A Survey of Topic Model Inference Techniques

Geoffrey Mariga Wambugu
Department of Information Technology
Murang'a University of Technology
Murang'a, Kenya
Email: gmariga [AT] mut.ac.ke

George Okeyo, and Stephen Kimani
Department of Computing
Jomo Kenyatta University of Agriculture and Technology
Nairobi, Kenya

Abstract— **Latent Dirichlet Allocation (LDA) is a probabilistic topic model that aims at organizing, visualizing, summarizing, searching, predicting and understanding the content of any given text data. The model enables users to discover themes in text, annotate, organize and summarize documents. LDA inference involves estimating the parameters and posterior distribution of a formulated mathematical relationship. This paper investigates topic modeling literature based on LDA and presents discoveries and state of the art in the topic. Presented also are challenges and popular tools. In conclusion, the paper identifies Gibbs sampling as a popular inference mechanism and notes that the method is limited for application in big data settings.**

Keywords-Topic Modelling; Latent Dirichlet Allocation; Sampling; Inference Techniques.

I. INTRODUCTION

The world is witnessing increasing amounts of vast digitized data sets, such as collections of video, images and text documents. Examples range from large collections of online books, to large collection of video, images and text at social media sites. Consequently, it becomes more difficult to find and discover what we are looking for. We therefore need new computational tools to help organize, search and understand these vast amounts of data. These data sets present major opportunities for machine learning, such as the ability to explore richer and more expressive models than previously possible, and offer new and exciting domains for the application of learning algorithms (Blei, 2012, Newman et al 2009).

A probabilistic topic model, simply topic model, forms a subclass of a Bayesian network (Welling, et. al. 2012) that aims at organizing, visualizing, summarizing, searching, predicting and understanding the content of any given text data by enabling model users to discover themes in text, annotate documents, organize and summarize text documents. The models, which are unsupervised generative algorithms, usually describe document content as a two-step generation process, that is, documents are observed as mixtures of latent topics, while topics are probability distributions over vocabulary words (Hu, 2009, Vulić, et al 2015).

Topic models uses probability distributions to describe data emanating from a system. These probability distributions requires the definition of parameters. In a Bayesian context, these parameters are also distributed according to some distribution as opposed to the frequentist approach of point estimates where the parameters are estimated as single values. The parameters that describe the distribution of other parameters are called hyperparameters ie parameters of parameters. A topic model has three main objectives: (1) learning distributions on words called "topics" shared by documents; (2) Learning a distribution on topics for each document and (3) Assigning every word in a document to a topic. In topic modeling, none of these objectives are known in advance and must be learned. Learning involves defining a model, and a learning algorithm. Each document is treated as a "bag of words".

While originally, the models were designed to analyse and classify large text document collections, they have nowadays been extended to model data from other fields, including computer vision. In particular, applications to computer vision are extensive for example (Li, et al (2009), Wang, C. et al (2009), Wang, X. et al, 2007). Some of the examples where researchers have used topic models in computer vision problems include sorting multiple images into scene-level classes, annotating images with words, and segmenting and labelling objects within images. Statistical topic models have also been extended to analysis of video (Wang, X. (2007), Wang, X. (2009)). Other fields where topic models have been applied are bio-informatics, finance and even social sciences (Vulić et al., 2015). In the original definition, a topic is defined to contain a cluster of words that frequently occur together. A topic model can associate words with comparable meaning and distinguish between uses of words with numerous meanings.

The main computational challenge in topic modelling is inference, namely estimating the posterior distribution over both parameters and hidden variables and ultimately estimating predictive probabilities and the evidence (Asuncion, et al, 2009, Heinrich, 2008).

II. BAYESIAN MODELLING AND TOPIC MODEL INFERENCE

Bayesian modelling describes data that one could observe from a system using mathematics of inverse probability to express all forms of uncertainty and associated noise. The models allow users to infer unknown quantities, adapt the models, make predictions and learn from data. Bayesian modelling and inference is based on Bayes' theorem which can be stated simply as follows:

$$P(hypothesis|data) = \frac{P(data|hypothesis\ )\,P(hypothesis)}{P(data)}$$

P(hypothesis|data) is called posterior, P(data|hypothesis) referred to as the Likelihood, P(hypothesis) prior and P(data) Evidence.

