Recommender Systems

Definition

Types: 1) Collaborative, Content-Based, Hybrid

NOTES:

Write ABOUT LDA itself -- > Probabilistic Topic Models of CTR paper

MN - my new addition

OPP+MN – other papers paraphrase + my new addition

PP – paper paraphrase

PO- paper original

**BOPE**

**.**

**.**

**.**

**.**

**Variational Inference**

Variational Bayesian Methods are a group of widely used techniques in a field of statistical Machine Learning. The reason behind their popularity is that VB methods let us reformulate statistical inference problems as optimisation problems. Here, statistical inference problems are the key algorithmic problems where we infer the value of a random variable given the value of another random variable, wheares optimization problems imply to find the values of parameters which minimize the given objective function [\*\*16\*\*].

The most common VB method is the Mean-Field Approximation and in the following section we will explain its optimization objective which is known as **Variational Lower Bound or Evidence Lower Bound (ELBO).**

Suppose the following probabilistic model which is a joint distribution of the observed variables ***X*** and the hidden variables ***Z***:

Inference of the hidden variable *Z* is obtained through its posterior distribution which is written as follows using the *Bayes’ Theorem*:

**KL divergence method**

For many interesting models, the posterior is intractable which means that the exact inference is not possible. Therefore, by considering the Mean-Field Approximation, we adopt the approximate posterior inference technique for these models.

The key idea behind it is to find an approximation distribution that are as **closed** as possible to the true posterior distribution . Note that are variational parameters of approximation distribution and we fit the variational parameters in order to make these both distributions closest [\*\*ShortShot\*\*].

First, we define this closeness by using Kullback–Leibler divergence (i.e. KL divergence) as the distance metric between these approximate and true distributions [\*\*15\*\*]:

where *L* above is called **the variational lower bound**. We reformulate the equation above as follow:

Because the *KL* divergence is always positive (i.e., we get . This proves that *L* is the lower bound of the log probability of observations.

So, our main goal is to **maximize this lower bound *L* i.e. minimizing *KL* distance with respect to variational parameters** Because *L* reaches to *iff* the approximation distribution is perfectly closed to the true posterior distribution [\*\*ShortShot\*\*]. Note that in the formula above is fixed against all variational parameter .

Diagram

Description automatically generated

[\*\*Blei-PPT\*\*]

**Jensen’s inequality method**

Apart from the derivation mentioned above, we also have an alternative way to arrive at the similar conclusions using the Jensen’s inequality which claims for the *concave log function*:

So, the last term in equation above is the **variational lower bound**. Note that in the equation belongs to the Shannon entropy:

where . Essentially, we again show that *L* is the lower bound of the log probability of the observations, and our goal is to maximize this lower bound if we want to maximize the marginal probability.

**Recommender Systems Introduction**

Nowadays, Recommender Systems are widely recognized as one of the most beneficial applications of Machine Learning, and there is no doubt that they drive nearly every aspect of our lives. The fundamental objective of these Machine Learning-driven Recommenders is to filter, prioritize and efficiently deliver the necessary information to the consumers in the midst of overwhelmingly numerous choices on the internet. It is also described as:

*“Any system that produces individualized recommendations as output or has the effect of guiding the user in a personalized way to interesting or useful objects in a large space of possible options.”* (Burke, 2002)

Therefore, many companies utilize the recommender systems for the purpose of helping the consumers discover new and relevant items such as movies, musics, jobs, etc. They use the consumer data which may be in explicit or implicit form (e.g. likes, clicks), in order to comprehensively assess consumers’ preferences and then recommend the relevant items to them. Due to the various criterias, there are multiple techniques of recommender systems, each of which differs in how a single recommendation is generated. The most common types of recommender systems are described in the following section.

**Types of recommender systems**

Although there exists a number of different recommender systems in the literature, we will focus on the three most common ones below:

[[[REMOVE FILTERING]]]]

Diagram

Description automatically generated

* **Collaborative Filtering recommender systems**

CF recommender systems are one of the most widely used systems next to the content-based recommender systems. Essentially, these systems create a user profile based on the ratings of various items and then aims to compare these against a wider user group[12]. As the word “collaborative” from the name implies, multiple users come together as group – taste of one user will be similar to the other users of group. Therefore, by utilizing the users data which contains their historical preferences on a set of items, system deploys an assumption that the users who have previously agreed are more likely to agree again in the future. So, the system creates the new recommendations by taking the similarities between users based on the ratings into consideration.

