Quadratic Approach for Fast Topic Selection in Modelling Big Text Analytics

[1]Geoffrey Wambugu, [2]George Onyango, [3]Stephen Kimani,
[1]*Department of Information Technology, Murang'a University of Technology, gmariga@mut.ac.ke*
[2]*Department of Computing, Jomo Kenyatta University of Agriculture and Technology, gokeyo@jkuat.ac.ke*
[3]*Department of Computing, Jomo Kenyatta University of Agriculture and Technology, Juja, Kenya, skimani@icsit.jkuat.ac.ke*

Abstract

One challenging issue in application of Latent Dirichlet Allocation (LDA) is to select the optimal number of topics which must depend on both the corpus itself and user modeling goals. This paper presents a topic selection method which models the minimum perplexity against number of topics for any given dataset. The research set up scikit-learn and graphlab on jupyter notebook in the google cloud compute engine's custom machine and then developed python code to manipulate selected existing datasets. Results indicate that the graph of perplexity against number of topics (K) has a strong quadratic behaviour around a turning point and opens upwards. This means that the graph has a minimum perplexity point that optimizes K. The paper presents a model of the optimum K in an identified interval and guides the calculation of this value of K within three iterations using quadratic approach and differential calculus. This improves inferential speed of number of topics and hyper parameter alpha thereby enhancing LDA application in big data.

Keywords: *Latent Dirichlet Allocation, Topic Modeling, Topics, Parameters*

1. Introduction

Progress in information technology from large mainframes to PCs to mobile devices to cloud has brought an information overflow, with transformative societal implications that affect all aspects of human life. A considerable and possibly the most significant portion of this information is in the form of text data, such as books, news articles, microblogs and instant messages. These vast quantities of text data can only be accessed and utilized using information technology devices, but the automated processing of text is only possible using technology specialized for human language. Text analytics in a broad sense refers to technology that allows the utilization of large quantities of text data among them topic models.

Topic Models are a class of Bayesian algorithms that can be used to analyse text documents to a lower dimension. A lot of research in the area of Topic Modeling has been carried out, pioneered by Hofmann, (1999). David M Blei, Ng, & Jordan, (2003) developed Latent Dirichlet Allocation (LDA), a generative topic modeling technique referred to by Roberts et. al. (2015) as a prominent example in the field of text data analysis.

Latent Dirichlet Allocation(LDA), was originally introduced by Blei et al. (2003) and has been widely researched and used in many applications such as text mining, information retrieval, and computer vision. In order to apply LDA, we need to specify alpha and beta dirichlet hyperparameters and number of topics K. The performance of many machine learning methods depends critically on hyperparameter settings (Hutter et al, 2014). This means that the predictive performance of LDA can be affected by the choice of hyperparameters. Often users must try different values of hyperparameters and select the best model. An alternative to setting the hyperparameters in advance is to infer them from the data. As with the other parameters of the model, the hyperparameters can be treated as random variables and have their posterior inferred (Wallach, 2008). Rather than full posterior inference, however, a more common practice is to perform MAP estimation with optimization algorithms, which has been empirically shown to be a competitive approach to sampling the hyperparameter values (Wallach et al., 2009a).

Selecting the optimal number of topics is one of the challenging issues in the application of LDA (Wang, et al, 2014 & Zhao, et al 2015). Several approaches exist, but ultimately, the appropriate number of topics must depend on both the corpus itself and user modeling goals. Collapsed Gibbs Sampling (CGS) offers a popular technique to infer the hyperparameters from data and guarantees convergence to actual values. However CGS is iterative and takes a lot of time to converge. Researchers have opted to set the number of iterations before hand since it is also difficult to access the point of convergence.

This study presents a topic selection method which obtains optimum value of K within three iterations of CGS thereby improving on number of topics inferential speed. The study models the minimum perplexity against number of topics for any given dataset using the CGS topic model application in graphlab create.

1.1 Approaches to Setting Number of Topics

The first is to set manually via "trial and error". If there is a human in the loop, it may be simplest to try multiple values of K within some reasonable range (e.g., K ϵ {5; 10; 25; 50; 100; 200}). The user can then quickly scan the learned topics associated with each value of T and select the value which seems most appropriate to the corpus.

