Gibbs Sampling And Non-Negative Matrix Factorization For Topic Modeling and Text Classification

Afeka College of Engineering,
Department Of Intelligent Systems

Rotem Bar 304840937
Uria Levkovich 036841377
Ravid Sherma 307834051
Pazit Lazar 204059919

ABSTRACT

Topic modeling for text classification is an unsupervised machine learning approach for finding abstract topics in a large collection of documents. It scans a set of documents, detects word and phrase patterns within them, and automatically clusters word groups and similar expressions that best characterize a set of documents. It is an important concept of the traditional Natural Processing Approach because of its potential to obtain semantic relationships between words in the document clusters. Gibbs sampling is a method used to approximate the posterior distribution of a parameter of interest. It is used as an approximation inference algorithm to enable Latent Dirichlet allocation (LDA) to approximate posterior for classification. LDA and non-negative matrix factorization (NMF) are common frameworks used for models to detect topics. LDA - Gibbs uses a probabilistic approach, whereas NMF uses a matrix factorization approach. This article will describe topic modeling, how LDA using Gibbs sampling and NFM models work, and how to implement a working system to perform learning with a topic model.

1. INTRODUCTION

Topic modeling is a branch of unsupervised natural language processing used to represent a text document with the help of several topics [1] that can explain the underlying information in a particular document [2]. A topic represents a collection of words that are grouped. A trained model may then be used to discern which of these topics occur in new documents. A first step in identifying the content of a document is determining which topics that document addresses. Very commonly used models in topic classification are latent Dirichlet analysis (LDA) [3] using Gibbs sampling [4] and non-negative matrix factorization (NMF). The aim of LDA is to find topics a document belongs to, based on the words in it. This algorithm finds the weight of connections between documents and topics and between topics and words. Using the Gibbs sampling method, the correct weights of the data will be identified. NMF is a statistical method for reducing the dimension of the input corpus using the factor analysis method.

2. Gibbs Sampling - overview

2.1. Background

Gibbs sampling is a popular approach to build a Monte Carlo Markov Chain (MCMC), a sampling method used to approximate the posterior distribution of a parameter of interest by randomly sampling from a probability distribution and constructing a Markov chain. It aims to describe a sequence of possible events, where, to approximate any current event step, it depends on the state attained from the previous one. Gibbs sampling is a specific case of MCMC technique called Metropolis-Hasting algorithm. As opposed to the latter, proposals are always accepted in the prior case (which improves efficiency). Gibbs Sampling generates a Markov chain of samples, each of which is calculated with its direct neighbors. For example, in Markov Random Field, each sample is associated with its Markov Blanket. This independency attribute simplified the problem and only needed the conditional probability P(s|s_neighbors) to get a sample value for states. It relies on the ability to sample from the conditional distributions of the target distribution (when the target distribution is not known explicitly or it is difficult to sample from directly). Figures 1 illustrate a multivariate probability distribution, where suppose there is a need to sample from it $\mathcal{P}(X,Y)$ (green), but it is impossible. Suppose the conditional probabilities $\mathcal{P}(X|Y)$ and $\mathcal{P}(Y|X)$ are known - by sampling from those probabilities enough iterations, we will obtain a close estimation of the multivariate probability distribution.

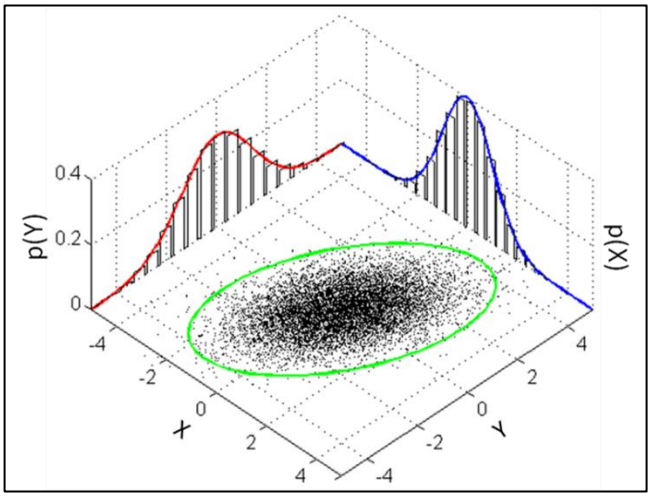

Figure 1: Multivariate probability distribution function of multiple variables

2.2. Gibbs algorithm

Let θ be a $K\times1$ parameter vector with associated posterior distribution $\mathcal{P}(\theta|\mathcal{Y})$ and write θ=[θ1 θ2…θK].