Diagram

Description automatically generated

Although, CF recommender systems have been used in the industry for many years [13], they still have a limitation such that they can not address the cold start problem –they are not able to recommend items which are not rated by any users (e.g, new items). As a result, it is possible that only famous items may get recommended. Furthermore, traditional CF systems are also memory-wise and computationally expensive and suffers from scalability problems.

* **Content Based recommender systems**

While CF recommender systems, as discussed above, recommend the products or items according to the similarities of user preferences which means that recommendation relies on the user-item interactions, Content Based recommender systems, on the other hand, aims to recommend products or items similar to those a given user has rated positive or liked in the past. So, CB systems generate recommendations based on the comparison between the content of the items and the user profile which was created according to the historical user data. Note that the content of items is described by terms, tags, features or even plots in case if the items are movies.

Diagram

Description automatically generated

An algorithm used to recommend the movies on the Netflix platform is the prominent example which resembles this recommender systems. If a certain user watches and comedy movie and rates it positive via votes or comments, then the new movie recommendations with the same label of that liked movie will be suggested to the user. In other words, based on the content of the consumed item, these recommender systems finds other similar items and recommends them. Note that the such website platforms often keeps the techniques of how the content is actually labeled and matched against each other as secret [12]. Contrary to the CF systems, CB system doesn’t suffer cold-start problem and they can suggest not only famous or older items, but also the unpopular or new items. In addition to this, they are memory-wise and computationally cheap because there is no need for the data of other users in order to be able to compute the recommendation for a specific user.

* **Hybrid recommender systems**

Hybrid recommender systems combine two or more types of traditional recommender systems in order to have better performance by benefiting from complementary advantages of subsystems. Hybrid systems which combine collaborative filtering and content based approaches, achieves state-of-the-art results in many cases and are used in many large scale recommender systems nowadays. Detailed comparison on advantages and disadvantages of Hybrid Recommenders along with Collaborative Filtering and Content Based Recommenders are shown in table below**[[[Insert table ##]]]:**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Collaborative Filtering** | **Content Based** | **Hybrid** |
| **Number of users** | Recommendation based on many users having similar interest | Recommendation based on single user | Combination of collaborative and content based filtering |
| **Disadvantages** | * Cold start problem * Data sparsity * Scalability * Memory-wise and computationally expensive | * Limited content analysis * Over-specialization | * Increased complexity * Increased expense of implementation |
| **Advantages** | * Serendipitous recommendation * User and item features are not required * Quality may improve over time as more users interact with items * - Minimal domain knowledge required | * User independent * No cold start problem * Interpretable results * Memory-wise and computationally cheap | * Avoids most of the shortcomings of other approaches. |

**Probabilistic Models for recommender systems**

The application of probabilistic modeling to the recommendation problem has a rich history which dates back to decades (Breese et al. 1998, Hofmann 2004, Marlin 2004). Many authors incorporated the probabilistic approaches into models which explained the dataset. Initial approaches were probabilistic graphical models such as Bayesian networks and Dependency networks which eventually left their place with subsequent novel topic models such as Latent Dirichles Allocation (LDA) [3] and Probabilistic Latent Semantic Analysis (pLSA) [22]. The term “latent” is used in their name, because both of them are considered probabilistic topic models and the topics they aim to find from the corpus are treated as latent or hidden variables. Additionally, each item of the corpus is represented in terms of topic proportions or latent themes which are also hidden variables.**[[[INSERT MORE ON THIS ON LDA SECTION]]]**

Because of these capabilities, both models have been extensively used for Content Based recommender systems. When it comes to the field of Collaborative Filtering recommender systems, the matrix factorization technique had gained a decent popularity, especially after combined with probabilistic approach [18], [19], [20], [21].

