Information Systems 94 (2020) 101582

Contents lists available at ScienceDirect

Information Systems

journal homepage: www.elsevier.com/locate/is

A review of topic modeling methods

Ike Vayansky [a], Sathish A.P. Kumar [b,*]

[a] Department of Computing Sciences, Coastal Carolina University, Conway, SC 29528, USA
[b] Department of Electrical Engg and Computer Science, Cleveland State University, Cleveland, OH, 44115, USA

ARTICLE INFO

Article history:
Received 17 March 2020
Received in revised form 13 June 2020
Accepted 15 June 2020
Available online 18 June 2020
Recommended by Dennis Shasha

Keywords:
Topic modeling
Probabilistic Bayesian networks
Text analysis
Topic correlation
Temporal analysis
Social Media analysis
Inference algorithms

ABSTRACT

Topic modeling is a popular analytical tool for evaluating data. Numerous methods of topic modeling have been developed which consider many kinds of relationships and restrictions within datasets; however, these methods are not frequently employed. Instead many researchers gravitate to Latent Dirichlet Analysis, which although flexible and adaptive, is not always suited for modeling more complex data relationships. We present different topic modeling approaches capable of dealing with correlation between topics, the changes of topics over time, as well as the ability to handle short texts such as encountered in social media or sparse text data. We also briefly review the algorithms which are used to optimize and infer parameters in topic modeling, which is essential to producing meaningful results regardless of method. We believe this review will encourage more diversity when performing topic modeling and help determine what topic modeling method best suits the user needs.

Published by Elsevier Ltd.

1. Introduction

One of the most critical goals of data analytics is determining the characteristics that data points share. In text analysis, this often means determining what events or concepts a document is discussing. This information is clear to a human reading a document, but a program is given only the text as it is written not the subject matter of each document. In order to accomplish this task in a program, data scientists utilize a method called topic modeling. Topic modeling is a popular statistical tool for extracting latent variables from large datasets [1]. It is particularly well suited for use with text data; however, it has also been used for analyzing bioinformatics data [2], social data [3], and environmental data [4]. This analysis can help with organization of large-scale datasets for more convenient access; a few examples of these applications is structuring databases of journals and articles into groups based on similar focus [1], social media users by similar post content [3], and genomic data by similar sequence structure [2]. Despite its popularity, topic modeling is prone to serious issues with optimization, noise sensitivity, and instability which can result in data which is unreliable [5]; some techniques are also not representative of real-world data relationships [6]. This is due to assumptions regarding key parameters in the calculation process and inefficiency of many optimization methods, which often attempt to overcome uncertainty by performing many time-consuming iterations to determine the best value for the parameter. There also are few academic reviews discussing the different methods of topic modeling and the approaches which can be used to better optimize topic modeling to a dataset. In this paper, we will critically review some of the most highly cited scientific literature on topic modeling processes, as well as the most recent work on the subject. By reviewing these works, we will analyze what proposals best combat the different limits of topic modeling and what models are best used for different analysis scenarios.

The extraction of meaningful statistics and features from a dataset relies on choosing the proper methods. Although current topic modeling approaches perform significantly better than early algorithms, they still require optimization and tuning to provide reliable results [5]. As previously stated, different topic modeling methods have been designed for use with more specific data relationships and structures, such as short texts [7], long-term sequential data [8], highly correlated data [6], and data with complex structural relationships [9]. For a researcher attempting to begin a data analysis project, it is imperative to understand the differences between models and their underlying algorithms in order to form a topic modeling procedure which best serves their purposes. Yet despite the wealth of research which has been performed on developing topic models and improving algorithms, we are not aware of any publication specifically focusing on the comparison of topic modeling approaches and methods. Thus, our goal in this review is to describe a selection of well-documented topic model structures which can be utilized to both aid those exploring possible methods for future studies as well as those

[*] Corresponding author.
E-mail address: s.kumar13@csuohio.edu (S.A.P. Kumar).

https://doi.org/10.1016/j.is.2020.101582
0306-4379/Published by Elsevier Ltd.

seeking to critique published work utilizing topic modeling. Here we shall describe several considerations for topic modeling, introduce a variety of different topic modeling approaches as well as some recent derivatives of these models, and discuss techniques and algorithms used when optimizing topic modeling for a dataset. Although this work is intended to outline and summarize the basic design of topic modeling methods, it should not be considered a complete guide to performing these algorithms. This also in not an all-inclusive review of topic model approaches, as there are numerous different models and alternative methods in this field of data analysis. We encourage those reading this paper to seek out the original publications cited here for any methods which may interest them in order to completely comprehend the topic modeling process, and to remain open to other approaches which they may discover when reviewing published works. This paper is intended to act as an overview of several frequently cited methods and offer options for different types of data scenarios. Where applicable, a link to the code for the method is provided.

Our review is organized in the following structure: Section 2 briefly discusses some of the techniques which contributed to the development of the topic model approaches used today. Section 3 defines important terms which will be used to describe topic modeling approaches within the paper. Section 4 outlines several different standard topic modeling approaches based on what data features they consider. Section 5 discusses methods of optimizing the topic modeling process through inference and approximation algorithms. Section 6 concludes the paper by outlining the topic modeling approaches we have presented and how they can be utilized to achieve the desired results from a dataset.

2. Background

Topic modeling was originally developed in the 1980's and branched off from the subject area of "generative probabilistic modeling" [2]. This type of modeling assumes that observed variables interact with unobserved, or latent, parameters in a specific probabilistic relationship which then generates the data within a dataset [10]. The development of these processes arose from the need to briefly describe elements within increasing large collections of data without compromising statistical relationships required to complete more straightforward analyses, like classification and summarization [11]. The first method developed for this task was called the *tf–idf* reduction scheme and was proposed by two researchers in the field of information retrieval, Salton and McGill, in 1983 [12]. In this method, each individual document in a corpus is regarded by its vocabulary, and the number of occurrences of each word is tallied to form a count value to form a term frequency count (*tf*) specific to that document for that word. The total instances of a word over the entire corpus, called an inverse document frequency count (*idf*), is also calculated. After normalizing these values to the dataset as needed, the two values are compared forming a term-by-document matrix containing the *tf–idf* values for all the document within the columns [12]. This reduces a corpus into a V-by-D dimensional matrix and documents to fixed-length vectors composed of real, positive numbers. Although effective at identifying sets of words that distinguish documents in a collection, the reduction of description length was relatively insignificant and the approach yielded little meaningful information about statistical relationships within or between documents [11]. *Tf–idf* reduction was succeeded by another dimensionality reduction method developed by Deerwester et al. called latent semantic indexing or analysis (LSI/LSA) in 1990 [13]. In LSI, the *tf–idf* matrix is factorized by singular value decomposition (SVD), resulting in three separate matrices—two unitary matrices of V-by-V and D-by-D dimensions respectively and a V-by-D matrix containing non-negative real numbers on its diagonal, known as the singular values of original matrix. The resulting matrices are used to find a linear subspace within the space of original *tf–idf* matrix which describes the majority of the variation in the corpus [13]. When performed on large corpora, LSI can achieve significant compression of data; however, a more direct method of analysis could be implemented by forming a generative model and fitting it using probabilistic methods [11]. This improvement was proposed by Hofman in 1999 and was called the probabilistic LSI or LSA (pLSI/pLSA) model, sometimes referred to as the "aspect model" [14]. This approach turned away from dimensionality reduction methods and instead focused more on probabilistic modeling. Within a document, each term is considered to be sampled from a mixture model made up of arbitrary multinomial variables which each can be regarded as representing a topic. This means a document is thus characterized as a collection of differing proportions of mixture components, somewhat like a document recipe. From this, the description of a document can be concentrated into a probability distribution over a predetermined array of topics, representing significant improvement in reduction over previous methods [14]. Unfortunately, this approach lacked a probabilistic model for the determining the mixture proportions of a documents; therefore, total model parameters increase linearly in relation to increasing corpus size, causing serious overfitting concerns. This also meant that pLSI was incapable of considering documents outside of the training set [11]. Despite its shortcomings, this marked the first instance of probabilistic methods being adopted for topic identification. These methods led to what are currently considered the most popular topic modeling approaches, which are primarily centered around Bayesian theorems, logistic normal distributions, or matrix factorization.

3. Definitions of terms

In topic modeling, the type of data analyzed can come from a variety of sources and forms, including genetic and biochemical sequences [2], images [15], video [16], and geospatial data [17]. For the sake of simplicity, we will discuss topic modeling in terms relating to its use for text-based data. Within this paper, a "word" or "term" will represent the fundamental unit of individual data, a "document" represents a string composed of N words, and a "corpus" represents a set made up of M documents generally encompassing the entire dataset. A "vocabulary" shall be defined as the collection of all distinct words within a corpus, and a "topic" shall be characterized as a probability distribution spanning a given vocabulary. Algorithmically, words can designated by unit vectors which span the dimensions of a vocabulary indexed by $\{1, \ldots, V\}$; considering this in superscript notation, the vth term of a vocabulary would be denoted as a V-dimensional vector w such that $w^v = 1$ and $w^u = 0$ for $v \neq u$. In more basic terms, a single component representing the word's position in the vocabulary in these vectors is equal to one and all other components are equal to zero. Following this reasoning, a document can be regarded as a unit matrix represented by $\mathbf{w} = (w_1, w_2, \ldots, w_N)$ where w_i represents the ith word in the sequence and N defining the number of words within the document. A corpus is thus represented by $\boldsymbol{D} = (\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_M)$ or $\boldsymbol{D} = (d_1, d_2, \ldots, d_M)$ in which d_n is equivalent to $\mathbf{w}_n$ and signifies the nth document in the corpus and M defines the total number of documents within the corpus. Topics are represented by different probabilistic or stochastic distributions depending on the method being used, and in some cases can also represent distribution over other topics as well. In its simplest form and not considering hierarchies or sequential relationships, this can be represented by $\mathbf{z} = (z_1, z_2, \ldots, z_K)$ with z_j representing the jth topic and K defining the number of topics spanning the corpus [11].

Unless otherwise specified, the parameters ϕ and θ are seen as mixture weights and characterize the significance of words for a given topic and the significance of topics within a specific document, respectively. Thus, a formula such as $\phi^{(z)} = P(w|z)$ refers to the multinomial distribution of terms over a given topic z whereas the formula $\theta^{(d)} = P(z_d)$ refers to the multinomial distribution of topics over a given document d [10].

In this review, we synthesize the information found within literature on topic modeling to the best of our abilities and in a manner which we believe will be most logical to the reader. The research which this review discusses are based on similar statistical theories and generative processes, but often independently define variables and thus representations of parameters may vary between papers [18–23]. For clarity, we have changed symbols from those outlined in the original works where needed in order to maintain consistency throughout this review. When reviewing these publications, please take to time to become thoroughly familiarized with variables as they are defined by the authors in order to avoid confusion.