The Gibbs sampling algorithm proceeds as follows:

1. Select an initial parameter vector θ(0)= [θ1(0) , ,θ2(0), …θK(0)]. This initial value could be arbitrarily chosen, sampled from the prior, or obtained from a crude estimation method such as least squares.
    a. Sample θ1(1) from the complete posterior conditional density:
    $\mathcal{P}\left(\theta_1|\theta_2=\theta_2^{(0)},\theta_3=\theta_3^{(0)},\ldots\theta_k=\theta_k^{(0)},y\right)$

    b. Sample θ2(1) from
    $\mathcal{P}\left(\theta_2|\theta_1=\theta_1^{(1)},\theta_3=\theta_3^{(0)},\ldots\theta_k=\theta_k^{(0)},y\right)$

    (k) Samples $\theta_k^{(1)}$ from

$$\mathcal{P}\left(\theta_{\mathrm{k}} \middle| \theta_1 = \theta_1^{(1)}, \theta_3 = \theta_3^{(1)}, \ldots \theta_k = \theta_{k-1}^{(1)}\right)$$

2. Repeatedly cycle through (1)→(K) to obtain $\theta(2)$ = [$\theta 1(2)$, $\theta 2(2)$, $\theta K(2)$], $\theta(3)$, etc., always conditioning on the most recent values of the parameters drawn (e.g., to obtain $\theta 1(2)$, draw from $\mathcal{P}\left(\theta_1 \middle| \theta_2 = \theta_2^{(1)}, \theta_3 = \theta_3^{(1)}, \ldots \theta_k = \theta_k^{(1)}\right), etc)$.

2.3. Gibbs Summery

The Gibbs Sampling is a Monte Carlo Markov Chain method that iteratively draws an instance from the distribution of each variable, conditional on the current values of the other variables to estimate complex joint distributions. Many interesting problems present A difficulty with sampling from suitable conditional distributions. Even if a closed-form expression for these distributions can be obtained, it is often impossible to obtain samples. Therefore, using this method presents a simplification and often is the only possible solution to sampling.

3. TOPIC MODELS

3.1. LDA - intro

LDA is a generative probabilistic framework for modeling sparse discrete vectors (BOW, image features) [13]. In the context of text data and topic modeling, the main assumption is that words in each document are generated by a mixture of topics [5], where a topic is represented as a multinomial probability distribution over words. The mixing coefficients for each document and the term-topic distributions are hidden and are learned using unsupervised learning methods. Assuming that documents are represented as random mixtures over latent topics, where each topic is characterized by a distribution over words, we can describe the generative process as follows:

Draw each topic parameter $\boldsymbol{\beta}_k \sim$ Dirichlet ($\boldsymbol{\varphi}$), for $k \in \{1 \ldots K\}$

For each document:

Choose the topic distribution $\boldsymbol{\theta}_{\mathrm{m}} \sim$ Dirichlet ($\boldsymbol{\alpha}$)

For each of the N words w_n :

a) Choose a topic $z_{mn} \sim$ Multinomial($\boldsymbol{\theta}_{\mathrm{m}}$)

b) Choose a word $w_n \sim$ Multinomial($\boldsymbol{\beta}_{\mathrm{k}}$)

The probability of a corpus (D) is the result of taking the product of the marginal probabilities of single documents, and the marginal distribution for a single document is obtained by integrating over $\boldsymbol{\theta}$ and summing over *z* topics.

Because the prior is Dirichlet distributed ($\boldsymbol{\alpha}$, $\boldsymbol{\beta}$) and the likelihood is multinomial distributed (z_{mn}, w_{mn}), the posterior is Dirichlet distributed and can be computed – but the term is intractable for exact inference.

$$P(\theta_{1:M}, \mathbf{z}_{1:M}, \beta_{1:k} | \mathcal{D}; \alpha_{1:M}, \eta_{1:k})$$

To solve this issue, approximation inference algorithms such as Laplace approximation, MCMC (Gibbs sampling) and variational Bayes algorithm are presented.