Bayes rule tells us how to do inference about the posterior from the evidence. Learning and prediction can be seen as a form of inference. Machine Learning seeks to learn models of data by defining a space of possible models; learning the parameters and structure of the models from data and making predictions and decisions. Our goal in Bayesian inference is to get an accurate representation of the posterior distribution, the P(hypothesis|data).

Bayesian inference is the process of fitting a probability model to a set of data and summarizing the result using a probability distribution on the parameters of the model and on unobserved quantities such as predictions for new observations. Bayesian method's central characteristic is their explicit use of probability for quantifying uncertainty in inferences based on statistical data analysis. The following three steps describe Bayesian data analysis: (1) Setting up a joint probability distribution for all observable and unobservable quantities in a problem. (2) Conditioning on observed data: calculating and interpreting the appropriate posterior distribution, the conditional probability distribution of the unobserved quantities of ultimate interest, given the observed data. (3) Evaluating the fit of the model and the implications of the resulting posterior distribution: how well does the model fit the data, are the substantive conclusions reasonable, and how sensitive are the results to the modeling assumptions in step 1? In response, one can alter or expand the model and repeat the three steps.

Bayesian modelling and inference is a popular and growing tool in statistics, machine learning and data mining. It is one of the two dominating perspectives used in probabilistic modelling and has certain interesting features for handling over-fitting, prior information and uncertainty, which can be useful in applications.

A. The Basic Inference Process

The basic inferential process illustrated on the Figure 1 below is a three step process. The first step in any inference procedure is to collect data from the system of interest. In the second step, a statistical model which usually depends on a number of unknown parameters is build. The last step involves making use of the data to infer the parameters of interest.

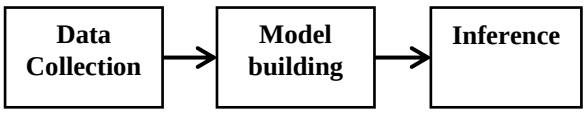

Figure 1: The Basic Inference Process.

After the model has been inferred, it can be used to make forecasts or for making decisions by taking uncertainty into account.

B. Topic Model Inference Techniques

For a fully defined topic model, we need to have the following: (1) well identified probability distributions of prior, likelihood and posterior over the problem in question; (2) A set of parameters associated with the defined model, to be estimated from data and fit into the model; (3) a notation to aid the mathematical analysis; (4) Conversion of data to digits that can be analysed using the defined probabilistic model. (5) Interpretation of the data obtained back to a human understandable form. (6) Presentation of the analytical results for utilisation by the expected audience. The last point includes visualisation which is the use of typography, animation and cinematography with modern human-computer interaction system to facilitate the comprehension of computer systems for easier and faster utility (Alice C 2014).

In probabilistic topic modeling the distributions are described as mathematical functions. Generally inference is concerned with drawing conclusions, from numerical data, about quantities that are not observed. Inferring posterior and parameter estimates involves integrations and because of the multi-parameter nature and correlation between the parameters of interest in the defined models, the complex integration leads to a challenge of solving intractable functions. Many researchers have used various mathematical approximations to deal with these intractable integrals that result to inference algorithms. These approximations have various advantages and disadvantages and are subject to scientific research in establishing optimal solutions in contextual setups.

Modern approximate posterior inference algorithms fall into two main categories: sampling approaches and optimization approaches. In sampling based approaches like the popular Markov Chain Monte Carlo (MCMC), the topic model parameters of interest are approximated through an iterative process where transitional matrices are improved to convergence. The conceptual idea of these methods is to generate independent samples that are provably distributed according to the posterior and then reason about the data of interest. This is achieved by drawing parameter values from approximate distributions and then correcting those draws to better approximate the target posterior distribution, say $p(\theta|y)$

where θ is the set of parameters to be estimated and y is the given data. The sampling is done sequentially, with the distribution of the sampled draws depending on the last value drawn; hence, the draws form a Markov chain. This is illustrated in the generic Gibbs sampler below which is a specific case of MCMC.