4. Standard topic model approaches

4.1. Basic approaches

4.1.1. Latent Dirichlet allocation

One of the earliest and more frequently utilized topic modeling method which will be discussed is called "Latent Dirichlet Allocation" (LDA), developed by Blei, Ng, and Jordan in 2003 [11] (https://github.com/blei-lab/lda-c); the provided code is from the author and utilizes variational EM inference, in the coding language C. This approach structures the data into three levels—word, topic, and document. LDA regards documents as generated from randomized mixtures of hidden topics, which are seen as probability distributions over words. In order to generate each document, it first samples a K-vector θ representing the mixture proportion of k topics from a Dirichlet prior distribution $p(\theta|\alpha)$. The variable k will define the dimension of this distribution and thus the dimension of the topic variable $\boldsymbol{z}$ as well, but also represents the number of total topics which will be returned in the model. It is important to note that this value is assumed to both static and known in this approach. Additionally, a matrix β with the dimensions $k \times V$ parameterizes word probabilities such that $\beta_{ij} = p(w^j = 1|z^i = 1)$ where $i = 0, 1, \ldots, K$ and $j = 0, 1, \ldots, V$. When $\theta_i \geq 0$ and $\sum_{i=1}^{k} \theta_i = 1$, a Dirichlet variable θ of k-dimensionality can occupy values in the (k-1)-simplex and has a probability density on this simplex determined by the following equation:

$$p(\theta|\alpha) = \frac{\Gamma\left(\sum_{i=1}^{k} \alpha_i\right)}{\prod_{i=1}^{k} \Gamma(\alpha_i)} \theta_1^{z_1 - 1} \ldots \theta_k^{z_k - 1} \tag{1}$$

The parameter α in Eq. (1) is the hyperparameter of the Dirichlet distribution. It can be seen as a prior count of the times an individual topic is observed in a document, embodied by a k-vector of elements $\alpha_i > 0$. In some situations, particularly those with an unknown number of topics contained in each document, it is beneficial to utilize a symmetrical Dirichlet prior where all α values are equal and can thus be simplified to a singular α over the whole corpus [10]. The combinations of topics are influenced by the value of this parameter, with higher α values causing the distribution of topics to be pushed away from the extremities of the simplex resulting in what is called "smoothing" [11]. It is common procedure to use $\alpha < 1$, which locates the modes of the Dirichlet distribution into the corners of the simplex and creating a bias toward sparsity [10]; this is especially recommended for work using social media content, which is often restricted by character limits [3]. The hyperparameter β is also a symmetrical Dirichlet prior, and can be similarly viewed as the number of occurrences of words sampled from a topic prior to the observation of any word within the corpus [10]; this value influences the degree of smoothing over word distributions in all the topics [11]. The selection of values for α and β depend on the size of the vocabulary and the number of topics selected; Steyvers and Griffith suggest $\alpha = 50/K$ and $\beta = 0.01$ as a broad choice which they have found to work well with a variety of text collections [10]. For further derivations and assumptions involving inference and parameter estimation in LDA see [11].

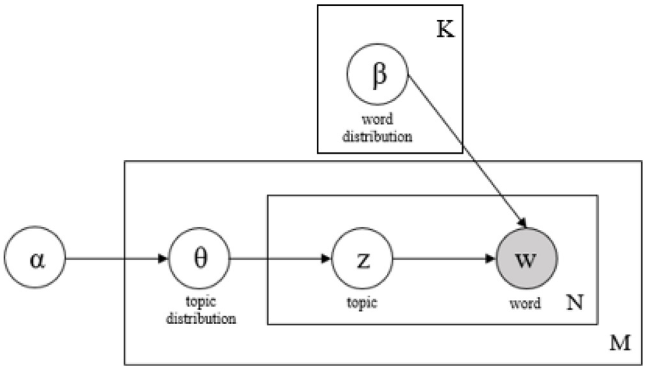

Fig. 1. Illustration of the LDA generative process. The outer box M denotes the repeated sampling for each document and the inner box N represents sampling within each document. The box K represents sampling for each topic.

The basic generative process for this model can be seen in Fig. 1; this format is called "plate notation" and will be used throughout this work as well as in many of the original publications. Boxes in this figure are called 'plates' and indicate replicate actions within the model, with the label of the plate referring to variable corresponding to the number of replications. Circles represent variables or parameters, with white circles indicating the variable is latent or hidden and shaded circles indicating information which is given. Arrows demonstrate the hierarchy of influence of one variable over another. For example, the plate K containing the parameter β indicates that words will be individually and independently sampled from distribution β for each topic of k topics.

The LDA approach demonstrated a significant improvement over previous work for considering the inference of unseen documents and understanding unstructured data; however, it requires several additional steps of parameter estimation for α and β in order to maximize marginal log likelihood of the data. Several processes proposed for this task were variational inference methods [11], Markov chain Monte Carlo (MCMC) [24], and fast collapsed Gibbs sampling [25]. The problem with these approximation methods is that the final results can differ greatly for the same data using the same algorithm depending on the initial conditions, which results in difficulty reproducing models and significant noise issues [10]. Additionally, LDA has demonstrated issues with sparsity when encountering large vocabulary size in corpora [11,26]. Instability and parameter inference are not the only obstacles for this topic modeling approach. LDA makes several assumptions in its calculations that might not be effective for generating accurate or realistic topic models given real world datasets. The first assumption is that the value of k is fixed and known [1]; this variable characterizes the topic structure that the model generates and thus significantly impacts resolution between topics. For most real-world applications using topic modeling, there is not an ideal number of topics established for a corpus prior to generating a model. The means for best determining this value is not well-defined for the majority of topic modeling methods, including LDA. In order to determine this value, researchers often iteratively generate topic models for the same data using different k values and then analyze certain metrics, such as perplexity, to compare the results [27]. For large datasets this is inefficient and imprecise, as the perplexity value of results can fluctuate due to the random nature of the LDA approximation algorithms and internal weight sampling; this process also takes up more time and computational power as the size of the corpus increases, since each individual topic model takes longer to produce [27]. The second assumption made in LDA is that all the topics are independent of each other, and therefore this model does not account for correlation between topics [11]. Such correlations are a common part of many types of data, especially text and social media data. This limits the ability of this algorithm to handle big data accurately and make predictions for new documents, which in turn reduces its application for real-world scenarios. Similarly, LDA also assumes that both documents and words within documents are independent, meaning that their order does not matter [1]; the document independence assumption is inherent to the Dirichlet probability [11] and the independence of words is called a "bag-of-words" assumption and is part of many statistical models [1]. The order of words within a sentence and the relationships between documents over time are both important characteristics of real world textual or linguistic data [8]. No amount of optimization or additional steps can overcome these faults in LDA since the independency assumption is rooted in algorithms of the Dirichlet distribution itself [11]. Thus, new methods had to be developed in order to consider topic and document correlations in these models. Additional concerns with this process are involved with the training of the algorithm. Like other data analysis methods and as previously mentioned with approximation methods, LDA is prone to what is called "order effects", or variation in results in response to the order of the training data used [5]. A general practice in data analytics is to generate training sets from only a portion of the total dataset using a random seed. This means with every iteration the model may create different topics from the same data and assign different words to similar topics between runs. This is a significant obstacle for classification tasks which utilize LDA for generating training data, especially those which do not optimize the model's default settings [5]. This causes major instability in LDA models and is one of the reasons that iterative testing of different k values is ineffective for optimizing topics.

There have been a variety of derivations of LDA proposed for specific tasks and improvements. A few examples of these models are the latent Dirichlet mixture model (LDMM) [28], matrix factorization through LDA (fLDA) [29], and the Hierarchical Latent Dirichlet Allocation (hLDA) [30]. Each of these new methods combine the traditional features of LDA with another probabilistic or statistical process in order to improve performance, integrate inference or optimization, or consider advance topic relationships while maintaining the basic LDA approach. Much of the new research published on topic modeling methods are derivatives of LDA or the combination of it with other models. Traditional LDA and revised LDA methods have been used for a vast number of purposes, making LDA the most cited method in topic modeling research [1]. It is prone to fault and requires significant optimization for quality results; however, it is adaptable in terms of joining with other models and applying to a variety of different data types. It also marked a significant turning point in modeling text and discrete data, and therefore is often referred to as a benchmark for all other models in publications. It has been utilized in studies regarding social media and web patterns, semantics, behavior, linguistics, document assessment, network patterns, predictive models, image and video processing, and many other topics [1]. It also can be performed as both a supervised, unsupervised, and semi-supervised approach with minimal alteration [11]. Despite its shortcomings, LDA could be adjusted to suit most general analysis tasks and has a wealth of previous work to review and consider. However, if this method is selected, optimization is crucial to producing meaningful data and techniques should be thoroughly reviewed to suit the needs of the study.

4.2. Topic models with advanced topic relationships

4.2.1. Correlated topic model

The next major development in topic model methods was the "correlated topic model" (CTM), which was also proposed by Blei in conjunction with Lafferty in 2006 [6] (https://github.com/blei-lab/ctm-c); the provided code is from the author and utilizes variational inference, in the coding language C. Similar to the prior method, this approach considers individual documents as mixing proportions of topics and follows a nearly identical generative process; however, where LDA samples from a Dirichlet distribution, the topic mixtures in CTM are sampled from a logistic normal distribution. This distribution transforms a multivariate normal random variable on a simplex and through a covariance matrix of the normal distribution allows for an overall arrangement among the variance between the distribution elements. This was originally studied for the analysis of observed compositional data, but Blei and Lafferty extend its use to a hierarchical model to describe the latent composition of topics associated within each document in a corpus [6]. The natural parameterization of the K-dimensional multivariate distribution and is determined from the following equation:

$$p\left(z|\eta\right) = \exp\left\{\eta^T z - a\left(\eta\right)\right\} \tag{2}$$

In this Eq. (2), z can take on values of K and is signified by a K-dimensional vector. The mapping function of the mean parameterization, or the simplex, and this natural parameterization is expressed as:

$$\eta_i = \log \theta_i/\theta_k \tag{3}$$

Under the logistic normal distribution, η is assumed to be normally distributed and using the inverse of Eq. (3) it is then mapped to the mean parameterization, or the simplex. The process by which documents are generated in CTM is illustrated in the following plate notation graph displayed in Fig. 2.

In order to demonstrate the value of this method in its original publication, two topic model approaches – LDA and CTM – were applied to a corpus of 15,744 *Science* articles; the mean held-out log likelihood, a statistic indicating the likelihood of a particular result, of the two models was calculated and compared used to judge performance. The results showed that the CTM approach provided a better fit to the data and additionally was capable of supporting a greater number of topics overall. The performance of the LDA model peaked at 30 topics before significantly declining, while the CTM showed a peak at 90 topics, with minimal decline in log likelihood for topics counts greater than 90. The differences between the model statistics showed that CTM always gave a better fit compared to LDA based on the models' perplexity. It was seen that after observing a relatively small proportion of words within a document CTM had a significantly lower perplexity value, and therefore significantly more certainty, compared to LDA. This is because CTM uses topic correlation to support its prediction by inferring that words from related topics may also be likely within the document; conversely, LDA is unable to anticipate a document's remaining words until its topics are fully represented, requiring the observed proportion of the document to be significantly greater [6].

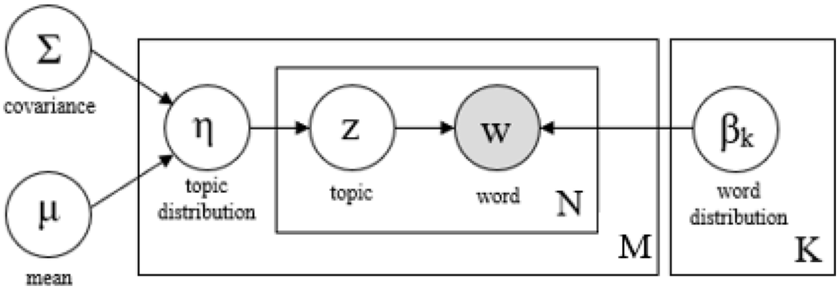

Fig. 2. A plate notation of the CTM approach. The designation of plates is comparable to those seen in Fig. 1.