3.2. LDA and Gibbs sampling

The task of the Bayesian inference is to compute [13] the posterior distribution over the latent topic indices **z**, the mixing proportions θ, and the topics φ, given the observed words. collapsed Gibbs sampling[4] is an efficient inference procedure, where the mixing proportions and topics are marginalized out, while only the latent variables **z** are sampled. After the sampler has burned-in we can calculate and estimate the latter parameters given latent variables. Gibbs sampling works because the limiting distribution of θ and φ is the desired posterior distribution. However, it is only so after many iterations. For this reason, it is necessary to discard the initial observations (i.e., the burn-in).

3.3. Collapsed Gibbs sampling algorithm

Let summations of the data being $N_{wjk} = \#\{i: \mathcal{X}_{ij} = w, z_{ij} = k\}$ and use the convention that missing indices are summed out $N_{jk} = \sum_w N_{wjk}$ and $N_{wk} = \sum_j N_{wjk}$ (N_{jk} is the number of times a word in document j has been assigned to topic k and N_{wk} is the number of times the word w is assigned to the topic k). Given the current state of all but one variable z_{ij}, the conditional probability is then:

$$p\left(z_{ij} = k \middle| \boldsymbol{z}^{\neg \boldsymbol{ij}}, \boldsymbol{x}, \alpha, \beta\right) = \frac{1}{Z} \alpha_{jk} \beta_{wk} \text{ where}$$

$$\alpha_{kj} = N_{kj}^{\neg ij} + \alpha \qquad \beta_{wk} = \frac{N_{wk}^{\neg ij} + \beta}{N_k^{\neg ij} + W\beta`}$$

Z is the normalization constant

$$Z = \sum_k \alpha_{kj} \beta_{wk}$$

the superscript ¬ij indicates that the corresponding datum has been excluded in the count summations N_{wjk}.

```
for i ← 1 to N
  do
{ u ← draw from Uniform[0, 1]
  for k ← 1 to K
    do
  { P[k] ← P[k − 1] + (N_kj^¬ij + α)(N_x_ij k^¬ij + β) / (N_k^¬ij + Wβ)
  for k ← 1 to K
    do
  { if u < P[k]/P[K]
    then z_ij = k, stop
```

Figure 3. Gibbs sampling algorithm

An iteration proceeds by drawing a sample for z_{ij} for each word i in each document j:

- calculate the normalization constant Z
- sample z_{ij} according to its normalized probability.
- Given the value sampled for z_{ij}, update the counts N_{jk} , N_k, N_{wk}
- Given a sample we can then get an estimate for $\hat{\theta}_j$ and $\hat{\varphi}_k$

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

Figure 4. Gibbs sampling mathematical expression

To simplify, Gibbs mathematical expression (figure 4) is explained as followed:

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

There are two parts two this equation. The first part tells us how much each topic is present in a document and the second part tells how much each topic likes a word. For each word, a vector of probabilities will explain how likely this word belongs to each of the topics. Finally, a topic will be randomly picked and will be assigned to a word, and then this step will be repeated for all other words as well.

3.4. NMF

Non-negative matrix factorization is a *linear*, *non-negative* approximate data representation [17] . Let's assume that our data consists of *T* measurements of *N* non-negative scalar variables. Denoting the (*N*-dimensional) measurement vectors $\mathbf{v}^t(t = 1, \ldots, T)$ a linear approximation of the data is given by:

$$V^t \approx \sum_{i=1}^{M} w_i h_i^t = Wh^t$$

Where **W** is an *N* ×*M* matrix containing the *basis vectors* **w***i* as its columns. The *M* basis vectors **w***i* can be thought of as the 'building blocks' of the data, and the (*M*-dimensional) coefficient vector $\mathbf{h}^t$ describes how strongly each building block is present in the measurement vector $\mathbf{v}^t$. Arranging the measurement vectors **v***t* into the columns of an *N*×*T* matrix **V**, we can now write: V ≈ WH