The second is to use domain knowledge. If the topics are meant to correspond to known entities in the world (e.g., in a vision task each topic may be a type of object), then we can simply set K equal to the true number of entities we are trying to model.

The third approach optimizes performance on secondary task. If the learned topics are to be used as input to another task, then it follows that K should be chosen according to performance on the ultimate task of interest. For example, if the topics are going to be used for document classification, K could be chosen to optimize classifier performance on a held-aside validation set.

Finally in the fourth approach the likelihood of held aside documents is optimized. Given a means of evaluating the probability P(w|K) of a validation set of held aside documents (Wallach et al., 2009), we can learn topics for different values of K and choose the value which maximizes the likelihood of the held-aside validation set (Rosen et al, 2004). Plotting these values, we can typically see the familiar pattern of P(w|K) increasing with larger K up to a point, beyond which the model overfits (Mitchell, 1997) the data and P(w|K) on the held-aside documents begins to fall. This study precisely uses mathematical theory to model this behavior with an aim of reducing the time taken to estimate K.

1.2 Model Evaluation: Likelihood and Perplexity

There are two ways of evaluating LDA. One is by measuring performance on some secondary task, such as document classification or information retrieval, and the second is by estimating the probability of unseen held-out documents given some training documents (Wallach 2009). This second approach is the most common way used to evaluate a probabilistic model and is achieved by measuring the log-likelihood of a held-out test set.

When applying the second approach we split the dataset into two parts: one for training, the other for testing. For LDA, a test set is a collection of unseen documents $\mathbf{w}_d$, and the model is described by the topic matrix Φ and the hyper-parameter α for topic-distribution of documents. This leads to the following function that we need to evaluate inorder to compute the log-likelihood:

$$\mathcal{L}(w) = log\ p\ (w|\Phi, \alpha) = \sum_d log\ p\ (w_d|\ \Phi, \alpha)$$

The computed likelihood of unseen documents can be used to compare models with a higher likelihood implying a better model. A further definition leads to perplexity measure which according to research is the most typical measure for evaluating LDA models (Bao & Datta, 2014; Blei et al., 2003)

Perplexity measures the modeling power by calculating the inverse log-likelihood of unobserved documents. Traditionally we use perplexity which is defined interms of likelihood as follows:

$$\text{perplexity(test set w)} = \exp\left\{-\frac{\mathcal{L}(\boldsymbol{w})}{\text{count of tokens}}\right\}$$

Perplexity is a decreasing function of the log-likelihood $\mathcal{L}(w)$ of the unseen documents $\mathbf{w}_d$ and lower perplexity corresponds to higher likelihood. This means that better models have lower perplexity which suggests less uncertainties about unobserved documents. The likelihood $p\ (\boldsymbol{w}_d|\Phi, \alpha)$ of one document however is intractable, which makes the evaluation of $\mathcal{L}(w)$, and therefore the perplexity, intractable as well.
The advantage of perplexity as a measurement is that it is normalized to the size of the data and can thus be compared across different datasets.

2. Materials

In this section we describe the experimental design for the study. We first describe the datasets, the evaluation procedures and the relevant experimental setup.

2.1 Dataset and Software Used

This experiment demonstrates use of Latent Dirichlet Allocation on an existing dataset called the 20 Newsgroups dataset which is a collection of approximately 20,000 newsgroup text documents, partitioned evenly across 20 different groups. The data can be found at GitHub and was originally collected by Lang (1995) for the learning to filter netnews paper. The dataset has become popular for experiments in text applications of machine learning techniques, such as text classification and text clustering as evidenced in a lot of published work for example Albishre & Li (2015), Mekala et. al. (2017), Dasgupta et. al. (2007), Harish & Revanasiddappa (2017) and Feng et. al (2017).

For topic modeling task there are many software packages that can be used to train a model such as Mallet, Matlab, R package etc. In this study, we set up Python 3.6 from Continuum Analytics Anaconda and configured scikit-learn, gensim and graphlab create libraries to work in jupyter notebook for data and procedures as the development environment set up in the google's cloud compute engine, virtual machine instances.