This model is more expressive than LDA, and accounts for more realistic relationships between different topics within a document and over corpora. The use of a covariance matrix allows for a structure which can be utilized for more diverse tasks as well, such as exploring and navigating a large corpus of documents, forming predictive distributions, and optimized collaborative filtering [6]. The most challenging element of this approach is accounting for posterior inference, which is once again intractable to compute. This problem is similar to the problem encountered with LDA, however a greater variety of approximate inference algorithms are suitable for use in LDA. The suggested method of approximating this distribution is through the use of mean-field variational methods; these methods seek to form a factorized distribution of the hidden variables which are parameterized by variational parameters, which are free variables fit such that the Kullback–Leiber (KL) divergence is sufficiently minimized between the approximated posterior and the true posterior. In CTM, the logistic normal is not conjugate to the multinomial, therefore a variational algorithm could not be automatically derived. Instead an algorithm needed to be derived by taking account for the special structure and distributions used in CTM, which Blei and Lafferty present in their paper. Their finding can be found in the appendix A of [6]. Additionally, this method also assumes k is a known value and therefore does not offer a method of determining the optimal k for a corpus. The CTM is less sensitive to k compared to LDA [31], but this still presents an obstacle for researchers using this method. One noteworthy limitation of this method is that correlations can only take place between two topics at once; there is no consideration for larger or more complex document relationships. Overall, this is a suitable approach for most any data and would work well with sets which are expected to have strongly correlating topics, such as the example set gathered from articles within a single journal.

4.2.2. *Pachinko Allocation Model*

Around the same time as the publication of CTM, another topic-correlation model called the "Pachinko Allocation model" (PAM) was developed [9]; although the authors do not have a publicly available implementation, the TOMOTOPY toolkit for topic modeling in Python has a function for PAM as well as many other methods (https://github.com/bab2min/tomotopy). Unlike the previous model which was designed using the framework of LDA, the authors of the PAM method – Li and McCallum – took a different approach. Whereas CTM recognizes correlations between any two topics at one time, PAM creates a directed acyclic graph (DAG) mixture model in order to capture a variety of different kinds of topic relationships. This includes arbitrary, nested, or sparse correlations as well as topics over topics, forming classes of super- and sub-topics within a corpus. This is accomplished by considering topic as not just distributions over words but also as distributions over other topics, resulting in a hierarchy of topic relationships. The basic structure of this model is composed of a randomly generated DAG in which each word in the vocabulary corresponds to a leaf node and each topic relates to an interior node; the interior nodes of the graph can thus be seen as "parent" nodes, which are distributed over the "children" nodes (both leaf and non-leaf) which join to it [9]. Due to this design, the structure of a traditional topic in LDA can be considered an interior node which only includes leaves as its children nodes. Interior nodes which contain non-leaf nodes within their children represent a mixture over topics and captures correlations among both words and topics. It is possible to arbitrarily set parameters for the distribution of an interior node over its children; however, Li and McCallum instead propose using a Dirichlet distribution parameterized by a vector with dimensionality equivalent to the total children nodes which it encompasses, similar to LDA [9]. As a result, the PAM model is composed of DAG where the distribution of every parent node over its children nodes is represented by a unique Dirichlet. Documents are generated by first sampling from each Dirichlet to select a multinomial before generating each word within the document. Words are formed by starting at the base vertex of the DAG, moving down to one of its children nodes and sampling in correspondence to its multinomial, and continuing sampling down each level until there are no more edges to follow and a leaf node is reached; this process results in the generation of a single word, and is repeated for each word in the vocabulary [9]. The structure is highly flexible and depending on the layout of interior and leaf nodes can range in final form from a tree plot to an arbitrary DAG with complex features such as cross-connections and edges which pass over levels [9].

In this model, the combined probability of producing a document d, the topic assignments $\mathbf{z}^{(d)}$, and the multinomials $\theta^{(d)}$ is expressed as:

$$P\left(d, \mathbf{z}^{(d)}, \theta^{(d)} \middle| \alpha\right) = \prod_{i=1}^{s} P\left(\theta_{t_i}^{(d)} \middle| \alpha_i\right) \times \prod_{w} \left(\prod_{i=2}^{L_w} P\left(z_{wi} \middle| \theta_{z_{w(i-1)}}^{(d)}\right) P\left(w \middle| \theta_{z_w L_w}^{(d)}\right) \right) \quad (4)$$

where $\mathbf{z_w}$ is a topic path, z_{wi} is a topic in the path, and L_w is the length of this path. For this model, s is considered the number of topics (equivalent to k in previous models) and is separated into sets based on the levels of the DAG; higher level topics are called super-topics while lower level topics are considered sub-topics. The paper presents a four-level PAM structure, which was used to compare performance to a traditional LDA model. A representation of this structure and its generative process is seen in Fig. 3.

This model has notable advantages over CTM and LDA. Firstly, it not only accounts for topic correlation, it also considers more complex topic relationships and levels of correlation. It allows for versatile model structure which can adapt to a variety of different corpora. Additionally, when comparing the performance of PAM to CTM, LDA, and hierarchical Dirichlet processes (HDP),

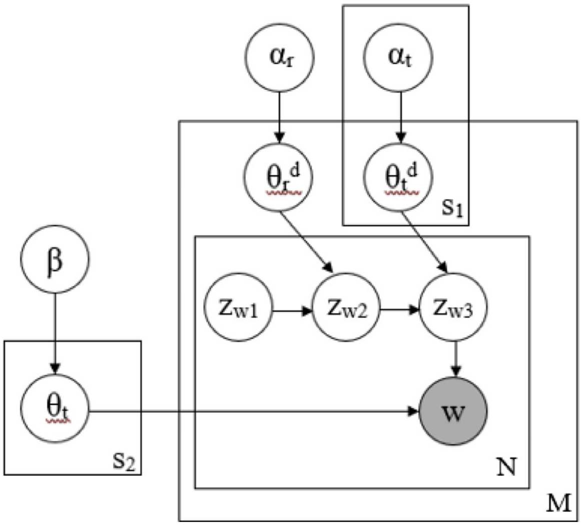

Fig. 3. The graphical model representation of four-level PAM structure. The plate labeled *s1* corresponds to the multinomial sampling iteration through super-topics, *s2* corresponds to the multinomial sampling for the sub-topics. The variables α_r and θ_r indicate the process of multinomial sampling for the root, the initial level of the DAG, which is done individually from the super-topics for each document. The variables z_{w1}, z_{w2}, and z_{w3} indicate the topics within a topic path to word *w*.

PAM demonstrated the ability to support significantly more topics compared to the next best-performing model, CTM. While CTM performance peaked at 60 topics and was seen to out-perform PAM at lower topic numbers, PAM performance peaked at 160 topics and performed significantly better over a greater range of topic numbers. However, since it is still founded on the same primary probabilistic algorithm, there is still no mechanism for learning the value of *s* (*k* in LDA and CTM) based on the corpus. Therefore, optimizing the number of topics is remains an inefficient process; this is compounded in PAM by the addition of a hierarchy over topics which thus require more consideration for model structure. The authors suggest using Gibbs sampling as the preferred method of inferring and estimating parameters in this process over other variational approaches because the former algorithm has been shown to struggle with PAM topic modeling tasks due to the many local maxima. Since in Gibbs sampling the Dirichlet parameters α are presumed to be given but in PAM these parameters need to be learned to accurately represent relations between sub-topics, α is approximated using moment matching to avoid iteration and simplify the process [9].

PAM may involve more optimization and computational expense than LDA, but the relationships it models are unique in that it does not just consider single relationships between two topics like CTM, it considers hierarchies and levels of topics. This is most likely best suited for studies where a detailed topic structure is desired as a result, such as topic modeling for consumer trends or very broad focus datasets for the purpose of pattern discovery.

4.3. Time-based topic models

4.3.1. Dynamic topic models

When modeling topics in a series of documents, there are other assumptions which LDA makes which is not necessarily how data interacts. For instance, within a corpus in LDA all documents in a corpus are treated interchangeably [11]. For sets which span long periods of time, this may not accurately reflect the changes that occur in topics over time; for instance, a paper about a specific disease in the early 1900's would discuss very different things than a paper about the same disease in the 2000's. It was for such scenarios that the dynamic topic model (DTM) was also developed by Blei and Lafferty, which considers the "evolution of topics over time" [8] (https://github.com/blei-lab/dtm); the provided code is from the author and is composed of mostly shell, C++, and C code. The consideration of evolving topics was performed using state space models on the natural parameter space for both the underlying topic multinomials and for logistic normal distributions, which will be used for modeling document specific topic proportions. Its basic approach was to chain together a sequence of models based on a defined unit of time, *t*, which could range from a period of days to a period of years depending on the needs of the dataset. This unit is considered a 'slice' of time and is used to form distinct K-component models in which topics produced for slice t are developed from the topics found for slice $t-1$ [8].

Considering a model with V words in the vocabulary, $\beta_{t,k}$ denotes the V-dimensional vector of natural parameters for a topic k within slice t. Using mean parameterization to represent the V-dimensional multinomial distribution with the mean parameter denoted by π, the ith component of the natural parameter β is given by the mapping $\beta_i = \log(\pi_i/\pi_V)$. Since the Dirichlet distribution is not suitable for sequential modeling, the natural parameters of each topic are chained using a space state model that evolves by Gaussian noise, the simplest version of which is modeled by the following function:

$$\beta_{t,k}\left|\beta_{t-1,k} \sim \mathcal{N}\left(\beta_{t-1,k}, \sigma^2 I\right)\right. \tag{5}$$

In DTM, uncertainty over document-specific topic proportions is expressed using a logistic normal with mean α. The sequential structure between models is once again represented with a simple dynamic model:

$$\alpha_t\left|\alpha_{t-1} \sim \mathcal{N}\left(\alpha_{t-1}, \delta^2 I\right)\right. \tag{6}$$

Through this approach, DTM ties a collection of topic models sequentially by forming a link between the distributions of both topics and topic proportions. Considering a serial collection of documents, the process for generating slice t therefore begins by drawing topics from Eq. (5) and then drawing distributions from Eq. (6). Then in document d, the mean parameter of multinomial distribution $\pi(\eta)$ is used to draw a topic Z for each word and word $W_{t,d,n}$ is drawn from the mean parameter of the multinomial distribution $\pi(\beta_{t,z})$. The generative process is illustrated in Fig. 4.