Lately, there has been a lot of interest in combining probabilistic topic modelling with matrix factorization in the field of hybrid recommender systems. One of the major for this is that when a content of item is represented by topic models, the models benefit from interpretable semantics of the latent space characterized by the topic mixtures and this leads to more an interpretable semantics of the item latent factor[\*\*ctmp\*\*]. Initially, Agarwal and Chen proposed probabilistic topic modelling in matrix factorization with fLDA [23]. where the item latent factor took the role of topic proportion in the LDA representation. Despite of being an accurate and interpretable model which handles both cold-start and warm-start scenarios, fLDA still had a limitation in dealing with distinguish items where we have an identical topic mixture, but content details that topic mixture can not cover are of concern to different groups of people. To elaborate on this limitation more, consider that we have two articles; A and B, and both of the articles are about an application of machine learning to social networks. Because both articles are identical in terms of their contents, they will also possesses same topic proportions. Now let’s consider that these two articles are of interest to different kind of users: Article A provides a prominent machine learning algorithm which is applied to social network applications, wheares article B implements a standard machine learning algorithm, but provides a crucial data analysis on social network data. As a result, users who work in machine learning will prefer article A and will hardly be interested in article B, wheares users who work in social networks will be more interested in article B instead of A. However, as the topic proportions of both articles are same, both of them will be recommended to both groups of users[\*\*CTR\*\*].

To tackle the limitation mentioned above, a novel approach called Collaborative Topic Regression (CTR) [\*\*CTR\*\*] has been proposed by David M. Blei and Chong Wang. The way CTR addresses that limitation is by allowing the item latent factor be an offset from topic proportion. So, by this way, an offset may help explain, for instance, an article A is more important to researchers interested in machine learning than it is to those interested in social network analysis. Therefore, CTR allows the item latent factor to also account for user ratings.

Fundamentally, CTR incorporates techniques of both collaborative filtering based on latent factor models and content analysis based on probabilistic topic modelling. According to CTR model, items are generated by a topic model while users are represented with topic interests [\*\*CTR\*\*]. Therefore, CTR is considered as one of the excellent hybrid models which shows that the combination of the content modelling with the matrix factorization methods produces more promising results compared to traditional recommender systems. The graphical model of CTR along with its algorithm is shown below.

**CTR Graphical Model**

Diagram

Description automatically generated

**CTR Algorithm**

1. For each user *u*,draw user latent vector
2. For each item *j*,
   1. Draw topic proportions
   2. Draw item latent offset and set the item latent vector as .
   3. For the *n*-th word of item *j*,
      1. Draw topic index
      2. Draw word
3. For each user-item pair (*u, j*), draw the rating

where is the confidence parameter for . For instance, we trust more if is large.

Despite its advantages, CTR model has significant computational limitations as well. The reason is that the model considers user ratings to have a Gaussian distribution which leads to iterating over all of the entries in rating matrix during training. Because of this, CTR is highly inefficient considering that real-world datasets are very big and sparse.

In order to address CTR’s inefficiency mentioned above, a newer model called Collaborative Topic Poisson Factorization (CTPF)[\*\*CTPF\*\*].

CTPF is only concerned with non-zero ratings during training, and therefore it is much more efficient and scalable. Nonetheless, for the purpose of making sure that the model is conditionally conjugate and has closed-form updates, CTPF tries to model the content generation by standard mixtures of Gamma.

CTMP algorithm which is used in this paper covers the limitation of CTR by considering ratings in Poisson distribution as CTPF does, while modelling contents with LDA [\*\*ctmp\*\*]. CTMP also makes the following contributions to the previous approaches:

1. CMTP has been tested in variety of fields where real-world recommendation is the most challenging one. It has been seen that CTMP outperforms the other existing models significantly. Its main competency is in recommending scientific articles and commercial product recommendation. Recommending movies are also amongst these, and indeed, this is what we will test on this paper.
2. Coordinate ascent algorithm is carried out because CTMP is non-conjugate model. The algorithm is fast and also scalable.
3. [[[[[SPARSITY???? Needs proof]]]

Thus, it is required for any recommender systems to scale efficiently with the big and highly sparse data in real-world. Moreover, the interpretability of the recommendation is also not to be ignored, in which valuable insight can be gained from the interpretation to address downstream analyses.