2.2 Hardware Environment

The experiments in this study were conducted under the google cloud compute engine's custom machine with a specification of 8 CPUs, 8 GB memory and 64 GB boot and local disk. This was accessed with a personal laptop with a CPU specification of Intel(R) Core(TM) i5-4210U CPU @ 1.70 GHz 2.40 GHz and a memory of 4.00 GB. The capacity of personal laptop hard drive is 500 GB running Microsoft Windows 8.1 single Language 64-bit Operating System, x64-based processor.

3. Methods

This study followed the experimental model to analyse the 20 Newsgroups datasets and to make a critical evaluation of the behaviour of different LDA hyperparameter settings. The researcher used these datasets as secondary data for purposes of experimentation and some tools developed by other researchers like scikit-learn, a python machine learning library (Pedregosa et at, 2011) and graphlab create, a framework for parallel machine learning (Low, et al 2014).

The research first set up graph lab environment on jupyter notebook in the google cloud compute engine's custom machine and then developed python code to run on this environment. The code was used to manipulate the data in the desired way and iterated many times on given datasets. Results were displayed on a two dimensional graph using python code. This allowed the researchers to observe the behaviour of parameters of interest under different settings, thereby conceptualising the set modelling goals. This procedure was repeated for different datasets with an aim of generalisation.

4. Results

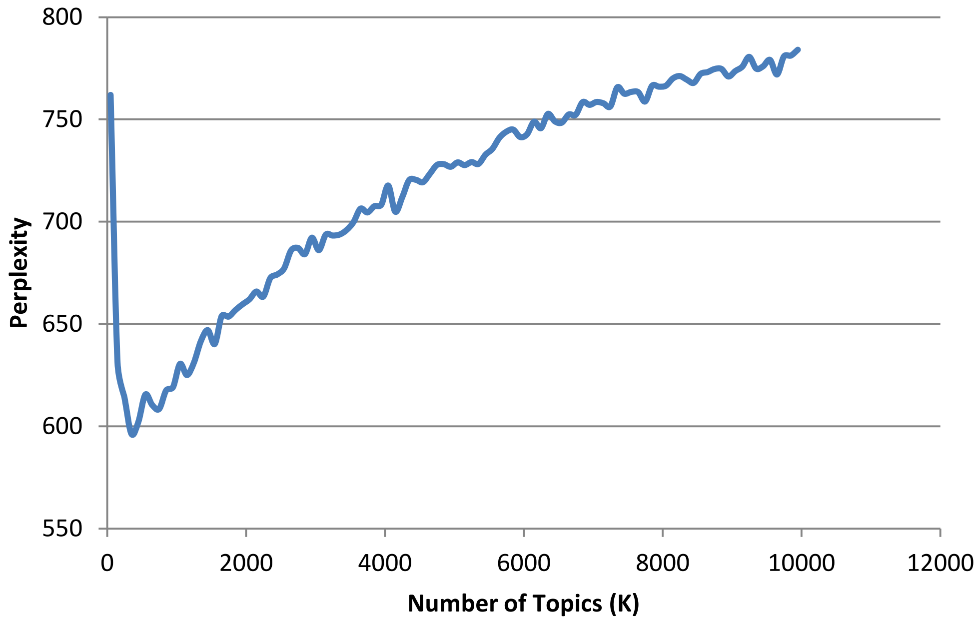
In this experiment, we set alpha at 0.1 and beta at 0.01 and K at 10,000, a sufficiently large value of K inorder to determine the 'infinite' trend of the graph. We observed that the value of perplexity increases monotonically beyond a point of minimum perplexity as K increases. This is illustrated on the graph shown below.

4.1 Perplexity against K for 50< K<10,000

The data obtained from this experiment is as tabulated below:

Topics	50	150	250	350	450	550	650	750
Perplexity	761.941	631.164	614.066	596.333	602.222	615.543	610.431	608.542
Topics	850	950	1050	1150	1250	1350	1450	1550
Perplexity	617.592	619.231	630.557	625.07	631.121	641.633	647.042	640.235
Topics	1650	1750	1850	1950	2050	2150	2250	2350
Perplexity	653.918	653.63	656.88	659.629	662.074	665.896	663.425	672.43
Topics	2450	2550	2650	2750	2850	2950	3050	3150
Perplexity	674.223	677.205	685.971	687.191	684.171	692.203	686.092	693.692
Topics	3250	3350	3450	3550	3650	3750	3850	3950
Perplexity	693.242	693.745	695.944	699.85	706.362	704.531	707.583	708.31
Topics	4050	4150	4250	4350	4450	4550	4650	4750
Perplexity	717.661	704.886	711.811	720.298	720.451	719.309	723.427	727.688
Topics	4850	4950	5050	5150	5250	5350	5450	5550
Perplexity	728.115	726.841	728.995	727.652	729.127	728.233	732.71	735.59
Topics	5650	5750	5850	5950	6050	6150	6250	6350
Perplexity	740.957	743.94	744.981	741.366	742.753	748.873	745.674	752.679
Topics	6450	6550	6650	6750	6850	6950	7050	7150
Perplexity	749.02	748.4	752.385	752.218	758.425	757.087	758.486	757.834
Topics	7250	7350	7450	7550	7650	7750	7850	7950
Perplexity	756.324	765.671	762.523	763.395	763.3	758.705	766.404	765.957
Topics	8050	8150	8250	8350	8450	8550	8650	8750
Perplexity	766.445	769.983	771.147	769.393	767.823	772.216	773.087	774.578
Topics	8850	8950	9050	9150	9250	9350	9450	9550
Perplexity	774.63	770.927	773.604	775.734	780.531	774.786	775.921	779.034
Topics	9650	9750	9850	9950				
Perplexity	771.91	780.743	781.154	784.001				

This table was represented on a line graph as shown on the graph 4.1 below:



Graph 4.1: Perplexity against K for 50< K<10,000

From the general trend, we observed that there is a minimum perplexity between a K value of 50 and for purposes of symmetry, an estimate of 690. We therefore changed the limit parameter values to (50, 690, 5) where 5 is the step. The step of five was chosen in order to increase precision in modelling K for this range.

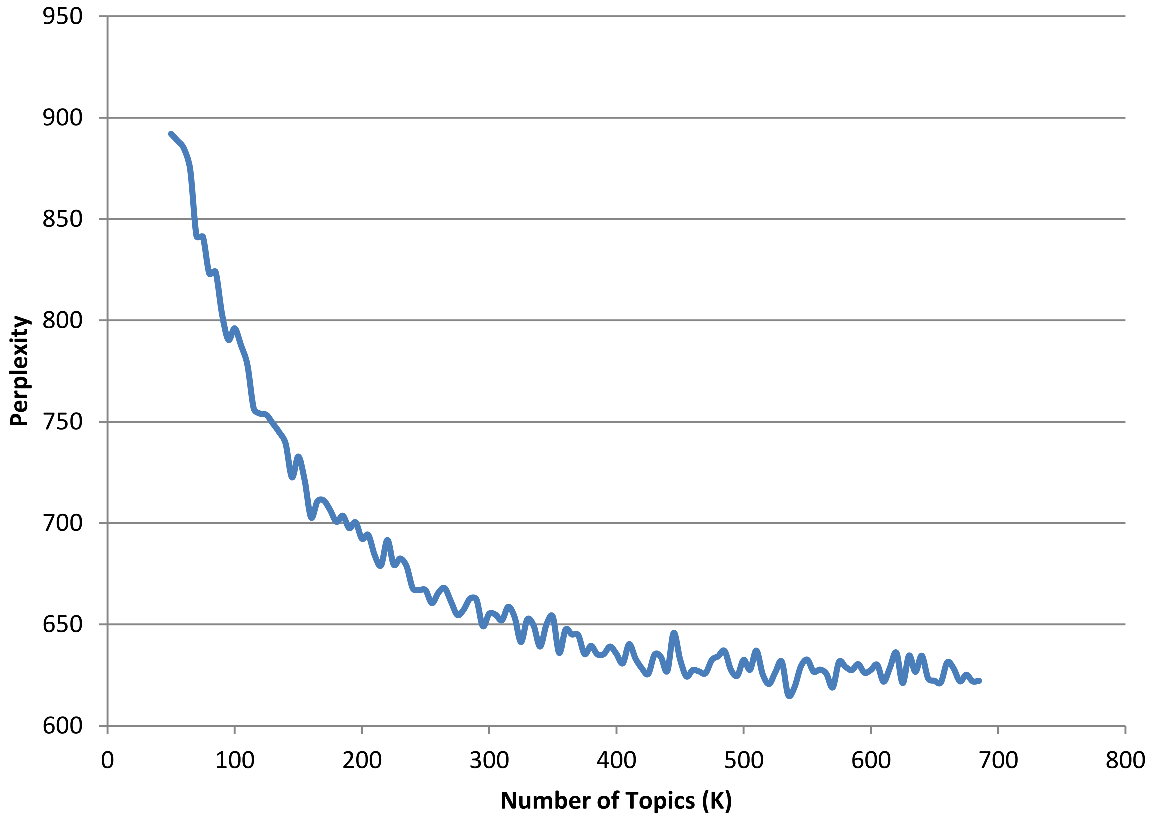
4.2 Perplexity against Topics for 50< K<690