C. Generic Gibbs Sampler

Given a joint sample x^s of all the variables, we generate a new sample x^{s+1} by sampling each component in turn, based on the most recent values of the other variables. For example, if we have three variables, we use

- $x_1^{s+1} \sim p(x_1 | x_2^s, x_3^s)$
- $x_2^{s+1} \sim p(x_2 | x_1^{s+1}, x_3^s)$
- $x_3^{s+1} \sim p(x_3 | x_1^{s+1}, x_2^{s+1})$

x_i is not sampled since it is a visible variable and therefore is already known.

The key to the method's success, however, is not the Markov property but rather that the approximate distributions are improved at each step in the simulation, in the sense of converging to the target distribution. The Markov property is helpful in proving this convergence (Gelman et al 2014). Unlike variational inference, MCMC has asymptotic guarantees.

The second category referred to as optimization approaches are based on variational inference and also called the Variational Bayesian (VB) methods. They involve finding an approximation of the posterior within an analytically tractable family of distributions. These methods optimize the closeness of the posterior, based on the Kullback-Leibler divergence, to a simplified parametric distribution (Špeh & Rupnik, 2013). The idea is to choose a distribution within the simplified family that is closest to the true posterior.

D. Drawbacks of Markov Chain Monte Carlo

The most widely used posterior inference methods in Bayesian nonparametric models, according to Gershman & Blei (2012), are the Markov Chain Monte Carlo (MCMC). Unlike Variational Bayesian (VB) methods MCMC are guaranteed to converge to the posterior with enough samples. However the methods have two major drawbacks: (1) the samplers must be run for many iterations before convergence and (2) it is difficult to assess convergence. These drawbacks slows the models in application hence raising a practical motivation for speed enhancement in MCMC approaches inorder to enhance their usage in the era of BigData analytics where mining of numerous data streams is performed in real-time. Despite the fact that faster hardware is continually appearing in the market, increases in hardware speed are outstripped by increases in our ambitions. For example, even allowing for good advances in hardware speed, we will need supremely efficient algorithms if we are to advance the state-of-the-art in realistic systems development (Zhang, 2013). A research therefore aimed at improving the speed and efficiency of MCMC algorithms will always find its place.

E. Strategies for Accelerating Inference

A drawback with these approaches are that a large number of samples from the posterior are usually required to obtain reasonable accuracy. This is often not a problem for simpler Monte Carlo methods, where samples can be generated efficiently. However, for more complicated models it can take considerable time to generate a sample.

There are two main approaches to mitigate this problem: (i) decreasing the computational cost of generating a sample or (ii) decreasing the required number of samples by making better use of them. Another aspect for the user is the time and complexity of the implementation of an inference algorithm for a new model. It can therefore be beneficial to apply simpler methods that are quicker to implement and tune than more elaborate algorithms that may be more efficient but takes longer time to get up and running. The total time for the entire inference can therefore be shorter in the former case than in the latter, even if the advanced algorithm is more efficient.

III. The Latent Dirichlet Allocation (LDA) Model

Latent Dirichlet Allocation (LDA) (Blei et al., 2003) is one of the most classical state of the art unsupervised probabilistic topic model which can be applied to a corpus of text documents in a bag-of-words form to discover semantic structure of documents, by examining statistical word co-occurrence patterns within a corpus of training documents. LDA assumes a hypothetical generative process is responsible for creating observed set of documents. The natural statistical assumption behind LDA is to model documents as arising from multiple topics, where a topic is defined to be a distribution over a fixed vocabulary of terms. LDA assumes that, since documents in a corpus tend to be heterogeneous, K topics are associated with a collection, and that each document exhibits these topics with different proportions.