**MN**

In this paper, we investigate the application of Bernoulli randomness for Online Maximum a Posteriori Estimation (BOPE) algorithm for solving MAP problem in Collaborative Topic Model for Poisson distributed ratings (CTMP) which is a hybrid model for recommender systems. For the purpose of evaluating the performance behaviour, we use datasets such as MovieLense(20M) and Netflix.

~~We also show that performance behaviour of CTMP is better with BOPE compared to OPE which was used in original paper where CTMP was proposed.~~

**Abstract-PP**

CTMP is a hybrid and interpretable probabilistic content-based collaborative filtering model. It allows both content representation by admixture topic modelling and computational efficiency via Poisson factorization coexisting together within a tightly linked probabilistic model, and therefore overcoming a limitation of existing models. CTMP's predictive ability is particularly good in different real-world recommendation contexts, and it is able to scale to very large datasets along with maintaining interpretable user profiles. Another crucial advantage of CTMP is that its sparse representation enables efficient storage of item contents which makes it efficient and desirable model in terms of industrial settings. Therefore, CTMP has been used successfully in industry due to its mentioned benefits.

**Introduction-PP**

CTMP, being a probabilistic hybrid model with scalability and interpretability, offers the following contributions:

1)

2)

3)

**A Collaborative Topic Model for Poisson distributed ratings**

In this section, we describe learning, prediction phases and key properties of CTMP – hybrid and interpretable probabilistic content-based collaborative filtering model.

ADD BELOW to DATASET DESCRIPTION PART

Graphical user interface, text, application, email

Description automatically generated

CTMP.1 - Formalization

Before going into technical parts, let’s provide some notations:

* *U:* represents the number of users inside the dataset
* *J:* representsthe number of items inside the dasaset
* describes the bag-of-word representation for each item *j* where expresses the frequency of term/word in item *j.*
* represents the vocabulary size of the corpus.
* describes the dataset where is a rating provided by user *u* to item *j,* while is the bag-of-word representation of item *j* as already explained above. represents the ratings given to movies by users. Every rating is expressed as binary 0 or 1. If user *u* liked an item *j,* then . On the contrary, if the user *u* do not know about the item *j* or do not like it, then .
* *K:* represents the number of topics inside corpus.
* describes the topic representation. More precisely, every topic *k* is a distribution over the vocabulary. It is described as and . Note that, lies in the (*k* – 1)-simplex.
* describes the topic proportion of the items. is the vector of the distribution on topics for item *j,* and . Note that, lies in the (*k* – 1)-simplex.

In order to learn the topics , we use the Latent Dirichlet allocation (LDA) and its Expectation-Maximization (EM) approach which was described in the respective section of LDA. Furthermore, by learning the topic proportion of each item we describe each item and user in the *K*-dimensional space. Note that these learning procedures will be explained in the further sections below.

Now, we present **latent factors** for each user and item in terms of *K*-dimensional vectors and , respectively. As discussed in [[[Related Work Section]]], the reason why we consider rather than as the latent factor for item is that in order to have better recommendation system, we allowed an offset between and which accounts for the user-specific preference on the item content that alone can not capture. Therefore we denote that where is an offset term which has Gaussian distribution. Note that in the formula above represents an *K*-dimensional identity matrix , and is a regularization parameter. So, we have .

Furthermore, as shown below, the ratings and users’ latent factors are modeled by Poisson and Gamma distributions, respectively. To put everything together, the generative process of CTMP is as follow:

Text, letter

Description automatically generated

1. For each user *u,* draw where
2. For each item *j*:
3. Draw topic proportion
4. For the *n*th word of item *j*:
   1. Draw topic index
   2. Draw word
5. Draw latent factor
6. For each user-item pair (*u, j*), draw

Note that step 2(a-b) correspond to LDA.