Similar to both LDA and CTM, the posterior inference is intractable and therefore needs to be approximated; since this model is not static like LDA or CTM, the approach for this is quite different. The authors suggest two different approaches to calculate approximate inference for the natural topic parameters, and state that the variational distribution of document-level latent variables follows the same calculations as used in LDA. The two methods for natural parameter inference are variational Kalman filtering, which considers the symmetrical properties of the Gaussian density to yield variational parameters, and variational wavelet regression [8]. The application of both approaches to data from the analysis of articles from 120 years of *Science* journal, 250 articles per year, showed that both approximations smoothed out local fluctuations compared to the control unigram model, however wavelet regression was capable of super-resolving very near spikes in topics whereas Kalman filtering provided a better fit overall for more dispersed topic timelines. The variational wavelet regression approximation may be best applied to sequential corpora which span shorter time periods and therefore require better resolution between peaks, or when there is an expectation of issues with localized topic resolution. Kalman filtering would be more suited for longer timespans

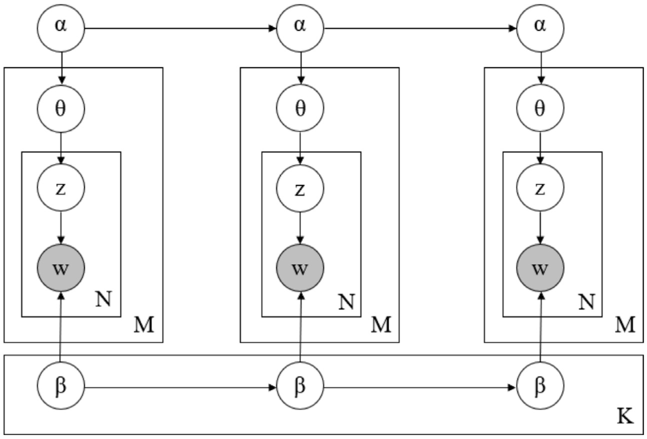

Fig. 4. A plate notation diagram for DTM approach. Horizontal arrows represent the evolution of parameters between slices of times (ex. t to $t-1$), while vertical arrows represent the progression of the generative process through each topic model in the sequence. The variables in this graph are the same as the ones seen in LDA, however the distributions in DTM are Gaussian, not Dirichlet.

which require a broader perspective of the data and where there is the expectation of a more even topic distribution.

The way DTM represents the passage of time is of some concern; this method requires the user to set a discrete unit of time. This brings questions of how t is selected and what characteristics of the data is lost by discretization of time over the corpus. Additionally, this model does not account for correlations between topics for simplicity, though a combination CTM–DTM approach is theoretical possible as CTM is not drawn from a Dirichlet distribution [8]. This approach does not provide an immediate method of determining the optimal k value for the dataset, similar to all previous methods. This process may be more difficult with this topic modeling approach due to the variation between data within time slices. The paper presents a 'perfect' scenario with an equivalent number of documents for each year, which could easily be reproduced for other studies; however, it does not describe the obstacles involved with a more realistic distribution of documents over time. However, for a filtered or specifically structured corpus this method can provide a much more informative topic model than LDA alone.

This approach to topic modeling is well suited to corpora which span significant time periods, at least several years. Sets which cover less than a year would likely not benefit from the evolution of topics as changes in topic distributions would be much less significant and more difficult to represent in distinct time slices; however, this does not mean that there are not short time span sets which would benefit from this approach. Significant number of papers have utilized this method for research involving topic patterns for academic publications, author citations, sequential or timeline overviews of topic development, and combined topic modeling methods [32]. It has also been highly cited in works involving studies on social media and online trends [33]; however, such tasks would likely be better suited to the methods described in the following section, which account for poor document length. It is theoretically possible to apply both DTM and the approaches within Section 4.4, although these methods are still quite new so no such work has been done. Research focusing on data which is not highly influenced by external events such as disasters or current events or is generally released in a structured sequential pattern could benefit from DTM, as it by design provides models in discrete units which can easily be compared.

4.3.2. *Continuous-time topic modeling*

As previously mentioned, there are several concerns with using a discrete time model for topic modeling. Although it may simplify computational processes, enforcing a fixed scale for creating models is not an accurate representation of how topics may change over time. For instance, topics within social media and networks may change very quickly as a result of viral trends but may still be influenced by reoccurring temporal events, such as holidays or elections, or by unpredictable external or global events, such as war, civil unrest, and disasters. All these influences may span different periods of time or co-occur during the same time periods. For basic LDA models and correlated topic model algorithms, these kinds of occurrences can confound data and lead to poor topic coherence and high levels of perplexity. Although DTM may be capable of expressing some degree of the topic evolution of in these situations, it cannot realistically map the changing patterns over time. In order to accurately model complex temporal topic patterns, time must be viewed as a continuous entity.

Several different methods have been developed in topic modeling which consider time in a continuous manner. As with previous models, there is a Bayesian design which considers time continually by the application of a mixture of concepts from homogeneous Markov processes and Bayesian networks called Continuous Time Bayesian Network (CTBN) [33]. This method provided improvements over methods such as Dynamic Bayesian networks (DBN) models by avoiding discretization but is not directly designed for the process of topic modeling. It also assumes at least one fixed point of knowledge over state transitions in a dataset, which may not be known when performing real world data analysis. A much more suitable method for topic modeling was designed using a Dirichlet per-document multinomial distribution of topics in combination with a Beta per-topic distribution over time. This method allows not only for the consideration of temporal patterns within topic models, it also can account for correlation between topics on a one-to-one basis and the changes in these co-occurrences over time. This approach was proposed by Wang and McCallum and is called Topics Over Time (TOT) [34]; the authors do not provide a public implementation, however an open-source Python code was made by Abhinav Maurya of Carnegie Mellon which directly relates to the one described in the paper (https://github.com/ahmaurya/topics_over_time). TOT has two basic methods for producing a model. The first is better suited for seen data, or data which is known to the algorithm. Here, for each topic, k multinomials (ϕ_z) are first generated from a Dirichlet prior, β, and per-document multinomials (θ_d) from a Dirichlet prior, α. Within a given document d, a topic z is extracted from θ_d for each word. A word w_{di} is then extracted from the multinomial ϕ_{zdi}, and finally a timestamp is extracted from the Beta distribution ψ_{zdi}. Although timestamps are obtained for every word, the document timestamp is viewed to be equivalent to the timestamps of all the words within that document. Gibbs sampling is used here to carry out approximate inference. The other approach used in TOT is more suited for unseen documents, such as would be encountered in large scale data analysis tasks and unsupervised learning tasks; this the procedure which is represented in Fig. 5. Unlike the prior approach, only one timestamp per document is generated from a combination of the per-topic Beta distributions over time, where the weight of a mixture is the per-document multinomial over topics which has been sampled from the Dirichlet distribution β. The model is thus allowed to predict a time assignment given the word contained within a document. The introduction of a balancing hyper-parameter, such as used for smoothing functions in LDA, makes the processes of parameter estimation between the two approaches nearly equivalent and rescales the likelihood for the combination with different models [34].

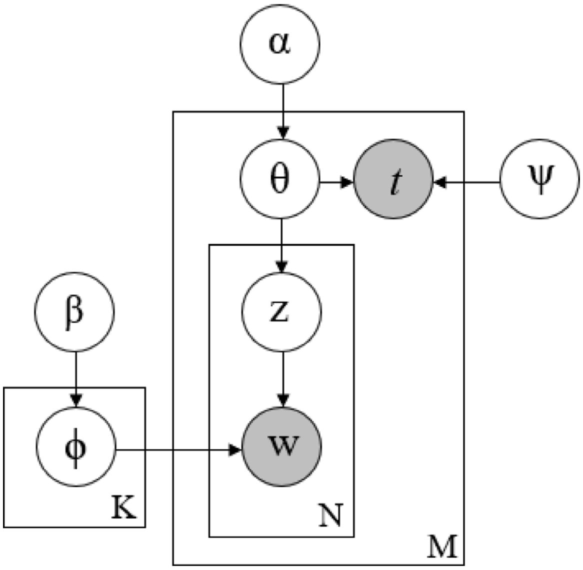

Fig. 5. A plate notation diagram of TOT process suited for encountering unseen data. In this graph, t indicates the timestamp of a single document while ψ describes the per-topic beta distributions over time. All other variables are similar to those seen in previous figures.

Although this method has been highly cited in other works and demonstrates great versatility in its applications, its original proposal only compared the performance of TOT against an LDA model and not against more comparable methods such as DTM or CDM topic modeling. The authors did discretize the LDA model into year units, however this was done post-hoc and does not include considerations for evolving topic distributions as was used in DTM [34]. This limits the ability to accurately assess the improvements of this method, as LDA is not designed for sequential data modeling and therefore would not be expected to perform well. The comparison of KL divergence scores between TOT and LDA over three distinctive corpora demonstrated that TOT was able to produce topics which were more differentiated from each other. Additionally, histograms of topics over time and beta distributions for the TOT models compared to the most similar LDA models produced showed that TOT was able to form highly resolved topic peaks in time. This allowed for very specific isolation of temporal patterns relating to historical events; for instance, the authors demonstrate that TOT can isolate the topic of the Panama Canal to within the years surrounding the turn of the 20th century, whereas the comparable LDA formed large peak structures around both 1850 and 1900 with significant signals up until 2000. Other corpora tested in the paper also demonstrated TOT applications in other fields such as the analysis of academic papers, email and correspondence documents, and written speeches [34]. In comparison to other topic models in computational difficulty, TOT is simple which gives it a benefit over some models which require several stages of parameter estimation and optimization. Like previous models however, k is assumed to be known for calculations which means that finding the ideal topic count requires additional steps.

Considering the results of the authors' own comparison and the applications of this method in literature, this approach is suited for fields such as social media analysis, news media analysis, and online studies. Although there have been many studies utilizing this approach for the analysis of academic archives and author–document studies, DTM has also been equally if not more popular for such tasks. This model would be recommended for any research which is highly dependent on complex or irregular temporal events, historical events, and viral trends. Additionally, TOT is well suited for combination with other topic modeling processes or with other probabilistic models to extract more abstruse measures, such as burstiness and spatiotemporal patterns. Overall, this approach is highly versatile and could be used for nearly any application, however its value for temporal and correlated data analysis should be considered predominantly when selecting over other topic modeling processes.

4.4. Short text optimized topic models

Although topic modeling is frequently used for online document classification and trend analysis, it has been shown that traditional topic modeling approaches struggle when it comes to documents gathered from social media sites, which are often subject to a character limit and thus have a significantly shorter document length [3]. The reduction in effectiveness of topic modeling in these situations is the result of the very limited set of co-occurrences between words which occur in documents of such a short length compared to other more traditional document sources. There have been many approaches developed to deal with these sparsity issues, and significant research is being done in this particular field of topic modeling. several strategies which have been proposed are directly considering word co-occurrences instead of modeling document identities [7], considering each short document as arising from one latent topic [35], and considering heuristic ties between documents in order to string them into larger "pseudo-documents" [36,37].

One early proposal for short text modeling was a process called a mixture of unigrams model (MU), which regarded each short document as being the result of sampling from a single latent topic [11,35]. This approach demonstrates significant improvement over LDA [7] and has shown to perform competitively with newer can more complicated methods [36], but does not reflect the reality that a short text can cover a variety of topics. Another similar approach was proposed called the biterm topic model (BTM) [7] (https://github.com/xiaohuiyan/BTM); the provided code is from the author and is written primarily in C++ and some Python. BTM directly models the generative process of co-occurrence; this is also similar to the Bigram model, which considers associations between words by applying the concept of word order to the modeling process; however, the Bigram model also uses co-occurrences as a statistic for supporting topic production instead of modeling the co-occurrences, and puts more significance on word order rather than co-occurrence [7]. Although BTM was intended to maximize the usage of co-occurrences in short texts, it often could not differentiate between short and long documents [37]. Another method called the dual sparse topic model (DSTM) replaced the Dirichlet prior in LDA with what is called the Spike and Slab priors, which in theory learns focused topics and terms for a document [38]. Both of the aforementioned approaches add little in the means of word co-occurrence and therefore still encounter data sparsity issues. As such, we will not be providing a detailed explanation of these methods in our review. The mixture of unigrams model will also not be described as we are seeking to give preference to more recent and advanced modeling methods for this section.