Following is a table and a corresponding line graph for the data obtained for the range parameter values of 50<Topics<690 and a step of 5.

Topics	50	55	60	65	70	75	80	85
Perplexity	892.04	888.72	884.92	874.55	841.59	841.31	823.13	823.92
Topics	90	95	100	105	110	115	120	125
Perplexity	803.26	790.43	796.10	787.66	778.01	756.49	754.00	753.38
Topics	130	135	140	145	150	155	160	165
Perplexity	749.16	744.83	739.30	722.60	732.85	721.10	702.79	710.77
Topics	170	175	180	185	190	195	200	205
Perplexity	711.15	706.45	700.64	703.63	697.53	700.39	692.30	694.13
Topics	210	215	220	225	230	235	240	245
Perplexity	684.22	679.11	691.65	679.49	682.54	678.86	667.77	666.97
Topics	250	255	260	265	270	275	280	285
Perplexity	666.92	660.49	665.58	667.96	661.29	654.68	657.35	662.82
Topics	290	295	300	305	310	315	320	325
Perplexity	662.32	649.18	655.28	654.84	652.06	658.71	653.62	641.35
Topics	330	335	340	345	350	355	360	365
Perplexity	652.58	649.37	639.09	649.92	653.86	635.98	647.44	645.06
Topics	370	375	380	385	390	395	400	405
Perplexity	644.74	635.36	639.48	635.28	635.34	639.09	635.49	630.82
Topics	410	415	420	425	430	435	440	445
Perplexity	640.26	633.19	628.36	625.60	635.12	633.90	627.04	645.82

Topics	450	455	460	465	470	475	480	485
Perplexity	632.90	624.36	627.51	626.81	625.91	632.44	634.32	636.95
Topics	490	495	500	505	510	515	520	525
Perplexity	627.66	624.68	632.53	627.63	637.09	625.40	620.63	626.52
Topics	530	535	540	545	550	555	560	565
Perplexity	631.61	615.19	619.10	629.32	632.64	626.72	627.72	625.70
Topics	570	575	580	585	590	595	600	605
Perplexity	618.94	631.54	629.09	627.48	630.41	626.17	627.53	630.10
Topics	610	615	620	625	630	635	640	645
Perplexity	621.76	628.70	636.12	621.07	634.65	626.61	634.57	623.37
Topics	650	655	660	665	670	675	680	685
Perplexity	622.20	621.36	631.31	628.33	621.93	625.21	621.93	622.17

Table 4.2: Perplexity against Topics for 50< Topics<690



Graph 4.2: Perplexity against Topics for 50< Topics<690

5. Discussion

The results presented above indicates that a graph of perplexity against number of topics (Topics) have a strong quadratic behaviour around a turning point which opens upwards as illustrated on the graph 4.2. This means the graph has a minimum perplexity point that optimizes Topics. Our interest was therefore to model the behaviour of K and to guide the calculation of the optimal value with fewer iterations thereby making a contribution to the body of knowledge. This would enhance the speed of hyper parameter's estimation hence the model for application in big text analytics.

Inorder to model this point of minimum perplexity, the following quadratic equation was set up:

p=aK^2 + bK + C ……………. ***equation 5.0***

where p is the perplexity, K is the number of topics, a, b and C are constants for the general quadratic function.