Below is graphical representation of CTR, CTMP and CTPF:

Diagram, engineering drawing

Description automatically generated

**Learning CTMP**

Full posterior of latent variables is given as follow:

(1)

The problem with this posterior is that it is intractable, and therefore exact inference is impossible. In order to tackle this problem, we have two methods:

1. Maximum A Posteriori (MAP) for point estimation
2. Bayesian Learning such as MCMC Sampling or Variational Methods for approximate inference

As the prior and posterior distributions of hidden variables and are not conjugate in CTMP model, using Variational Methods of Bayesian Learning in order to infer these hidden variables does not get us closed-form solution. Therefore, we will carry out the point estimates of and using MAP – coordinate ascent algorithm developed by authors of original paper of CTMP [\*\*CTMP\*\*].

Furthermore, in order to facilitate the learning, authors added **a** **new auxiliary variable *y****,* where Poisson( and . Note that we approximate the posterior of and via mean-field variational inference [\*\*5\*\*]. The mean-field variational inference is a type of Variational Bayesian Method which allows to re-write a statistical inference problems as an optimization problem[[\*\*\*insertsomething\*\*]]. Therefore, we can convert the inference problem of CTMP into a full optimization problem where the single objective function which needs to be maximized is as follow:

(2)

As shown in (2), the term of integration and summation over the whole space causes optimization to be intractable. However, Variational method [\*\*5\*\*] also tackles this problem which will be discussed in detail below.

Note that has Poission distribution, and the *K*-dimensional vector follows multinomial distribution: [\*\*6\*\*] So, we get the variational distribution as follow:

(3)

where such that is variational parameter of , and are variational parameters of . Note that . Now we get the **evidence lower bound (*l*)** by applying Jensen’s inequality[[\*\*insertpaperofJensen\*\*]]:

(4)

Note that before learning the hidden parameters, are considered as fixed parameters in the model.

Next, the lower bound *l*( shp, rte, ) is maximized with respect to shp, rte, . According to Appendix A, we express the terms in detail as follow:

(5)

**Learning Parameters**

Equation (5) is the optimization problem and as mentioned before we solve it by coordinate ascent algorithm. CTMP algorithm for learning and is demonstrated below in Algorithm 1

Algorithm 1.

**Input:** Observed data *w, r,*Bernoulli parameter

**Output:** Estimates

**1. init** Initialize

[[[CHANGE THE CIRCLE OF USERS AND DOCS AND MENTIONE WHY WE CHANGED THE SEQUENCE – BECAUSE OF INITIALIZATIONS]]]

**Learning** . In order to find the point estimate of local topic proportion where

(6)

we use Bernoulli randomness for Online Maximum a Posteriori Estimation (BOPE) algorithm [\*\*7\*\*]. Note that in original paper of CTMP [\*\*CTMP\*\*], authors used a simple Online Maximum a Posteriori Estimation (OPE) algorithm and this difference is the most important one between this and original CTMP paper. By using Bernoulli randomness, BOPE has a faster convergence rate, is more general and flexible compared to OPE. Furthermore, “BOPE implicitely employs a prior which plays a regularization role”[\*\*7\*\*]. Comparison of BOPE with respect to OPE has been carried out in BOPE section in detail. Include this in comparison::: such properties are not found in the common approximate posterior inference methods for topic models, such as Gibbs sampling and variational Bayes. [[[or discuss here???]]] Note that both algorithms tries to lead the solution of the optimization to the closed neighbours of the vertices in the convex hull of compact input domain and they have a fast convergence rate of along with proven quality bound [\*\*8\*\*]. Furthermore, as mentioned earlier too, every topic proportion holds and lies in the (*k* – 1)-simplex. BOPE algorithm for learning is described in Algorithm 2.

Algorithm 2

**Input:** Bernoulli parameter

**Output:** which maximizes over the compact domain }

**1. init** Initialize

**5.**

**Learning** If we know the estimates of other hidden variables, then solving analytically is possible because the objective function regarding the is a *concave* function.

