant. There are several ways to tackle it. Let's consider optimization of the following:

$$minimize||P - CEC^T||_F^2 + \mu|detE|$$

subject to $C \geq 0$, $C^T 1 = 1$, where $\mu \geq 0$. This condition balances the data fidelity and the minimal determinant criterion. However, term CEC^T makes this optimization problem tri-linear. It cannot be easily decoupled. Finding a good μ to tune in is difficult too. The easier procedure of handling minimization of the determinant is summarized in the algorithm below, which is called as Anchor Free. Note that matrix P is symmetric and positive semidefinite. To solve $P = BB^T$, we apply square-root decomposition, where $B \in R^{VXF}$. Eigen decomposition of sparse matrices can be easily computed using available strategies. We have, $B = CE^{1/2}Q$, $Q^T.Q = Q.Q^T = I$, and $E = E^{1/2}.E1/2$ The coefficients of $CE^{1/2}$ must be orthonormal because P is symmetric. Also,

$$minimize|detE^{1/2}Q|$$

subject to $B = CE^{1/2}Q, C^T 1 = 1, C >= 0, Q^T Q = I$ has the same optimal solution as stated in section(B). Since Q is unitary, it does not affect the determinant. Therefore, letting $M = Q^T E^{-1/2}$, the optimization problem translates to,

$$maximize|detM|$$

subject to $M^T B^T 1 = 1, BM \leq 0$. Now, C has been marginalized and we have only F^2 variables left. This value is significantly smaller as compared to the original variable size of $VF + F^2$. (Here, V is the vocabulary size.)

We apply co-factor expansion to deal with the determinant objective function. If we fix all the columns of M except the fth one, determinant M becomes a linear function wrt M(:,f). We can write it as

$$detM = \Sigma_{k=1}^{F}(-1)^{f+k}M(k, f)$$

$$detM_{k,f} = a^T M(:, f)$$

where $a = [a_1, ..., a_F]^T, a_k = (-1)^{f+k}detM_{k,f}, \forall k = 1, ..., F$, and $M_{k,f}$ is a matrix obtained by removing k-th row and f-th column of M. Maximizing $|a^T x|$ subject to linear constraints is still a non-convex problem, but can be solved by maximizing both $a^T x$ and $-a^T x$, followed by choosing the solution that gives larger absolute objective. We obtain an alternating optimization(AO) algorithm by cyclically updating the columns of M.

Each linear program involves only F variables; so, the worst-case complexity comes down to $O(F^{3.5})$ flops.

F. Algorithm

```
input : D, F;
P ⟵ Co-Occurrence(D);
P = $BB^T, M$ ⟵ I;
while !convergence do
    for f = 1, ..., F do
        $a_k = (-1)^{f+k} detM_{k,f}$ , $\forall k = 1, ..., F$
        // remove k-th row and f-th column of M to obtain $M_{k,f}$;
        $m_{max} = argmax_x a^T x, stBx \geq 0, 1^T Bx = 1$;
        $m_{min} = argmin_x a^T x, stBx \geq 0, 1^T Bx = 1$;
        M(:, f) = $argmax_{mmax,mmin}(|a^T m_{max}|, |a^T m_{min}|)$;
    end
end
$C_* = BM$;
$E_* = (C_*^T C_*)^{-1} C_*^T P C_* (C_*^T C_*)^{-1}$;
output : $C_*, E_*$;
```

Algorithm 1: Anchor Free

III. Implementation

We have implemented the model in python and used scikit and cvx optimization libraries. The model was run on a data set of news articles from BBC. The dataset consisted on 1440 news articles from 4 different topics - Politics, Technology, Entertainment and Economy. Each news article was about 500 words long. The vocabulary size was 19316 which is the set of unique words.

A. Preparing the Data

The first step to preparing the dataset involves cleaning: Cleaning is an important step before any text mining task, which includes the tasks of removing the punctuations, stop words and normalizing the corpus. Tokenization involves splitting the text into sentences and the sentences into words. The words are lowercased and the punctuation is removed. Words that have fewer than 3 characters are removed. All stop words are removed. Words are lemmatized — words in third person are changed to first person and verbs in past and future tenses are changed into present. Words are stemmed — words are reduced to their root form. All the text documents combined is known as the corpus. To run any mathematical model on text corpus, it is a good practice to convert it into a matrix representation. For example, the LDA model looks for repeating term patterns in the entire DT matrix. Each word is assigned a specific token id to represent the matrix as a sparse matrix. We use the scikit learn library methods to represent the documents in a vectored notation. This gives us the D matrix whose each column represents a document. Each element represents the term-frequency of the word in the document.

Datasets used: News article datasets, originating from BBC News, provided for use as benchmarks [8].