4.4.1. Self-aggregating topic models

The 'pseudo-document' approach has been shown to be significantly more efficient when encountering such sparsity problems. Many methods of this are documented in social media research such as twitter studies, and commonly is aggregated by some observable metric such as author, hashtags, or location [3]. These methods do not require significant changes to the modeling

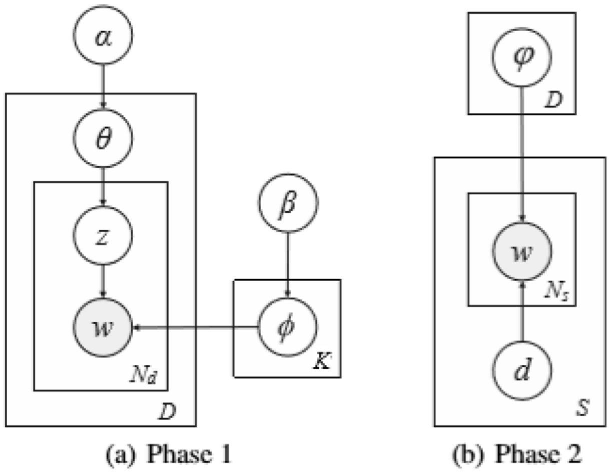

Fig. 6. From Quan et al. a plate notation diagram representing the generative process for SATM. Here, *D* indicates the number of pseudo-documents while *S* represents the short texts present in the original data corpus [36].

process, as aggregating data around given parameters can be performed easily by data preprocessing. Although such simple aggregation has been beneficial to specific studies with special interest in modeling topics around users or locations, this is not applicable to more general short text data where such information may not be readily available. The development of generalized aggregation methods for short texts have been developed more recently, and overcomes the obstacle of limited context. The first of these approaches is called the "Self-Aggregation based topic model" (SATM), developed by Quan, Kit, Ge and Pan in 2015 [36] (https://github.com/WHUIR/SATM); this code is from Wuhan University Information Retrieval Group and is written in Java. This model does not aggregate by author or by keyword metrics, such as hashtags, but instead aggregates short texts with similar topics. In a corpus, each original short text is presumed to be sampled from a long unobserved pseudo-document which is assigned by finding their "documentship"; this method is accomplished by integrating the modeling process and the self-aggregation process during the course of topic inference, initially basing aggregation upon the general topic affinity of texts as applicable to various short text formats. There are two phases in model generation, the first of which follows the assumption of LDA to generate *D* documents of standard length; an individual generated document *d* is made up of a string of words $\mathbf{w}_d$ with a document size of N_d, representing the total number of words. This corresponds to the development of the long pseudo-documents. The second phase, each of these documents is used to generate a few short texts with each short text belonging to exactly one document following the assumptions of a mixture of unigrams. A short text snippet *s* is composed of a sequence of words $\mathbf{v}_s$ made up of N_s total words. This process is laid out in Fig. 6.

This method was evaluated using the NIPS dataset to produce 200,879 short text pieces, treating the original documents in this set as the latent long pseudo-documents to compare performance against. A collection of 88,120 questions gathered from Yahoo! Answers, which were taken as short texts was also used for evaluation. For the NIPS set, the model was judged based on its ability to produce topics from the set of short pieces which were comparably meaningful to topics generated from the original long documents. It was demonstrated that SATM outperformed both the Unigram and K-means clustering approaches in aggregating texts. Additionally, SATM consistently outperformed LDA, Unigram, and BTM at forming topics which aligned with the original long documents at three different *K* values. The performance of SATM using the short text corpus gathered from Yahoo! showed that although it outperformed BTM and LDA, it performed significantly better than either of the other models when first trained with an SVM classifier using the question categories taken from the Yahoo! Answers site. When considering post inference, BTM performs competitively but still under SATM [36].

Despite this model's strong performance in its proposal paper, there are many setbacks for this particular approach. First, this model employs additional latent unobserved variables which must be considered when optimizing—the number of pseudo-documents. This is discussed within the paper, and it is proposed that the number of pseudo-documents which best suits a dataset likely depends on the optimal topic count; therefore, to find the optimal value for pseudo-documents, more iterative testing would be needed which requires a large amount of computation for each iteration and also makes determining the optimal *k* more difficult as well. The total variables in SATM increases as larger datasets are used, making the model susceptible to overfitting and resulting in a time complexity which is unacceptable for practical use. Although more work is needed to improve this model, it does demonstrate potential and is worth considering for those with the means to perform such algorithms. However, for work related to social media or other context-rich documents, simple aggregation methods such as author or keyword aggregation would be more suitable.

Building off the work of SATM, a new method called the "Pseudo-document-based topic model" (PTM) was developed by Zuo et al. in 2016 [37]; there is no public implementation provided by the authors, however this model has been incorporated into another project by. This method reduced the self-aggregation of short texts into a single generative process [37]. Here the number of documents will be represented by *P* while the number of short texts is represented by *D*. The multinomial distribution of short texts over documents is represented by ψ, which is sampled from a Dirichlet λ. Whereas sampling a word within SATM costs *O(PK)* time due to its two-step process, the cost of sampling a word in PTM is *O(K)* and therefore is less intensive [37]. Also, the second step of SATM means that the probability of pseudo-documents on short texts must be estimated by the inference process independently, resulting int the linear increase of parameters with increasing corpus size; this is what causes SATM to be prone to overfitting in cases of limited training data. In comparison, PTM produces short texts following the procedure utilized in LDA given the singular pseudo-document to which they belong. Each pseudo-document is essentially a "hybrid topic" generated from the individual topics of the multiple short texts which it contains [37]. This unfortunately means that in instances where the total number of pseudo-documents is reduced, topics tend to be less specific and meaningful. Therefore, considering scenarios where *P* is small, Zuo et al. also propose a sparsified version of PTM which utilizes the Spike and Slab prior, called SPTM [37]. This method is significantly more complex than PTM and will not be described in detail, but can be found in [37]; the generative processes for both PTM and SPTM are provided in Fig. 7.

Using data from news websites, titles for conference papers in several computer science fields, questions asked on a particular website, and labeled website-linked tweets, it was demonstrated that SPTM outperformed LDA, MU, and DSTM consistently at short text topic modeling, although other methods may have been competitive. SATM greatly outperformed SPTM at modeling tweets and narrowly outperformed SPTM at modeling when the amount of training data was small [37]. This method shows great potential and has been frequently cited in recent works; however, even within its proposal paper it was outdone on several of the datasets and often was very close to the performance of other methods. It does represent a much more cost-effective solution to modeling short texts which have limited context compared to SATM, and therefore should certainly be considered if one is seeking to model such data.

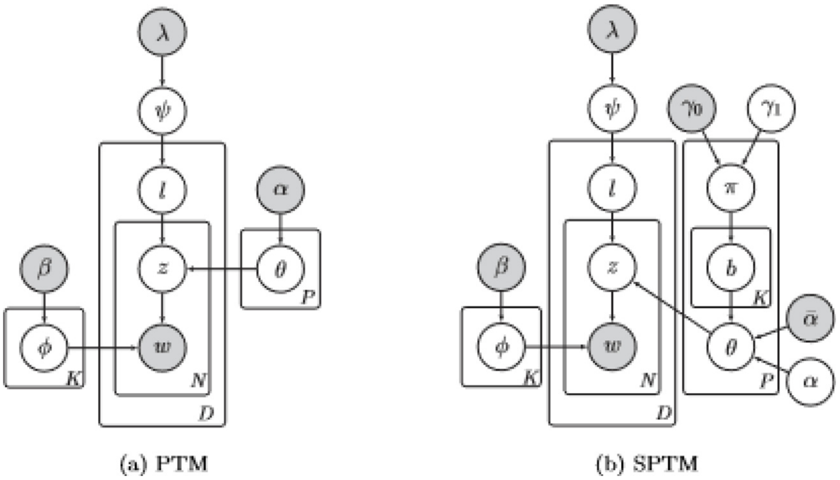

Fig. 7. From Zuo et al. a plate notation depicting the generative process for both PTM and SPTM. The variable *l* represents a pseudo-document when generating short texts [37].

4.5. Other significant topic model designs

An alternative algorithm to LDA is "Non-negative Matrix Factorization" (NMF) [39], developed in 1999 by Lee and Seung. Although it preceded the proposal of LDA, this method has been shown to work effectively for extracting topics from text corpora [40]. NMF is unsupervised, compared to LDA which can be either supervised or unsupervised, and utilizes methods of dimensionality reduction for 'non-negative matrices', or a matrix made up of only positive or zero value components [39]. This is similar to many of the approaches which topic modeling evolved from, such as *tf–idf* reduction [12] and LSI/LSA [13]. Conversely to *tf–idf* reduction, a corpus here is considered as a document-by-term matrix of V-by-D dimension and this approach seeks to approximate this matrix as the product of two k-dimensional non-negative factors containing the membership weights of their components, which can be called **W** and **H** [39,41]. The rows of **H** can be viewed as the weights for every respective term of the V words contained in the vocabulary of the corpus, representing k topics, giving it a dimensionality of k-by-V. The columns of **W** then represent the weights relating to each of the D documents in relation to each topic, giving it a dimensionality of D-by-k. If one takes **H** and orders each of its rows, a ranking of the terms in relation to that row's topic is formed thus providing a topic descriptor. This method is generally initialized by the random assignment of weights to the elements of **W** and **H**, and then iteratively improving these factors by applying an optimization algorithm aimed at reducing the error in approximation until the process reaches a localized minimum [41]. While NMF has fewer parameters which need to be selected for the modeling process and has been shown to distinguish more realistic topics than LDA, the initial values for the topic-term and topic-document weights have great influence on the final results during optimization which have been show to persist through iterative runs of the model [41]. Since these values are randomly assigned, this results in high levels of inaccuracy and a lack of reproducibility for final factors. An improvement to this approach was the "Non-negative Double Singular Value Decomposition" (NNDSVD), proposed by Boutsidis and Gallopoulos in 2008, which selects pre-modeling factors through an estimation of the corpus matrix and the use of sparse singular value decomposition [42]. Theoretically, NNDSVD could converge to identical factor pairs between individual model runs depending on which application of SVD is employed, and has demonstrated significant efficiency when using text-based data and other sparse sources [41].

There are several other methods have been published for improving topic models which are worth noting to those seeking to perform such data analysis. One such technique is the "Hierarchical Dirichlet Processes" (HDP) which was presented by Teh et al. [43] (https://github.com/blei-lab/hdp); the provided code is from the author and utilizes Gibbs sampling inference, in the coding language C++. This is a nonparametric Bayesian modeling approach, specifically designed for handling issues with clustering of multiple datasets and for "multi-task learning" using the Dirichlet process (DP) mixture model. This method automatically learns the appropriate number of mixture components to use for a dataset and can share statistical strength across groups of data [43]. It has been shown that this can be applied with an LDA model as well as with other topic model approaches to facilitate automatic topic learning, or the ability to find the ideal topic number without user selection, [44,45]. However, in order to utilize this functionality, two models must be constructed for a single corpus, which can further complicate the topic modeling process by requiring two stages of inference algorithm calculations and parameter estimation to reach the ideal model design. The addition of a nonparametric Bayesian prior process for determining s in PAM was also suggested by Li et al. [9], which is developed from a variation of the HDP approach; however, such a process encounters the same issues with efficiency and computing demands as it did in LDA. Of the discussed topic modeling methods, only HDP is designed in such a way that the algorithm learns the best number of topics automatically and does not need to be fed a k value by the user [43,45]; however, this method does still require significant optimization to produce desirable results [45].