Inorder to find the optimal value of K, the quadratic equation 5.0 was differentiated, resulting derivative equated to zero and on solving the resulting equation the optimal value of K was obtained as follows:

$\frac{dp}{dK}$=2ka + b

- ⇨ 2Ka + b = 0 at optimum from differential calculus.
- ⇨ **K** = $\frac{-b}{2a}$ ……………. ***equation 5.1***

We further embarked on the task of modeling the evaluation of **a** and **b** for any dataset of interest and used those values to calculate the optimal value of **K** in a general perspective as shown on equation 5.1 above. To accomplish this task, we solved the following set of three general quadratic equations, ***equation 5.2, equation 5.3 and equation 5.4.*** The reason why we set up three equations is because we have three unknowns: **a, b** and **C**. We therefore needed only three data points (k_1, p_1), (k_2, p_2) and (k_3, p_3) to estimate optimum K for any dataset as opposed to the previous approach where many iterations on data has to be performed.

p_1=ak_1^2 + bk_1 + C ……………. ***equation 5.2***

p_2=ak_2^2 + bk_2 + C ……………. ***equation 5.3***

p_3=ak_3^2 + bk_3 + C ……………. ***equation 5.4***

From equation 5.2, **a** = $\frac{p_1-bk_1-C}{k_1^2}$ ……………. ***equation 5.5***

From equation 5.3, **a** = $\frac{p_2-bk_2-C}{k_2^2}$

- ➔ $\frac{p_1-bk_1-C}{k_1^2} = \frac{p_2-bk_2-C}{k_2^2}$
- ➔ $k_2^2(p_1 - bk_1 - C) = k_1^2(p_2 - bk_2 - C)$
- ➔ $k_2^2p_1 - k_2^2bk_1 - k_2^2C = k_1^2p_2 - k_1^2bk_2 - k_1^2C$
- ➔ $k_2^2p_1 - k_1^2p_2 + k_1^2bk_2 - k_2^2bk_1 = k_2^2C - k_1^2C$
- ➔ $(k_2^2p_1 - k_1^2p_2) + bk_1k_2(k_1 - k_2) = C(k_2^2 - k_1^2)$
- ➔ $bk_1k_2(k_1 - k_2) = C(k_2^2 - k_1^2) - (k_2^2p_1 - k_1^2p_2)$
- ➔ $bk_1k_2(k_1 - k_2) = C(k_2^2 - k_1^2) + (k_1^2p_2 - k_2^2p_1)$

Let d = $(k_1^2p_2 - k_2^2p_1)$

- ➔ $bk_1k_2(k_1 - k_2) = C(k_2^2 - k_1^2) + k_1^2p_2 - k_2^2p_1$
- ➔ $b = \frac{(k_1^2p_2-k_2^2p_1)+C(k_2^2-k_1^2)}{k_1k_2(k_1-k_2)} = \frac{k_1^2p_2-k_2^2p_1}{k_1k_2(k_1-k_2)} + \frac{C(k_2^2-k_1^2)}{k_1k_2(k_1-k_2)}$
- ➔ b = e + Cf

Where

$e = \frac{d}{k_1k_2(k_1-k_2)}$ and $f = \frac{k_2^2-k_1^2}{k_1k_2(k_1-k_2)}$ …….………. ***equation 5.6***

$\therefore b = \frac{k_1^2p_2-k_2^2p_1}{k_1k_2(k_1-k_2)} + \frac{C(k_2^2-k_1^2)}{k_1k_2(k_1-k_2)}$

From equation 5.5,

- ➔ **a** = $\frac{p_1-(\text{e + Cf})k_1-C}{k_1^2}$
- ➔ **a** = $\frac{p_1-(\text{e + cf})k_1-C}{k_1^2} = \frac{p_1}{k_1^2} - \frac{ek_1}{k_1^2} - \frac{Cf}{k_1^2} - ck_1^2$
- ➔ **a** = $\frac{1}{k_1^2}(p_1 - ek_1) - C\left(\frac{f}{k_1^2} + k_1^2\right) = g - C(\frac{f+k_1^4}{k_1^2})$
- ➔ a = g – Ch