B. Running the Model

The next step is to create an object for the model and train it on Document-Term matrix. The number of topics F is input to the algorithm. From the Document Term matrix D, we obtain a new matrix co-occurrence matrix P^{VXV} which is the expectation of each document vector multiplied with it's transpose. P matrix shows the probability of two words occurring together in a document. We perform eigen decomposition on the P matrix to get matrix B. Where BB^T gives P. Now we run Anchor Free algorithm for 10 iterations and observe the C matrix which gives us the word topic frequency values and print the top 15 words in each category. The algorithm converges in about 10 iterations and comes up with really good word topic distributions.

IV. Results

Anchor Free topic models come up with really good topic distributions in a short span of time. Here we have compared the performance with LDA which is a popular topic modelling algorithm. The Anchor Free topic modelling took 24.23 seconds to run while the Gibbs Sampling Algorithm took 134.32 seconds to run for 1000 iterations. The Anchor Free has also come up with better word topic distributions than the Gibbs Sampling algorithm.

Table 1

film award best star music game year nomination actor oscar won include festival new actress band play people number director album tv uk prize	brown tax govern minister prime howard people plan chancellor lord campaign told new public leader mp claim conserve	game mobile people phone music technology service year new digit user firm like player compute make software microsoft uk broadband device video	year economic growth rate govern bank market economy price companies rise sale firm tax new people oil spend 2004 month uk dollar figure
Entertainment	Politics	Technology	Economy

The top 15 words in each category is shown in Table 1. We can see that the algorithms has identified the topic word distributions in a really good manner. The first row has identified the top words associated with Entertainment. The second row shows the top words associated with Politics. The third and fourth rows shows Technology and Economy respectively.

Table 2

film number show including top music won UK years British director TV made BBC star US films New band	government Labour Blair public people plans election minister party BBC spokesman Howard Brown made general make prime Lord	people make technology mobile music games users digital net phone million service game software video security information work computer	US year company sales market growth chief economic deal economy expected firm business rise oil figures cut money European
Entertainment	Politics	Technology	Economy

Table 2 shows the results from LDA using Gibbs sampling algorithm and we can see that the results obtained by Anchor Free Topic modeling is better. Gibbs sampling requires a large number of iterations to converge (more than 1000) while Anchor Free topic modelling takes only 10 iterations to come up with good results. We can also see that Gibbs Sampling took 134.32 seconds to run compared to Anchor Free Topic modelling. Different evaluation methods can be considered to benchmark the performance of topic modelling algorithms. We have used visual inspection and time as the factors for comparing the models.

V. Conclusion

In this paper we have implemented the Anchor Free topic modelling algorithm and compared it's performance with LDA using Gibbs sampling which is a famous topic modelling algorithm. LDA uses statistical methods to identify the topic distribution in different documents. Anchor Free topic modelling does not have the Anchor word assumption and comes up with much better results as compared to the LDA model. The Anchor word assumption may not be possible in every scenario and this algorithm proves to be a very efficient method in such scenarios. The Anchor Free topic model also converges in a very less number of iterations. Experiments on the text corpus proved Anchor Free topic modelling to be a very effective topic modelling strategy. The paper presents the results by Anchor Free topic modeling and LDA and we have come to the conclusion that Anchor Free topic modelling is very efficient and effective method for Topic modelling.

References

[1] D. M. Blei, A. Y. Ng, and M. I. Jordan, *Latent Dirichlet allocation*, Journal of Machine Learning Research, vol. 3, pp. 993–1022, 2003.

[2] Thomas Hofmann, *Probabilistic Latent Semantic Indexing*, Proceedings of the Twenty-Second Annual International SIGIR Conference on Research and Development in Information Retrieval (SIGIR-99), 1999.

[3] Thomas K Landauer, Peter W. Foltz and Darrell Laham, *An introduction to latent semantic analysis, Discourse Processes*, pp. 259-284, DOI: 10.1080/01638539809545028, 1998.

[4] David M. Blei, John D. Laerty, *Text Mining: Classification, Clustering, and Applications*, Chapter 4, pp. 24-48, Chapman and Hall/CRC, DOI: https://doi.org/10.1201/9781420059458, 2009.

[5] Philip Resnik, Eric Hardisty, *Gibbs Sampling for the uninitiated*, University of Maryland, College Park, MD, June 2010.

[6] Xiao Fu, Kejun Huang, and Nicholas D. Sidiropoulos, *On Identifiability of Nonnegative Matrix Factorization*, IEEE Transactions, something, 2016.

[7] Kejun Huang, Xiao Fu, Nicholas D. Sidiropoulos, *Anchor-Free Correlated Topic Modeling: Identifiability and Algorithm*, IEEE Transactions, something, 2018.

[8] News article datasets, BBC News, *http://mlg.ucd.ie/datasets/bbc.html*.