Other models which have not been discussed include those which consider the order of words. A few examples of these are the hierarchical Dirichlet language model (HDLM) [46] and the bigram model [47]. Because the inter-document relationships are often of greater interest than intra-document relationships in big data analysis, these methods were not described in detail for this review. If word order is of interest for a particular dataset, these methods should be studied.

5. Improvements and optimization of topic modeling

5.1. Inference and approximation methods

It is well established that many of these models, particularly LDA, require tuning and optimization in order to produce reliable results from a dataset. For this reason, inference and approximation of hidden variables is highly recommended over default settings which may be included in topic modeling software [5]. The posterior distribution is incomputable in all of the previously mentioned methods, and therefore exact inference is not possible; however, these distributions are necessary in order to use these processes. There are a variety of approximation algorithms which will work for many of these topic modeling methods, although several methods may be better suited for specific types of algorithms over others. We shall describe these algorithms here and discuss which methods they would suit best under certain circumstances. We encourage reviewing the original literature for any topic modeling methods of interest for the suggested approximation methods, as they are often adjusted to better perform with the model.

5.1.1. Markov chain Monte Carlo and Gibbs sampling

A common set of approximation and inference algorithms for topic modeling are from a family of methods called Markov Chain Monte Carlo (MCMC). These approaches are given this name because they utilize Monte Carlo integration, a method developed by physicists which use randomly generated numbers to compute integrals, to form a "Markov chain"—. By "Markov chain", we mean a string of arbitrary variables whose transition probabilities among distinct values in the state space are solely contingent on the existing state of the random variable. Thus, this process uses the value of the preceding sample to produce the value of the subsequent sample at random; the goal of this approach is to simulate directly drawing from a complicated probability distribution. The MCMC methods arise from the concepts of the "Metropolis–Hastings algorithm", which is able to generate a Markov chain from complex probability distributions which are otherwise difficult to sample, such as high-dimensionality distributions [48].

The most popular approach in Bayesian inference is the Gibbs sampling method. This is also derived from the MCMC structure of algorithms, considered to be a special case of Metropolis–Hastings with a Hastings ratio always equal to 1 [48]. Gibbs sampling seeks to generate a Markov chain with the desired posterior distribution as its stationary distribution, meaning that after several iterations drawing samples from the stationary should approach drawing from the posterior distribution of choice. Specifically, Gibbs sampling works by sampling from conditional distributions of the posterior variables. Given a joint distribution $p(\mathbf{x}) = p(x_1, \ldots, x_m)$ with no closed form solution for $p(\mathbf{x})$ but provided with a representation for conditional distributions, Gibbs sampling first randomly initializes each x_i. Then for each iteration t we follow the following process [48]:

$$x_1^{t+1} \sim p\left(x_1 \middle| x_2^{(t)}, x_3^{(t)}, \ldots, x_m^{(t)}\right) \tag{7}$$

$$x_2^{t+1} \sim p\left(x_2 \middle| x_1^{(t+1)}, x_3^{(t)}, \ldots, x_m^{(t)}\right) \tag{8}$$

$$x_m^{t+1} \sim p\left(x_m \middle| x_1^{(t+1)}, x_2^{(t+1)}, \ldots, x_{m-1}^{(t+1)}\right) \tag{9}$$

This is repeated until the samples begin to converge to a sample that would be taken from the true posterior distribution. In practice, this technique performs well and is theoretically guaranteed to converge; however, it is not possible to know how many iterations are necessary to adequately approach the stationary distribution. Convergence is typically estimated by calculating the log-likelihood and in some cases visually observing the plotted data points to judge if the model has reached an equilibrium-like state or 'stationarity'. Once the sampling has reached this stationarity, any fluctuation between data points is considered standard error [48]. Often samples from the beginning of the chain may not accurately represent the desired distribution and are discarded before beginning to collect data; this span of data is generally called the "burn-in" or "warm-up" period; furthermore, any data points discarded are generally called "burn-in" values [48,49]. Although this process has been recommended by many in the field, it is not necessary to reach convergence; at most it reduces the amount of data which needs to be collected for evaluating convergence in scenarios where the algorithm has sampled from a far end of the probability distribution [49]. The general practice in the field is to use a burn-in period of no more than the initial 1 or 2% of the total number of runs, or to calculate a statistic called autocovariances and use the approximate time it takes for this value to decay to insignificant levels [49]. When considering multi-modal target distributions there is the possibility for what is called poor mixing, which is a chain which appears to be confined to a small region of the parameter space [49]. This can be combated by either utilizing multiple initial values which are well dispersed to start many simultaneous chains [49], or to perform simulated annealing on single chain [48]. Simulated annealing essentially mimics the pattern of a cooling crystal structure to allow for more variation in movement at the beginning of the "cooling" period, and then gradually reduce the probability for down-hill movements over time. This process is described in detail by Walsh [48] and should be reviewed if visual observation of the sampling process appears to be poor mixing, if the autocovariance values remain significant, or if the target distribution is known to be multimodal [48,49]. Well mixing chains are chains which demonstrate a high degree of exploration over the parameter space before converging to equilibrium.

The basic Gibbs sampling method is directly recommended and utilized for approximating posterior inference in the publications for several topic model approaches, including PAM, BTM, and HDP [7,9,43]; in SATM, the process is recommended twofold throughout the method to append short texts as special states in order to better model them [36]. In a model such as LDA, users are interested in the hidden latent variables—the document-topic distributions θ_d, the topic-word distributions $\phi^{(z)}$, and each word's topic index assignments z_i. Although, traditional Gibbs sampling can be used for LDA by deriving the conditional distributions of these latent variables there is an easier method. As $\mathbf{z}$ can be used as an appropriate statistical measure for both θ_d and $\phi^{(z)}$ and thus can be estimated using z_i, the multinomial parameters can be integrated out and sampling can be performed on z_i alone. This is called a collapsed Gibbs sampler technique and can be in LDA and its direct derivative methods [25] and is also suggested for PTM [36]. A similar approach is also taken when using Gibbs sampling in TOT, except sampling is done from the beta distribution ψ_z instead of the topic index assignment [34]. Applications for Gibbs sampling have also recently been proposed for approximating the posterior in DTM using a block-wise approach; the proposal, by Bhadury et al. suggests a parallelizable inference algorithm which combines Gibbs sampling with Stochastic Gradient Langevin Dynamics (GC-SGLD) [50]. This approach proves to be much faster than the approximation algorithms recommended by Blei et al. which utilizes variational Kalman filtering and wavelet regression [8]. As this new algorithm is much more scalable than previous methods, it allows for larger topic dynamics to be captured using DTM which opens up this topic model approach for more industrialized usage. For any researcher considering the use of DTM for Big Data analytics, it is highly suggested that this publication is reviewed as a possible method of improving the speed as well as the performance of topic modeling.

MCMC algorithms and Gibbs sampling are generally very versatile and can easily be adapted to many different applications. These approaches offer a calculated approximation to the exact posterior distribution using a collection of samples, and often performs well for sampling from difficult probabilities. However, this method does not allow for convergence time to be easily assessed and can suffer from slow mixing; in these cases, methods such as multiple starting points or simulated annealing needs to be considered to amend the course to convergence. Generating many small chains has been speculated as a method for those with access to parallel processing; however, it is not ideal as it is unlikely that convergence will be able to be reached for many of the runs [49]. Instead, it is recommended to run several short runs to determine the length of iterations needed to approach convergence and then running one or more longer runs to accurately converge to the desired distribution. It is also recommended that when performing multiple runs to begin each new run from the ending point of the previous run; this is recommended under the assumption that the sample from the previous run was randomly distributed but was in the process of converging to the target distribution [49]. A similar rule is recommended for seeding random number generators, which states that attempts to restart the process with a "random" seed will destroy the randomness of that generator and only when used as a continuous stream does the generator retain the properties that made it desirable to the user.

5.1.2. Variational approximation

The concern with MCMC inference is that for high-dimensionality datasets or noncongruent models, this method is not computationally feasible and takes significant lengths of time to perform [51]. Variational Bayesian (VB) approximation methods are therefore recommended over MCMC for large datasets and mixture models [51–53]. VB methods differ from MCMC sampling methods when performing statistical inference in one key manner; whereas MCMC methods attempt to approximate a numerical solution of the exact posterior distribution using a sample set, VB approaches instead attempt to provide an exact analytical solution in the form of a discrete formula to an approximation of the posterior. This can be described as applying a group of densities over the unknown variables which are parameterized by "free variational parameters", and optimizing these using a specific metric called Kulback–Liebler (KL) divergence to find which of these parameters are most similar to the conditional density of interest. Minimizing this metric is equivalent to maximizing the evidence, or the marginal density of observations [51]. These methods are often more computationally manageable than MCMC but come at the cost of the theoretical guarantees of simulation methods; however, there are a variety of criteria for evaluating the convergence of these algorithms. There are many different approaches which fall under this class of inference algorithms. In Blei et al. the variational Bayesian algorithm recommended is a mixture of a traditional variational expectation–maximization (EM) framework and a variational maximum likelihood (ML) approach [11]. The CTM is recommended with mean-field approximation methods [6], and DTM is recommended with for the variational Kalman filtering approach as well as the Wavelet regression approach under the same variational distribution of document-level latent variables as LDA [8]. There also has been a proposal by Teh et al. describing the process of using a collapsed variational Bayesian inference algorithm, which combines elements of both variational Bayesian methods and collapsed Gibbs sampling, for LDA [52]. We shall start by discussing similar approaches which are often referred to during these approximations.

VB approaches can be regarded as an expansion on the "Expectation–Maximization" (EM) algorithm [51]. This algorithm alternates between an "E" step, in which the distribution of latent variables is inferred by assuming the present estimated parameter that describes the stochastic dependencies between variables, and an "M" step, which maximizes the likelihood found in the E step by re-estimating this parameter [51,53]. This process continually updates and at each iteration the E step maximizes the lower boundary in reference to the distinct distribution for the current data point, and the M step maximizes the variables' dependencies [53]. The use of EM for topic modeling is often limited to constrained or approximate optimization as the posterior distributions for these methods are intractable [53]. In this approach the posterior distributions are constrained to be of a tractable form, such as factorized over a variable, and the maximization during the E step serves to also minimize the KL divergence between the variational distribution and the exact hidden variable [53]. Refer to Beal for a detailed explanation of the algorithmic process and several optimizations relevant to working within a topic modeling setting, such as variational Bayesian EM, hyper parameter optimization, and its application in several types of probabilistic models [53]. This is an iterative approach and therefore can be time-consuming when working with large sets of data; there has been significant research on possible methods of accelerating EM algorithms, such as employing a partial E step on a portion of the total dataset to allow for more frequent M steps to be performed, thus more frequently update parameters [54]. Although this method could potentially speed up convergence with little overhead, it is difficult to perform in parallel due to the need for clustering features to be updated globally. This requires synchronization of workers within a distributed environment which can result in significant overhead; however, Yin et al. discuss how controlling the frequency of parameter updates can improve the performance of parallel EM algorithms [54]. This new approach can be used in one of two ways—"partial concurrent" or "subrange concurrent". In the first method, every E step estimates the hidden variable distribution within a block of the total set and the size of the block will determine how frequently the parameters update. The subrange method works in a similar fashion, except each E step is responsible for processing the distribution of hidden variables within a subrange and the size of the subrange determines the update frequency. Both approaches were shown to maintain the EM algorithm's convergence properties, however the partial concurrent approach performed best compared to the traditional concurrent as well as the subrange concurrent approach for LDA [54].