(7)

The partial derivative of function with respect to , i.e. for all *k,* is the estimate of . This is also so called the stationary point of . Because is the quadratic function in terms of we can Vieta’s formula for the analytical derivation of the function’s root as follow:

(8)

where

(9)

**Learning** . We use the mean-field variational inference for approximating the conditional posterior of and as in [\*\*9\*\*]. So, in order to solve for the variational parameters of and which are , we solve for the stationary point of with respect to each variational parameter, while holding the others same. The update expression of variational parameters is given in Table/Figure?? [\*\*x\*\*] below. The detailed derivation of these expressions is described in Appedix B and Appendix C. One of the biggest advantages of CTMP algorithm is that whenever , we get and and therefore, we only have to update over non-zero ratings (. This property of our model diminishes the training time significantly, so the total training time is much lower than of other models such as CTR, especially whenever the rating dataset is highly sparse. Because, during each epoch of training, we only consider the positive ratings for updating the expression of and skip all *zero* ratings.

Variational parameter updates

Note that the function in the update expression of denotes the digamma function [\*\*10\*\*] where

(10)

**Learning** . So far, we have provided the update expression of the variables regarding both documents and users such as . Now, we have to do the remaining task which is to solve for First, we express the log likelihood of the whole items corpus *C* as in [\*\*10\*\*]:

(11)

By using Jensen’s inequality, the last term is derived, due to the fact that Next, the lower bound of is maximized with respect to as in [\*\*10\*\*]:

(12)

where Note that the each is separable from each other inside the objective function of . So, we can solve solve each individually. This is carried out by considering the Lagrange function and setting its derivatives to 0 which results in formula of as follow:

(13)

**Prediction**

We rank the items in order to generate recommendations for each user *u* based on their predictive score after we have learned all the parameters. Because the ratings in the dataset are discrete Poisson variables, can be the expectation of the rate parameter given the observed data i.e. as in CTPF[\*\*CTPF\*\*]. However, the derivation in CTMP is a bit different because CTMP neither aims to approximate solely by point estimate nor require **conjugacy** between the complete **conditional distributions** for the inference as CTPF does [\*\*CTMP\*\*] :

Note that only is the MAP estimation of complete conditional distribution. Furthermore, is nearly the expectation over the respective variational distributions of ’s:

Note that both and are the estimation of variational parameters that we learned in above section.

**CTMP key properties**

**.**

**.**

**.**

**Interpretable user profiles**

**.**

**.**

**.**

**Evaluation**

**.**

**.**

Two most common tasks in recommender systems are predicting the score the user might give for a product (the rating prediction task), and recommending a ranked list of most relevant items (the top-N recommendation task)

[[Take definitions of “in-matrix” and “out-of-matrix’ from CTR paper]]

**.**

**.**

**LDA**

*INCLUDE SOMEWHERE: Note that LDA possesses so called hidden generative process and according to this process, the model is assumed to generate the observed data (i.e. items, users or ratings). Obviously, this was just a generative assumption in order to facilitate the algorithm and it does not illustrate the true process of the real data [17].*

*INCLUDE SOMEWHERE:* *Topic modeling algorithms [5] are used to discover a set of “topics” from a large collection of documents, where a topic is a distribution over terms that is biased around those associated under a single theme. Topic models provide an interpretable low-dimensional representation of the documents [8]. They have been used for tasks like corpus exploration, document classification, and information retrieval.[[TAKEN FROM CTR]]*

*INCLUDE SOMEWHERE: To generalize to unseen articles, our algorithm uses topic modeling. When topic modelling used in our recommender system, this component can recommend articles that have similar content to other articles that a user likes. The topic representation of articles allows the algorithm to make meaningful recommendations about articles before anyone has rated them.*

In machine learning, topic modeling is a statistical model for discovering the abstract topics that occur in a collection of documents. [1] It is also considered as a probabilistic model for an unsupervised classification of the documents.