LDA has since found various applications in different fields such as text mining, computer vision, population genetics, social network analysis and bio-informatics, (AlSumait et al., 2008, Cao & Fei-Fei, 2007, Zheng et al., 2006, Zhang et al., 2007, Jiang et al. 2015, Liu et al 2016). While LDA is applicable to any corpus of grouped discrete data, for this paper we refer to the standard Natural Language Processing use case where a corpus is a collection of text documents, and the data are words. The original generative process for a document collection D under the LDA model is defined by Blei (2003) as follows:

1.For k = 1, …, K

(a) $\phi^{(k)} \sim Dirichlet\ (\beta)$

2.For each document d ϵ **D**

(a) Choose $\theta_d \sim Dirichlet\ (\alpha)$

(b) For each word w_i ϵ d:

i) $z_i \in$ Discrete(θ_d)

ii) $w_i \in$ Discrete($\phi^{(zi)}$

where K is the number of latent topics in the collection, $\phi^{(k)}$ is a discrete probability distribution over a fixed vocabulary that represents the *k*th topic distribution, θ_d is a document-specific distribution over the available topics, z_i is the topic index for word w_i, and α and β are hyperparameters for the symmetric Dirichlet distributions that the discrete distributions are drawn from.

The above described generative process results in the following joint distribution:

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

(equation 1)

What is of interest to us is the hidden variables **z**, **θ,** and ϕ. Each θ_d is a low-dimensional representation of a document in "topic"-space, each z_i represents which topic generated the word instance w_i, and each $\phi^{(k)}$ represents a K X V matrix where $\phi_{i,j} = p(w_i|z_j)$. Therefore, one of the most interesting aspects of LDA is that it can learn, in an unsupervised manner, words that we would associate with certain topics, and this is expressed through the topic distributions ϕ. Table 1 is an example of top 10 words for 3 topics that can be learned using LDA on a sample dataset. On the table 1 below, topic labels and word selection have been added manually for illustration purposes.

Table 1: Three topics learned using LDA on Sample Dataset.

"environs"	**"travel"**	**"war"**
Discharge	travel	combat
Conservational	tourism	fight
Air	Hotel	hatred
License	hostel	match
Herb	Fares	animosity
Facility	City	ethnic
Part	town	battle
Ape	Visit	war
Water	Miles	against
Location	Deal	team

IV. Inference

In topic modeling the main problem is posterior inference of hidden variables given some parameters from observed data which amounts to reversing the defined generative process and learning the posterior distributions of the latent variables in the model. In Latent Dirichlet Allocation, this involves solving the following equation:

$$p(\theta, \phi, \boldsymbol{z}|\mathbf{w}, \alpha, \beta) = \frac{p(\theta, \phi, \boldsymbol{z}, \mathbf{w}|\alpha, \beta)}{p(\mathbf{w}|\alpha, \beta)} \quad (2)$$

We note from literature that this distribution is intractable to compute. However, researchers have developed a number of approximate inference techniques that can be applied to the problem. Three of the most common are: one, variational inference which was used in the original LDA paper and further improved to Collapsed Variational Bayes (Teh & Welling, 2007). The second is Markov Chain Monte Carlo approach called Gibbs Sampling which samples from the posterior over topic assignments by repeatedly sampling the topic assignment conditioned on the data and all other topic assignments. Many researchers have improved this technique for topic modeling for example Qiu et al (2014). This paper will further explore Gibbs Sampling. Lastly we have online variational Bayes algorithm (Hoffman, et. al. 2010) which is based on online stochastic optimization with a natural gradient step.

V. Gibbs Sampling

Gibbs sampling was developed in 1876 by Josiah Willard Gibb as a means of determining the energy states of gasses at equilibrium. The method he used was modelled as a Bayesian posterior distribution in 1990 by Alan Gelfand and Adrian Smith (Bolstad, 2012). It defines a Markov Chain whose stationary distribution is the posterior of interest. Independent samples are collected from the stationary distribution to approximate the posterior. This produces a Monte Carlo estimate of an expectation using independent samples from the posterior (Heinrich, 2009).