Another method within this class of approximations is the mean-field approximation method. This method is best suited for models which are known to have a dense distribution [55]. It assumes that there are many influences of each node of a graphical model from surrounding nodes to the point that every individual effect is small and the overall influence can be considered appropriately additive. Thus, every node can be roughly represented by its average value, principally in cases where the "law of large numbers" can be utilized [55]. Each latent variable under this assumption is considered independently of other latent variables and each is distinctly factorized in the variational density [51]. A variational factor can undertake a parametric form which is suitable to its matching random variable—such as a continuous factor for a continuous variable or a categorical factor for a categorical variable. In mean-field variational methods, the objective is to form a factorized distribution of these variables which are parameterized by free variational parameters [55]. As with EM, the parameters are fit to the data such that the KL divergence value is minimized; however, it requires some optimization in order to do so effectively. Mean-field variational inference can be adapted for several situations, such as applied to mixtures of variational densities and consideration of dependencies between variables; these come with increased computational difficulty when performing optimization. Although this method is useful as it can express any marginal density of hidden variables, it is not able to consider the correlations among latent variables [51]. This can result in certain limitations, especially when considering highly correlated distributions. An example is when considering this approximation for a Gaussian distribution of two-dimensions, as described by Blei et al. [51]. The target distribution is highly elongated due to the correlation between the Gaussians; however, the optimal mean-field approximation is only centralized around the mean of the two distributions and therefore under represents the target density [51].

There are many other VB approaches which have been employed in topic modeling, however we have chosen to focus on those which were generalized for a variety of methods or were specifically mentioned in the reviewed publication. Both [51] and [53] offer significant insight into the design and optimization of these processes, along with recommendations for which methods to utilize.

5.2. K-selection methods

As previously discussed, many topic modeling methods share the same flaw of assuming the ideal topic count to be a given value, thus creating the need for optimal topic number to be determined external or through iterative processes in order to achieve meaningful results. There have been several approaches to overcoming this obstacle by model evaluation; these methods assume that the optimal number of topics will produce the best fit to the data by applying topic modeling to a subset of training data and iterating over different k values.

5.2.1. Perplexity

A recent work by Zhai et al. proposes a perplexity-based approach for evaluating topic models where the "rate of perplexity change" (RPC) is calculated over iterations, and the turning point of this value is used to select for the best fitting number of topics [27]. For their application of this method against the previous perplexity-based approach, the data was randomly divided into m subsets ($S_1, S_2, \ldots, S_m$) and models were constructed utilizing m-fold cross validation for a topic range $t = 1$ to $t = r$. The average perplexities from m testing sets for each topic value t are represented as $P_1, P_2, \ldots P_r$. Therefore, the RPC for a topic count t_i, given that i is greater than or equal to 1 and less than r can be calculated by:

$$RPC\ (i) = \left| \frac{P_i - P_{i-1}}{t_i - t_{i-1}} \right| \tag{10}$$

Zhao et al. found that when applying both approaches to a dataset of 119 *Salmonella* strains, the RPC-based approach demonstrated more exactness by selecting the same optimal topic number in 80% of its replicate models, compared to the 46% majority selected by perplexity alone [27]. Additionally, when calculating the mean entropy of the two methods the RPC approach scored significantly lower than perplexity, 1.0 to 1.853 respectively. When K-means clustering was used to determine the accuracy of the topic models, both methods were found to have selected topics which resulted in the highest purity, 0.96; however, RPC had chosen the lower topic number which would result faster running times and therefore was considered to be more efficient [27]. Although perplexity alone has generally been the standard method for determining topic numbers iteratively and comparing models, the RPC-based method demonstrated considerable potential as an evaluative tool.

The perplexity method on it on is prone to instability and can vary greatly between iterations even over the same subset of data, therefore is not the ideal approach for determining k for a dataset; however, in many cases it is the best method available to researchers and is still frequently employed. Although the RPC method described by Zhao et al. demonstrated significant improvement over the previous approach, it still requires iterative topic model generation in order to determine optimal k values [27].

5.2.2. Hierarchical Dirichlet process

As previously stated, the HDP model is a viable method of determining the number of topics automatically [43]. It is commonly used in combination with LDA or mixture models but has been applied to many other methods as well [53,55]. For more in-depth discussion of HDP, please refer to the second paragraph of Section 4.5.

6. Conclusions

Latent Dirichlet Allocation (LDA) is the most frequently utilized method for topic modeling, however it is far from the only one. Although LDA can be well-suited for general topic modeling tasks using a variety of data, it is not capable of modeling more advanced data relationships and performs poorly when documents are not of a sufficient length. This is a limitation for many research applications because complex relationships with time and between topics are present in all types of data. Ignoring the existence of these metrics causes the real-world value this analysis to be limited. There are a variety of other approaches which often are overlooked by those seeking to perform topic modeling on a dataset; in many cases the application of a better suited method can greatly improve results and is the difference between obscure and irrelevant topics and meaningful data analysis. Yet despite the wealth of approaches available, we are unaware of any comprehensive reviews of current topic modeling methods. Therefore, there is a great need for such a paper in order to more effectively promote the depth of work which has been done in the field.

We present in this paper an in-depth review of topic modeling methods, ranging from the time-tested and expertly developed to the new and innovative techniques. This review begins with a discussion of the algorithms and approaches upon which modern modeling methods have been built. Then we establish the terms and symbols which are used throughout the paper to describe the different topic modeling methods. The review of methods has been separated into sections which allow for easy identification of each method's use and its special characteristics. We also briefly discuss the techniques of optimizing these algorithms to best reflect the dataset, which is incredibly important to producing meaningful results when using probabilistic modeling approaches. Through this review, we encourage more diversified approaches when implementing topic modeling for research purposes. The classification of methods in our review is flexible and allows for additions and updates as new methods are developed. We believe this review paper is a resource to an engineer or scientist seeking to carry out such studies which can direct them to an appropriate modeling method, and act as a generalized reference for those seeking to critique studies which utilize topic modeling.

In order to better direct readers to the methods which would best suit their individual needs, we have produced a decision tree using information on each model's characteristics and performance when compared to other models (see Fig. 8). This will allow for a method to be selected based on the goals of the research that one wishes to perform and features of the data available to them. They should then review the section about the selected method to confirm that it will be a good match for what they wish to achieve. This is not a detailed guide, and considerations such as the degree of topic relationship complexity and amount of optimization needed is described in more detail throughout the paper and in the original method publications.

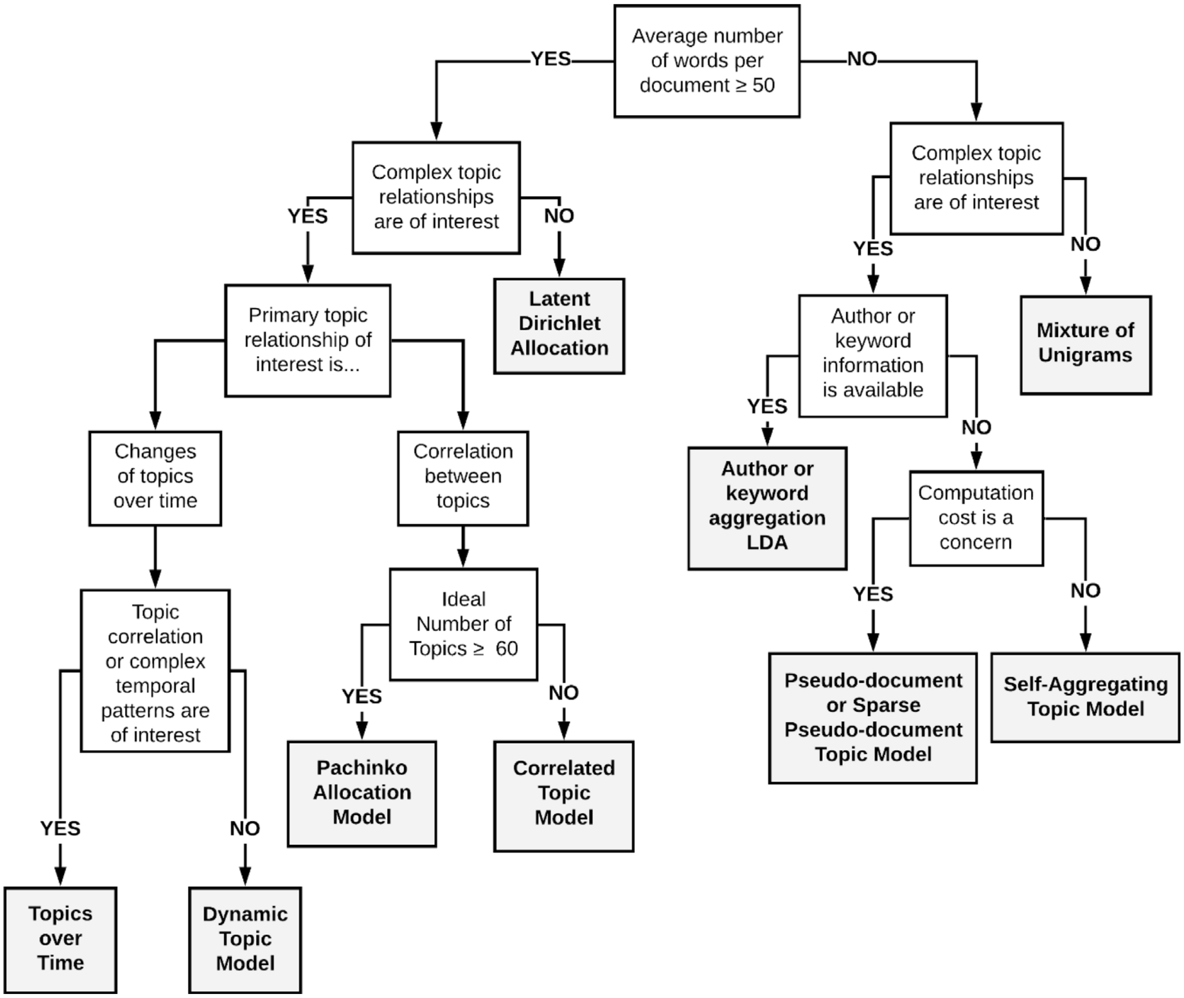
Average number of words per document ≥ 50
YES
NO
Complex topic relationships are of interest
YES
NO
Primary topic relationship of interest is...
Latent Dirichlet Allocation
Changes of topics over time
Correlation between topics
Topic correlation or complex temporal patterns are of interest
YES
NO
Topics over Time
Dynamic Topic Model
Ideal Number of Topics ≥ 60
YES
NO
Pachinko Allocation Model
Correlated Topic Model
Complex topic relationships are of interest
YES
NO
Mixture of Unigrams
Author or keyword information is available
YES
NO
Author or keyword aggregation LDA
Computation cost is a concern
YES
NO
Pseudo-document or Sparse Pseudo-document Topic Model
Self-Aggregating Topic Model

Fig. 8. Decision tree for selecting a topic modeling method.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] D.M. Blei, Probabilistic topic models, Commun. ACM 55 (4) (2012) 77.