where $g = \frac{1}{k_1^2}(p_1 - ek_1)$ and $h = \frac{f+k_1^4}{k_1^2}$

Equation 5.4 can therefore be restated as follows:

$$p_3 = (g - Ch)k_3^2 + (e + Cf)k_3 + C = gk_3^2 - Chk_3^2 + ek_3 + Cfk_3 + C$$

$$\Rightarrow Chk_3^2 - Cfk_3 - C = gk_3^2 + ek_3 - p_3$$

$$\Rightarrow C(hk_3^2 - fk_3 - 1) = gk_3^2 + ek_3 - p_3$$

$$\Rightarrow C = \frac{gk_3^2 + ek_3 - p_3}{hk_3^2 - fk_3 - 1}$$

We proceeded to remove arbitrary variables d, e, f, g and h inorder to find the generic values of a, b and C as follows:

$$C = \frac{\left(\frac{1}{k_1^2}\left(p_1 - \left(\frac{d}{k_1k_2(k_1-k_2)}\right)k_1\right)\right)k_3^2 + \left(\frac{d}{k_1k_2(k_1-k_2)}\right)k_3 - p_3}{\left(\frac{\left(\frac{k_2^2 - k_1}{k_1k_2(k_1-k_2)}\right) + k_1^4}{k_1^2}\right)k_3^2 - \left(\frac{k_2^2 - k_1^2}{k_1k_2(k_1-k_2)}\right)k_3 - 1}$$

$$a = \frac{1}{k_1^2}\left(p_1 - \left(\frac{k_1^2p_2 - k_2^2p_1}{k_1k_2(k_1-k_2)}\right)k_1\right) - C\left(\frac{f + k_1^4}{k_1^2}\right)$$

$$b = \frac{k_1^2p_2 - k_2^2p_1}{k_1k_2(k_1-k_2)} + \frac{C(k_2^2 - k_1^2)}{k_1k_2(k_1-k_2)}$$

Which can be stated as follows when using arbitrary variables d, e, f, g and h

$$C = \frac{gk_3^2 + ek_3 - k_3}{dk_3^2 - fk_3 - 1}$$

a = $g - ch$

b = $e + cf$

where:

$d = k_1^2p_2 - k_2^2p_1$; $e = \frac{d}{k_1k_2(k_1-k_2)}$; $f = \frac{k_2^2 - k_1}{k_1k_2(k_1-k_2)}$

$g = \frac{p_1 - ek_1}{k_1^2}$ and $h = \frac{f + k_1^4}{k_1^2}$

Further the following algorithm for estimating topic model hyperparameters was conceptualised:

Algorithmic Steps: Pseudocode for our proposed estimator.

Initialise α, β and K

FOR iteration i=1, 2, 3

do

READ k_i , p_i

END FOR

$C \leftarrow \frac{gk_3^2 + ek_3 - k}{dk_3^2 - fk_3 - 1}$

$a \leftarrow g - ch$

$b \leftarrow e + cf$

$\boldsymbol{K} \leftarrow \frac{-b}{2a}$

end function

where:

$d = k_1^2p_2 - k_2^2p_1$; $e = \frac{d}{k_1k_2(k_1-k_2)}$; $f = \frac{k_2^2 - k_1}{k_1k_2(k_1-k_2)}$; $g = \frac{p_1 - ek_1}{k_1^2}$ *and*

$h = \frac{f + k_1^4}{k_1^2}$

6. Conclusion

In the collapsed gibbs sampling algorithm, the values of the Dirichlet priors, α and β are assumed to be known (Nguyen et al. 2012). Many researchers in the area of topic modelling use the heuristic values for the hyper parameters. In particular, common values are α = 50/K where K is the total number of topics. The experiments on beta shown above indicates that a value of β = 0.01 yields better likelihood for all test cases and therefore is a reasonable estimate.

In conclusion, the study established that the iterative procedure of finding optimal number of topics can be modified in order to reduce number of iterations as follows: First initialise α and β to 0.1 and 0.01 respectively and then iterate K three times from a value of 250 with a step of 50. We then use quadratic formula and differential calculus to obtain optimal value of K for the particular dataset. To obtain the new alpha, we use the relationship $\alpha = 50/K$. From experimentation, it has also been shown that a β value of 0.01 is optimal.