Gibbs Sampling is one member of a family of algorithms from the Markov Chain Monte Carlo (MCMC) framework (Gilks et. al, 1999). The MCMC algorithms aim to construct a Markov chain that has the target posterior distribution as its stationary distribution. In other words, after a number of iterations of stepping through the chain, sampling from the distribution should converge to be close to sampling from the desired posterior. Gibbs Sampling is based on sampling from conditional distributions of the variables of the posterior.

For example, to sample x from the joint distribution p(x) = $p(x_1, \ldots, x_m)$, where there is no closed form solution for p(x), but a representation for the conditional distributions is available, one would perform the following using Gibbs Sampling:

1. Randomly initialize each x_i
2. For $t = 1, \ldots T$:

2.1 $x_1^{t+1} \sim p(x_1|x_2^{(t)}, x_3^{(t)}, \ldots, x_m^{(t)})$

2.2 $x_2^{t+1} \sim p(x_2|x_1^{(t+1)}, x_3^{(t)}, \ldots, x_m^{(t)})$

…

2.*m* $x_m^{t+1} \sim p(x_m|x_1^{(t)}, x_2^{(t+1)}, \ldots, x_{m-1}^{(t+1)})$

This procedure is repeated a number of times until the samples begin to converge to what would be sampled from the true distribution. Though convergence is guaranteed theoretically with Gibbs Sampling, it is not possible to know the number of iterations required to reach the stationary distribution. Therefore, diagnosing convergence is a real problem with the Gibbs Sampling approximate inference method. Many authors however note that Gibbs sampling is pretty powerful in practice and has fairly good performance. Most applications provide a calculation of the log-likelihood which is an acceptable estimation of convergence.

For LDA, we are interested in the latent document-topic portions θ_d, the topic-word distributions $\phi^{(z)}$, and the topic index assignments for each word z_i. While conditional distributions, and therefore an LDA Gibbs Sampling algorithm, can be derived for each of these latent variables, we note that both θ_d and $\phi^{(z)}$ can be calculated using just the topic index assignments z_i (i.e. z is a sufficient statistic for both these distributions). Therefore, a simpler algorithm can be used if we integrate out the multinomial parameters and simply sample z_i. This is called a collapsed Gibbs sampler.

The collapsed Gibbs sampler for LDA needs to compute the probability of a topic z being assigned to a word w_i, given all other topic assignments to all other words. In a mathematical statement, we are therefore interested in computing the following posterior up to a constant:

$$p(z_i|\boldsymbol{z}_{-i}, \alpha, \beta, w) \tag{3}$$

where z_{-i} means all topic allocations *except* for z_i. To begin, the rules of conditional probability tell us that:

$$p(z_i|\boldsymbol{z}_{-i}, \alpha, \beta, w) = \frac{p(z_i, \boldsymbol{z}_{-i}, \mathbf{w}|\alpha, \beta)}{p(\boldsymbol{z}_{-i}, \mathbf{w}|\alpha, \beta)} \propto p(z_i, \boldsymbol{z}_{-i}, \mathbf{w}|\alpha, \beta) = p(\boldsymbol{z}, \mathbf{w}|\alpha, \beta) \tag{4}$$

We then have:

$$p(\mathbf{w}, \mathbf{z}|\alpha, \beta) = \iint p(\mathbf{z}, \mathbf{w}, \theta, \phi|\mathbf{w}, \alpha, \beta)\, d\theta d\phi \tag{5}$$

Following the LDA model defined in equation (1), we can expand the above equation to get:

$$p(\mathbf{w}, \mathbf{z}|\alpha, \beta) = \iint p(\phi|\beta)\, p(\theta|\alpha) p(z|\theta) p(w|\phi_z) d\theta d\phi \tag{6}$$

Then, we group the terms that have dependent variables:

$$p(\mathbf{w}, \mathbf{z}|\alpha, \beta) = \int p(z|\theta) p(\theta|\alpha) d\theta \int p(\phi|\beta) p(w|\phi_z) d\phi \tag{7}$$