There are many topic modeling algorithms, among which Latent Dirichlet allocation (LDA) is the most popular one. LDA is a three-level hierarchical Bayesian model, and its basic idea is that documents are represented as random mixtures over underlying set of topics, where each topic is characterized by a distribution over words. [\*\*original-lda-paper\*\*]. By this way, topic probabilities express an explicit representation of each document. This can also be be explained as:

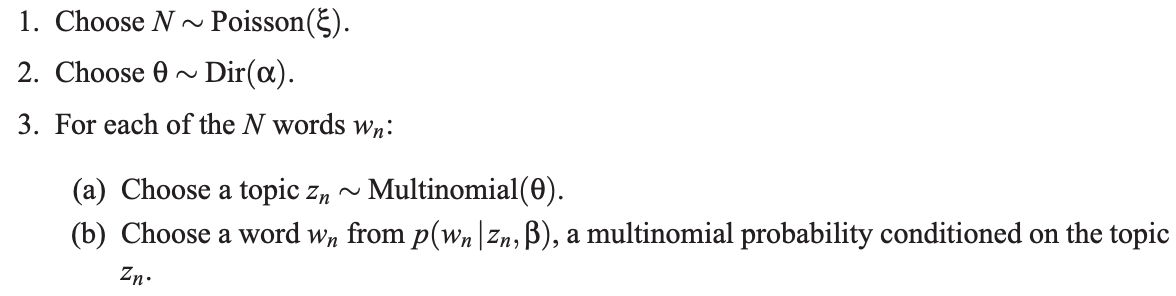
* **Each document is a mixture of topics**. By this way, we consider that each document contains terms/words from some topics in specific proportions. For instance, if we consider that there are 2 topics in the whole corpus, then we might say some document could be 75% topic A, and 25% topic B, while another document might be consisted of 30% topic A, and 70% topic B”.
* **Each topic is a mixture of words**. In this case, we consider that each topic is expressed by the words that explain it most. For example, if we consider that there are 2 topics, namely, “sports” and “education”, in the whole corpus, then the most used words for the sports topic could be “teammate”, “win”, and “play”, while the education topic could contain the words such as “lecture“, “book” and “class”. It is necessary to note that the same words can appear on the multiple topics. For example, the word “time” could participate in both sports and education topics.

By this way, documents overlap with each other with relationship to their contents, rather than being seperated into different individual groups.

To put it into terminology:

* A *word* is a term of the vocabulary and it is indexed by
* A *document* is a series of words given by where is the *n*th word inside the document.
* A *corpus* is a collection of a total *M* documents and it is given by

Generative process of LDA for each document in the whole corpus is described below:

replace this with LDA algo from CTR paper. Also take how EM is used

In the figure above, is the topic proportion of some document, is the topic representation of some topic and is the Dirichlet prior parameter. The Poisson assumption of the proccess is not essential to all that follows and more realistic document length distributions can be used as appropriate. So, *N* is completely independent of all data generating variables such as z. [\*\*original-lda\*\*]

Topic proportion is a *k*-dimensional Dirichlet random variable and its domain is in the (*k* − 1)-simplex. In other words, *k*-vector is in the (*k* − 1)-simplex if . The Dirichlet is a exponential family distribution on the simplex. Some of its important properties are that it is conjugate to the multinomial distribution. [\*\*4\*\*]. Note that, all of these properties help the development of the inference and parameter estimation for LDA. Additionally, the probability density of ’s simplex is as following:

,

where is a *k*-vector with components and is the Gamma function.

Probabilistic graphical model of LDA is represented below [\*\*original-lda\*\*]:

Diagram, schematic

Description automatically generated

The outer box illustrates the documents, and the inner box illustrates repeated choice of topics and words inside documents [\*\*original-lda\*\*]. According to the figure above it is once again shown that LDA is a three-level hierarchical Bayesian model. The parameters that are considered as corpus-level areand , and they are supposed to be sampled during the process of generating the corpus. [\*\*original-lda\*\*]. The variables given as are considered as document-level, and they are sampled once per document. Lastly, the variable given as and are considered as word-level, and they are sampled once for each word inside every document. During the LDA process, topics are sampled repeatedly within each document.