Acknowledgement:

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] Albishre, K., Albathan, M., & Li, Y. (2015, December). Effective 20 newsgroups dataset cleaning. In *Web Intelligence and Intelligent Agent Technology (WI-IAT), 2015 IEEE/WIC/ACM International Conference on* (Vol. 3, pp. 98-101). IEEE.

[2] Bao, Y., & Datta, A. (2014). Simultaneously discovering and quantifying risk types from textual risk disclosures. *Management Science*, *60*(6), 1371-1391.

[3] Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent dirichlet allocation. *The Journal of Machine Learning Research*, *3*, 993–1022.

[4] Dasgupta, A., Drineas, P., Harb, B., Josifovski, V., & Mahoney, M. W. (2007, August). Feature selection methods for text classification. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 230-239). ACM.Hofmann, T. (1999). Probabilistic latent semantic indexing. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval* (pp. 50–57).

[5] Feng, X., Liang, Y., Shi, X., Xu, D., Wang, X., & Guan, R. (2017). Overfitting Reduction of Text Classification Based on AdaBELM. *Entropy*, *19*(7), 330.

[6] Harish, B. S., & Revanasiddappa, M. B. (2017). A Comprehensive Survey on various Feature Selection Methods to Categorize Text Documents. *International Journal of Computer Applications*, *164*(8).

[7] Lang, K. (1995). Newsweeder: Learning to filter netnews. In *Proceedings of the 12th international conference on machine learning* (Vol. 10, pp. 331-339).

[8] Low, Y., Gonzalez, J. E., Kyrola, A., Bickson, D., Guestrin, C. E., & Hellerstein, J. (2014). Graphlab: A new framework for parallel machine learning. *arXiv preprint arXiv:1408.2041*.

[9] Mekala, D., Gupta, V., Paranjape, B., & Karnick, H. (2017). SCDV: Sparse Composite Document Vectors using soft clustering over distributional representations. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing* (pp. 670-680).

[10] Mitchell, R. K., Agle, B. R., & Wood, D. J. (1997). Toward a theory of stakeholder identification and salience: Defining the principle of who and what really counts. *Academy of management review*, *22*(4), 853-886.

[11] Nguyen, P. X., Wang, P., & Nam, S. (2012). Experiments with Latent Dirichlet Allocation.

[12] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., & Vanderplas, J. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*(Oct), 2825-2830

[13] Roberts, M. E., Stewart, B. M., & Tingley, D. (2015). STM: R package for structural topic models. R package version 1.1. 0.

[14] Rosen-Zvi, M., Griffiths, T., Steyvers, M., & Smyth, P. (2004, July). The author-topic model for authors and documents. In *Proceedings of the 20th conference on Uncertainty in artificial intelligence* (pp. 487-494). AUAI Press.

[15] Wallach, H. M. (2008). *Structured topic models for language* (Doctoral dissertation, University of Cambridge).

[16] Wallach, H. M., Murray, I., Salakhutdinov, R., & Mimno, D. (2009, June). Evaluation methods for topic models. In *Proceedings of the 26th annual international conference on machine learning* (pp. 1105-1112). ACM.Wang, B., Liu, Y., Liu, Z., Li, M., & Qi, M. (2014, August). Topic selection in latent dirichlet allocation. In *Fuzzy Systems and Knowledge Discovery (FSKD), 2014 11th International Conference on* (pp. 756-760). IEEE.

[17] Wang, B., Liu, Y., Liu, Z., Li, M., & Qi, M. (2014, August). Topic selection in latent dirichlet allocation. In *Fuzzy Systems and Knowledge Discovery (FSKD), 2014 11th International Conference on* (pp. 756-760). IEEE.

[18] Zhao, W., Chen, J. J., Perkins, R., Liu, Z., Ge, W., Ding, Y., & Zou, W. (2015). A heuristic approach to determine an appropriate number of topics in topic modeling. *BMC bioinformatics*, *16*(13), S8.
[19] Hutter, F., Hoos, H., & Leyton-Brown, K. (2014, January). An efficient approach for assessing hyperparameter importance. In *International Conference on Machine Learning* (pp. 754-762).

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.