Both terms are multinomials with Dirichlet priors. Because the Dirichlet distribution is conjugate to the multinomial distribution, our work is vastly simplified; multiplying the two results in a Dirichlet distribution with an adjusted parameter. Beginning with the first term, we have:

$$\int p(z|\theta) p(\theta|\alpha) d\theta = \int \prod_i \theta_{d,z_i} \frac{1}{B(\alpha)} \prod_k \theta_{d,k}^{\alpha_k} d\theta_d = \frac{1}{B(\alpha)} \int \prod_k \theta_{d,k}^{n_{d,k}+\alpha_k} d\theta_d = \frac{B(n_{d,\cdot}+\alpha)}{B(\alpha)} \tag{8}$$

where $n_{d,k}$ is the number of times words in document d are assigned to topic k, a . indicates summing over that index, and B(α) is the multinomial beta function, $B(\alpha) = \frac{\prod_k \Gamma(\alpha_k)}{\Gamma(\sum_k \alpha_k)}$. Similarly, for the second term (calculating the likelihood of words given certain topic assignments):

$$\int p(\boldsymbol{w}|\phi_z) p(\phi|\beta) d\phi = \int \prod_d \prod_i \phi_{z_{d,i} w_{d,i}} \prod_k \frac{1}{B(\beta)} \prod_w \phi_{k,w}^{\beta_w} d\phi_k$$

$$= \prod_k \frac{1}{B(\beta)} \int \prod_w \phi_{k,w}^{\beta_w + n_{k,w}} d\phi_k = \prod_k \frac{B(n_{k,\cdot}+\beta)}{B(\beta)} \tag{9}$$

Combining equations (8) and (9), the expanded joint distribution is then:

$$p(\mathbf{w}, \mathbf{z}|\alpha, \beta) = \prod_d \frac{B(n_{d,\cdot}+\alpha)}{B(\alpha)} \prod_k \frac{B(n_{k,\cdot}+\beta)}{B(\beta)} \tag{10}$$

The Gibbs sampling equation for LDA can then be derived using the chain rule (where we leave the hyperparameters α and β out for clarity). Note that the superscript $(^{-i})$ signifies leaving the i^{th} token out of the calculation:

$$p(z_i|z^{(-i)}, w) = \frac{p(\mathbf{w}, \mathbf{z})}{p(\mathbf{w}, \boldsymbol{z}^{(-i)})} = \frac{p(\mathbf{z})}{p(\boldsymbol{z}^{(-i)})} \cdot \frac{p(\mathbf{w}|\mathbf{z})}{p(\boldsymbol{w}^{(-i)}|\boldsymbol{z}^{(-i)}) p(w_i}$$

$$\propto \prod_d \frac{B(n_{d,\cdot}+\alpha)}{B(n_{d,\cdot}^{(-i)}+\alpha)} \prod_k \frac{B(n_{k,\cdot}+\beta)}{B(n_{k,\cdot}^{(-i)}+\beta)}$$

VI. Existing Topic Modelling Tools

There are various text analysis software tools and frameworks available for use and which work upon different methods of topic modelling. Each one requires different skills for use. Some of the more popular software tools for Topic Modelling are discussed in this section. Table 2 below shows a list of various tools that implements the latent Dirichlet allocation model.

Table 2: Tools implementing latent Dirichlet allocation.