So, the joint distribution of topic proportions topics set of topics z, and set of words w is as following:

*~~[[[[[because above posterior distribution is intractable, we can use variational inference]]]]~~*

*~~[[[[[ geometric interpretation of LDA]]]]~~*

*~~[[[[ Learning theta, beta by EM algorithm because EM was referenced in Formalization part of CTMP]]]~~*

*~~Geometric interpretation of LDA~~*

*~~Illustrating the geometry of the latent space is a another good way for grasping the concept of LDA. As shown below in the figure,~~*

**Inference and Parameter Estimation**

Computing the posterior distribution over the hidden variables given some document is the the main problem we have to overcome in order to be able to use LDA:

Unfortunately, the posterior distribution derived above is intractable for an exact inference. However, there are some methods of an approximate inference such as Variational Bayesian Methods which is used in a complex statistical models. The main idea behind the variational inference is to get an adjustable lower bound on the log likelihood by using Jensen’s inequality (Jordan et al., 1999). Primarily, we consider a collection of lower bounds which are indexed by a set of variational parameters and these variational parameters are chosen by our optimization procedure that tries to find the tightest possible lower bound[\*\*lda\*\*].

In order to be able to deploy variational inference method for getting the tractable family of lower bounds, we make some adjustments to our original graphical model of LDA:

**Diagram

Description automatically generated**

Where respectively. Now, our aim is to optimize the problem that determines the values of the variational parameters and . This is exactly what finding a tight lower bound on the log likelihood is. Essentially, by minimizing the Kullback-Leibler (KL) divergence between the true posterior distribution and the variational distribution, we optimize the values of variational parameters. It is worthwhile to note that this minimization can be obtained via iterative fixed-point method as in [\*\*lda\*\*]??.

TAKEN FROM LDA:: It is important to note that the variational distribution is actually a conditional distribution, varying as a function of w. This occurs because the optimization problem in Eq. (5) is conducted for fixed w, and thus yields optimizing parameters (γ∗,φ∗) that are a function of w. We can write the resulting variational distribution as q(θ, z| γ∗(w),φ∗(w)), where we have made the dependence on w explicit. Thus the variational distribution can be viewed as an approximation to the posterior distribution p(θ, z|w,α,β)

First, thank you for this fresh opportunity that many students are looking for and interested in nowadays.

APPENDIX A

.

.

.

.

.

.

[1] <https://en.wikipedia.org/wiki/Topic_model>

2?? <https://www.tidytextmining.com/topicmodeling.html>

[3] <https://dl.acm.org/doi/pdf/10.5555/944919.944937>

[4] <https://stephentu.github.io/writeups/dirichlet-conjugate-prior.pdf>

[5] <https://people.eecs.berkeley.edu/~jordan/papers/variational-intro.pdf>

[6] Univariate Discrete Distributions, vol. 444 [[[[take from CTMP]]]]

[7] <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9138369>

[8] <https://arxiv.org/pdf/1512.03308.pdf>

[9] <https://www.scopus.com/record/display.uri?eid=2-s2.0-84982318199&origin=inward&txGid=117c9f14425c2abc105f8cd8ac63fa5f>

[10] <https://www.sciencedirect.com/topics/mathematics/digamma-function>

[11] Fully Sparse Topic Model (FSTM)

[12] <http://www.diva-portal.org/smash/get/diva2:1219240/FULLTEXT01.pdf>

[13] [https://datajobs.com/data-science-repo/Recommender-Systems-[Netflix].pdf](https://datajobs.com/data-science-repo/Recommender-Systems-%5bNetflix%5d.pdf)

[14] <https://en.wikipedia.org/wiki/Collaborative_filtering>

[15] <https://link.springer.com/referenceworkentry/10.1007%2F978-3-642-04898-2_327>

[16] https://blog.evjang.com/2016/08/variational-bayes.html

[17] <https://dl.acm.org/doi/pdf/10.5555/1378245.1378272>

[18] <https://www.cs.toronto.edu/~amnih/papers/bpmf.pdf>

[19] <https://pubmed.ncbi.nlm.nih.gov/24467759/>

[20] <https://papers.nips.cc/paper/2007/file/d7322ed717dedf1eb4e6e52a37ea7bcd-Paper.pdf>

[21] <https://www-users.cs.umn.edu/~baner029/papers/10/gpmf.pdf>

[22] <https://arxiv.org/pdf/1301.6705.pdf>

[23]https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.564.5299&rep=rep1&type=pdf

https://people.eecs.berkeley.edu/~jordan/papers/variational-intro.pdf -

In particular, they make a link between this lower bound and parameter estimation via the EM algorithm