[2] L. Liu, L. Tang, W. Dong, S. Yao, W. Zhou, An overview of topic modeling and its current applications in bioinformatics, Springerplus 5 (1) (2016).

[3] L. Hong, B.D. Davison, Empirical study of topic modeling in Twitter, in: Proceedings of the First Workshop on Social Media Analytics, SOMA '10, Washington D.C. District of Columbia, 2010, pp. 80–88.

[4] Y. Girdhar, P. Giguère, G. Dudek, Autonomous adaptive underwater exploration using online topic modeling, in: J.P. Desai, G. Dudek, O. Khatib, V. Kumar (Eds.), Experimental Robotics: The 13th International Symposium on Experimental Robotics, Springer International Publishing, Heidelberg, 2013, pp. 789–802.

[5] A. Agrawal, W. Fu, T. Menzies, What is wrong with topic modeling? And how to fix it using search-based software engineering, Inf. Softw. Technol. 98 (2018) 74–88.

[6] J.D. Lafferty, D.M. Blei, Correlated topic models, in: Y. Weiss, B. Schölkopf, J.C. Platt (Eds.), Advances in Neural Information Processing Systems, Vol. 18, MIT Press, 2006, pp. 147–154.

[7] X. Yan, J. Guo, Y. Lan, X. Cheng, A biterm topic model for short texts, in: Proceedings of the 22nd International Conference on World Wide Web, Rio de Janeiro, Brazil, 2013, pp. 1445–1455.

[8] D.M. Blei, J.D. Lafferty, Dynamic topic models, in: Proceedings of the 23rd International Conference on Machine Learning, Pittsburgh, PA, USA, 2006, pp. 113–120.

[9] W. Li, A. McCallum, Pachinko allocation: DAG-structured mixture models of topic correlations, in: Proceedings of the 23rd International Conference on Machine Learning, Pittsburgh, PA, USA, 2006, pp. 577–584.

[10] M. Steyvers, T. Griffiths, Probabilistic topic models, in: T.K. Landauer (Ed.), Handbook of Latent Semantic Analysis, first ed., Routledge, New York, NY, USA, 2011, pp. 427–440.

[11] D.M. Blei, A.Y. Ng, M.I. Jordan, Latent Dirichlet allocation, J. Mach. Learn. Res. 3 (2003) (2003) 993–1022.

[12] G. Salton, Some research problems in automatic information retrieval, in: Proceedings of the 6th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, New York, NY, USA, 1983, 252263.

[13] S. Deerwester, S.T. Dumais, G.W. Furnas, T.K. Landauer, R. Harshman, Indexing by latent semantic analysis, J. Am. Soc. Inf. Sci. 41 (6) (1990) 391–407.

[14] T. Hofmann, Probabilistic latent semantic indexing, ACM SIGIR Forum 51 (2) (2017) 8.

[15] Z. Zhou, J. Zhou, L. Zhang, Demand-adaptive clothing image retrieval using hybrid topic model, in: Proceedings of the 2016 ACM on Multimedia Conference, MM '16, Amsterdam, The Netherlands, 2016, pp. 496–500.

[16] T. Hospedales, S. Gong, T. Xiang, Video behaviour mining using a dynamic topic model, Int. J. Comput. Vis. 98 (3) (2012) 303–323.

[17] Y. Ju, B. Adams, K. Janowicz, Y. Hu, B. Yan, G. McKenzie, Things and strings: Improving place name disambiguation from short texts by combining entity co-occurrence with topic modeling, in: E. Blomqvist, P. Ciancarini, F. Poggi, F. Vitali (Eds.), Knowledge Engineering and Knowledge Management, Vol. 10024, Springer International Publishing, Cham, 2016, pp. 353–367.

[18] Eason, Kumar, Valuation of text mining techniques using twitter data for hurricane disaster resilience, AGU Fall Meeting Abstracts (2019).

[19] Dahal, Kumar, Li, Topic modeling and sentiment analysis of global climate change tweets, Social Network Analysis and Mining 9 (1) (2019).

[20] Vayansky, Kumar, Li, An evaluation of geotagged twitter data during hurricane irma using sentiment analysis and topic modeling for disaster

resilience, in: 2019 IEEE International Symposium on Technology and Society (ISTAS), 2019.

[21] Singleton, Kumar, Li, Twitter analytics-based assessment: are the United States coastal regions prepared for climate change ƒ, in: 2018 IEEE International Symposium on Technology and Society (ISTAS), 2018.

[22] Singleton, Kumar, Twitter analytics: are the US coastal regions prepared for climate change in 2017, AGU Fall Meeting Abstracts (2017).

[23] Harvey, Kumar, Bao, Machine learning-based models for assessing impacts before, during and after hurricane florence, in: 2019 IEEE Symposium Series on Computational Intelligence (SSCI), 2019.

[24] T.L. Griffiths, M. Steyvers, Finding scientific topics, Proc. Natl. Acad. Sci. 101 (Suppl. 1) (2004) 5228–5235.

[25] I. Porteous, D. Newman, A. Ihler, A. Asuncion, P. Smyth, M. Welling, Fast collapsed gibbs sampling for latent dirichlet allocation, in: Proceeding of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 08, Las Vegas, Nevada, USA, 2008, p. 569.

[26] K. Vorontsov, A. Potapenko, Tutorial on probabilistic topic modeling: Additive regularization for stochastic matrix factorization, in: D.I. Ignatov, M.Y. Khachay, A. Panchenko, N. Konstantinova, R.E. Yavorsky (Eds.), Analysis of Images, Social Networks and Texts, Vol. 436, Springer International Publishing, Cham, 2014, pp. 29–46.

[27] W. Zhao, et al., A heuristic approach to determine an appropriate number of topics in topic modeling, BMC Bioinformatics 16 (Suppl. 13) (2015) S8.

[28] J.-T. Chien, C.-H. Lee, Z.-H. Tan, Latent Dirichlet mixture model, Neurocomputing 278 (2018) 12–22.

[29] D. Agarwal, B.-C. Chen, fLDA: matrix factorization through latent dirichlet allocation, in: Proceedings of the Third ACM International Conference on Web Search and Data Mining, New York, NY, USA, 2010, pp. 91–100.

[30] D.M. Blei, T.L. Griffiths, M.I. Jordan, The nested chinese restaurant process and bayesian nonparametric inference of topic hierarchies, J. ACM 57 (2) (2010) 1–30.

[31] J. Cao, T. Xia, J. Li, Y. Zhang, S. Tang, A density-based method for adaptive LDA model selection, Neurocomputing 72 (7–9) (2009) 1775–1781.

[32] D.M. Blei, J.D. Lafferty, Chapter 4: Topic models, in: A. Sricastava, M. Sahami (Eds.), Text Mining: Classification, Clustering, and Applications, CRC Press, Boca Raton, FL, USA, 2009, pp. 71–89.

[33] U. Nodelman, C.R. Shelton, D. Koller, Continuous time bayesian networks, in: Proceedings of the Eighteenth Conference on Uncertainty in Artificial Intelligence, Alberta, Canada, 2002, pp. 378–387.

[34] X. Wang, A. McCallum, Topics over time: a non-Markov continuous-time model of topical trends, in: Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Philadelphia, PA, USA, 2006, pp. 424–433.

[35] K. Nigam, A. Mccallum, S. Thrun, T. Mitchell, Text classification from labeled and unlabeled documents using EM, Mach. Learn. 39 (2/3) (2000) 103–134.

[36] X. Quan, C. Kit, Y. Ge, S.J. Pan, Short and sparse text topic modeling via self-aggregation, in: IJCAI, 2015.

[37] Y. Zuo, et al., Topic modeling of short texts: A pseudo-document view, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD ’16, San Francisco, California, USA, 2016, pp. 2105–2114.

[38] T. Lin, W. Tian, Q. Mei, H. Cheng, The dual-sparse topic model: mining focused topics and focused terms in short text, in: Proceedings of the 23rd International Conference on World Wide Web, WWW ’14, Seoul, Korea, 2014, pp. 539–550.

[39] D.D. Lee, H.S. Seung, Learning the parts of objects by non-negative matrix factorization, Nature 401 (6755) (1999) 788–791.

[40] D. Kuang, J. Choo, H. Park, Nonnegative matrix factorization for interactive topic modeling and document clustering, in: M.E. Celebi (Ed.), Partitional Clustering Algorithms, Springer International Publishing, Cham, 2015, pp. 215–243.

[41] M. Belford, B. Mac Namee, D. Greene, Stability of topic modeling via matrix factorization, Expert Syst. Appl. 91 (2018) 159–169.

[42] C. Boutsidis, E. Gallopoulos, SVD based initialization: A head start for nonnegative matrix factorization, Pattern Recognit. 41 (4) (2008) 1350–1362.

[43] Y.W. Teh, M.I. Jordan, M.J. Beal, D.M. Blei, Sharing clusters among related groups: Hierarchical Dirichlet processes, in: L.K. Saul, Y. Weiss, L. Bottou (Eds.), Advances in Neural Information Processing Systems, Vol. 17, MIT Press, 2005, pp. 1385–1392.

[44] C.P. George, D.Z. Wang, J.N. Wilson, L.M. Epstein, P. Garland, A. Suh, A machine learning based topic exploration and categorization on surveys, in: 2012 11th International Conference on Machine Learning and Applications, Boca Raton, FL, USA, 2012, pp. 7–12.

[45] Y.W. Teh, K. Kurihara, M. Welling, Collapsed variational inference for HDP, in: Advances in Neural Information Processing Systems, 2008, pp. 1481–1488.

[46] D.J.C. MacKay, L.C.B. Peto, A hierarchical Dirichlet language model, Nat. Lang. Eng. 1 (03) (1995).

[47] H.M. Wallach, Topic modeling: beyond bag-of-words, in: Proceedings of the 23rd International Conference on Machine Learning, Pittsburgh, PA, USA, 2006, pp. 977–984.

[48] B. Walsh, Lecture Notes for EEB 581: Markov Chain Monte Carlo and Gibbs Sampling, University of Arizona, 2004.

[49] C.J. Geyer, Practical Markov chain Monte Carlo, Statist. Sci. 7 (4) (1992) 473–483.

[50] A. Bhadury, J. Chen, J. Zhu, S. Liu, Scaling up dynamic topic models, in: Proceedings of the 25th International Conference on World Wide Web, Montreal, Quebec, Canada, 2016, pp. 381–390.

[51] D.M. Blei, A. Kucukelbir, J.D. McAuliffe, Variational inference: A review for statisticians, J. Amer. Statist. Assoc. 112 (518) (2017) 859–877.

[52] Y.W. Teh, D. Newman, M. Welling, A collapsed variational Bayesian inference algorithm for latent Dirichlet allocation, in: B. Schölkopf, J.C. Platt, T. Hoffman (Eds.), Advances in Neural Information Processing Systems, Vol. 19, MIT Press, 2007, pp. 1353–1360.

[53] M.J. Beal, Variational Algorithms for Approximate Bayesian Inference (Doctorate thesis), University of London, London, England, 2003.

[54] J. Yin, Y. Zhang, L. Gao, Accelerating distributed Expectation–Maximization algorithms with frequent updates, J. Parallel Distrib. Comput. 111 (2018) 65–75.

[55] T.S. Jaakkola, M.I. Jordan, Improving the mean field approximation via the use of mixture distributions, in: M.I. Jordan (Ed.), Learning in Graphical Models, Springer Netherlands, Dordrecht, 1998, pp. 163–173.