Tool	Description
LDA-C	Developed by Blei and is available from <http://www. cs.princeton.edu/~blei/lda-c/>. The software is written in C and provides implementation of Variational inference of LDA. It was one of the earlier software programs available so does not have as many options to tailor the analysis.
GibbsLDA++	This software was developed by Xuan-Hieu Phan and Cam-Tu Nguyen and is available online from <http://gibbslda.sourceforge. net/>. It is

	implemented in C/C++ and uses Gibbs sampling inference for LDA model fitting. According to ____ it has little room for customization.
Gensim	Developed by Radimˇ Rehuˇrek in python programming language. The software is available from <http://radi mrehurek.com/gensim/> and provides implementation based on the papers "Online learning for Latent Dirichlet Allocation" and "Online variational inference for the Hierarchical Dirichlet Process". Python based and allows for more customisation. There are support materials on how to run a topic model analysis. The preferred file formats are plain text files.
Lda-1.0.5	Implements LDA using collapsed Gibbs sampling in python. It is fast and is tested on Linux, OS X, and Windows. The interface follows conventions found in scikit-learn.
Matlab topic modelling toolbox	This is a Matlab implementation developed by Mark Steyvers. It is available online from <http://psiexp.ss.uci.edu/research/program s_ data/toolbox.htm> and provides implementation of Gibbs sampling version of LDA based on the paper "Finding Scientific topics" by
Mallet topic modelling	Mallet is a JAVA implementation of LDA developed by Andrew McCallum and available from <http://mallet. cs.umass.edu/ topics.php>. This software provides an implementation of "Probabilistic Topic Models" by Steyvers and Griffiths (2007). Further to the original implementation of Mallet, there is a package to be used from R and a complete tutorial on its usage from R is provided at Topic modelling in R
R topic modeling packages.	Mallet, Topic Models, and LDA are three packages capable of doing topic modelling analysis in R. The mallet package in R is based on the Mallet software package but rather than being capable of running a large selection of text analysis algorithms it can only do topic modelling analysis. It provides an interface to the Java implementation of latent Dirichlet allocation
Stanford Topic Modelling Toolbox	The Stanford Topic Modeling Toolbox was written at the Stanford NLP group by: Daniel Ramage and Evan Rosen, first released in September 2009. It is available online at <http://nlp.stanford.edu/software /tmt/tmt-0.4/>
GraphLab Create	GraphLab Create is a proprietary Python library, backed by a C++ engine, for quickly building large-scale, high-performance data products. In text analytics GraphLab Create can be used for automated text analytics including detecting user sentiment regarding product reviews, creating features for use in other machine learning models and understanding large collections of documents. GraphLab implements Collapsed Gibbs Sampling in LDA inference.
SCIKIT learn	Scikit-learn is a free Python programming machine learning library which features classification, regression and clustering algorithms including support vector machines, random forests, gradient boosting, k-means and DBSCAN, and is designed to interoperate with NumPy and SciPy. Scikit-learn implements Online Latent Dirichlet Allocation with variational inference.

VII. CONCLUSION

Gibbs sampling is a popular method applied to approximate intractable integrals in probabilistic generative models such as latent Dirichlet allocation. This method has however the crucial drawback of high computational complexity due to its iterative nature, which limits its application in big data environments. This paper identifies this as a gap in BigData analytics that researchers in the field can focus on.

ACKNOWLEDGMENT

The authors gratefully acknowledge that the funding for this work was provided by the National Commission for Science, Technology and Innovation, Kenya Government, reference: NACOSTI/RCD/ST&I/7th CALL/PhD/048

REFERENCES

[1] Asuncion, A., Welling, M., Smyth, P., & Teh, Y. W. (2009, June). On smoothing and inference for topic

models. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence* (pp. 27-34). AUAI Press.

[2] Blei, D. M. 2012. "Probabilistic Topic Models," *Communications of the ACM* (55:4), pp. 77-84.

[3] Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent Dirichlet allocation. *the Journal of machine Learning research*, *3*, 993-1022.

[4] Gelman, A., Carlin, J. B., Stern, H. S., & Rubin, D. B. (2014). *Bayesian data analysis* (Vol. 2). Boca Raton, FL, USA: Chapman & Hall/CRC.

[5] Heinrich, G. (2008). Parameter estimation for text analysis. *University of Leipzig, Tech. Rep*.

[6] Hoffman, M., Bach, F. R., & Blei, D. M. (2010). Online learning for latent Dirichlet allocation. In *advances in neural information processing systems* (pp. 856-864).

[7] Hu, D. J. (2009). Latent Dirichlet allocation for text, images, and music. *University of California, San Diego*.

[8] Li J, Socher R, Fei-Fei L. Towards total scene understanding: Classification, annotation and segmentation in an automatic framework; Proc. IEEE Conf. Computer Vision and Pattern Recognition; 2009.

[9] Liu, L., Tang, L., Dong, W., Yao, S., & Zhou, W. (2016). An overview of topic modeling and its current applications in bioinformatics. SpringerPlus, 5(1), 1608.

[10] Newman, D., Asuncion, A., Smyth, P., & Welling, M. (2009). Distributed algorithms for topic models. *The Journal of Machine Learning Research*, *10*, 1801-1828.

[11] Qiu, Z., Wu, B., Wang, B., Shi, C., & Yu, L. (2014, August). Collapsed Gibbs sampling for latent Dirichlet allocation on spark. In *Proceedings of the 3rd International Conference on Big Data, Streams and Heterogeneous Source Mining: Algorithms, Systems, Programming Models and Applications-Volume 36* (pp. 17-28). JMLR. org.

[12] Špeh, J., Muhič, A., & Rupnik, J. (2013). Algorithms of the LDA model [REPORT]. *arXiv preprint arXiv:1307.0317*.

[13] Teh, Y. W., Newman, D., & Welling, M. (2007). A collapsed variational Bayesian inference algorithm for latent Dirichlet allocation. In *Advances in neural information processing systems* (pp. 1353-1360).

[14] Vulić, I., De Smet, W., Tang, J., & Moens, M. F. (2015). Probabilistic topic modeling in multilingual settings: An overview of its methodology and applications. *Information Processing & Management*, *51*(1), 111-147.

[15] Wang C, Blei D, Fei-Fei L. Simultaneous image classification and annotation; Proc. IEEE Conf. Computer Vision and Pattern Recognition; 2009.

[16] Wang X, Grimson E. Spatial latent Dirichlet allocation; Proc. Neural Information Processing Systems (NIPS'07); 2007.

[17] Wang X, Ma X, Grimson E. Unsupervised activity perception by hierarchical Bayesian models; Proc. IEEE Conf. Computer Vision and Pattern Recognition; 2007. Jun, pp. 1–8.

[18] Welling, M., Teh, Y. W., & Kappen, H. (2012). Hybrid variational/Gibbs collapsed inference in topic models. *arXiv preprint arXiv:1206.3297*.

AUTHORS:

Geoffrey Mariga Wambugu obtained his BSc degree in Mathematics and Computer Science from Jomo Kenyatta University of Agriculture and Technology in 2000, and his MSc Degree in Information Systems from the University of Nairobi in 2012. He is currently pursuing his PhD degree in Information Technology at Jomo Kenyatta University of Agriculture and Technology. His research interest is in probabilistic machine learning, Big text data analytics.

George Onyango Okeyo is the Chairman of Computing Department at Jomo Kenyatta University of Agriculture and Technology. He obtained his BSc Mathematics and Computer Science from Jomo Kenyatta University of Agriculture and Technology in 2001 his MSc degree in Information Systems from the University of Nairobi in 2007, and his PhD degree in Computer Science from the University of Ulcer in 2013 and. His research interests are in intelligent agents, smart homes, ambient assisted living, Semantic Web, and knowledge representation and reasoning.

Stephen Kimani is the Director of School of Computing and Information Technology at Jomo Kenyatta University of Agriculture and Technology. He obtained his BSc Mathematics and Computer Science from Jomo Kenyatta University of Agriculture and Technology in 1995 his MSc degree in Advanced Computing (HCI & Multimedia) University of Bristol, UK in 1998 and his PhD degree in PhD in Computer Engineering, DIAG, Sapienza University of Rome, Italy in 2004. His research interests are in Human-Computer Interaction (HCI) as it relates to areas such as: User Interfaces, Usability, Accessibility, Evaluation, Mobile Computing, and Information Visualization.