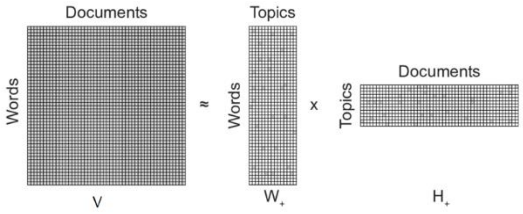

Figure 5: Non-negative matrix factorization diagram

Where each column of **H** contains the coefficient vector $\mathbf{h}^t$ corresponding to the measurement vector $\mathbf{v}^t$. Written in this form, it becomes apparent that a linear data representation is simply a factorization of the data matrix. Principal component analysis(PCA), independent component analysis, vector quantization, and non-negative matrix factorization can all be seen as matrix factorization, with different choices of the objective function. Whereas PCA does not in any way restrict the signs of the entries of **W** and **H**, NMF requires all entries of both matrices to be non-negative. What this means is that the data is described by using additive components only. Matrix V is factorized into a n*t document-topic matrix and a t*m topic-term matrix, where t is the number of topics produced [19]. Equation V ≈ WH is achieved by finding *W* and *H* that minimize the error function $||V - HW||$ where W>0 , H>0 . A common method of measuring how good the approximation W, H is by the Frobenius norm:

$$||V - WH||^2 \approx \sum (V - WH)^2$$

(Hoyer PATRIKHOYER, 2004)There are several ways in which the W and H may be found, Lee and Seung's multiplicative update rule algorithm is:

- initialize W and H non-negative.
- Then update the values in W and H by computing the following, with n as an index of the iteration.

$$\mathbf{H}^{n+1}_{[i,j]} \leftarrow \mathbf{H}^{n}_{[i,j]} \frac{((\mathbf{W}^n)^T\mathbf{V})_{[i,j]}}{((\mathbf{W}^n)^T\mathbf{W}^n\mathbf{H}^n)_{[i,j]}}$$

and

$$\mathbf{W}^{n+1}_{[i,j]} \leftarrow \mathbf{W}^{n}_{[i,j]} \frac{(\mathbf{V}(\mathbf{H}^{n+1})^T)_{[i,j]}}{(\mathbf{W}^n\mathbf{H}^{n+1}(\mathbf{H}^{n+1})^T)_{[i,j]}}$$

Figure 6: H , W Matrix iterative update

Where n is the iteration index, and i,j are the indexes in the matrix.

4. COHERENCE MEASURES

Coherence can be defined as a set of statements or facts that are connected by meaning and support each other. A coherent fact set can be interpreted in a context that covers all or most of the facts. There are different coherence measures [18] and each of them is based on a different method of calculation. We will focus on the most popular one, the CV. CV is based on a sliding window, a one-set segmentation of the top words, and an indirect confirmation measure that uses normalized pointwise mutual information (NPMI) and the cosine similarity. This CV coherence measure retrieves co-occurrence counts for the given words using a sliding window. The counts are used to calculate the NPMI of every top word to every other top word, thus, resulting in a set of vectors, one for every top word. The one-set segmentation of the top words leads to calculating the similarity between every top word vector and the sum of all top word vectors.

$$MPMI = \sum f(w,k|\alpha) = \alpha log(\varphi_{k,w}) + (1-\alpha) log(\frac{\varphi_{k,w}}{p_w})$$

Where $\varphi_{k,w}$ is the probability of word W in topic k, and p_w is the probability of the word w in the corpus. α is a value between 0 and 1 that determines the relative importance of first and second terms.

5. EXPERIMENTATION

5.1. *Data Set*

In our work, we will use two datasets to examine the models, 'ABC-news' and 'CNN-news'. Both contain sentences from the media, so they contain many topics. The main difference between them is the length of the average sample. Where in ABC dataset is 731.8 compared to 6.5 in CNN. The number of samples in the CNN dataset is 301 compared to over 1 million samples on the ABC dataset. In our work, we will examine 300,000 documents from the ABC dataset.

5.2. Pre-Processing

To work with the raw data, we want to do two things, first is to transform the text to a suitable format for the model. Second, we want to clean the words from irrelevant information. Therefore, we will perform several actions such as splitting the sentences into tokens based on a word, dropping characters that are not letters, and dropping words with no meaningful semantic meaning, such as 'I', 'me', 'our'. Additionally, we will perform stemming, which conveys a word to the base form.

6. RESULTS

After the pre-processing, we run each model, Gibbs and NMF, on a topic range of 5 to 30.

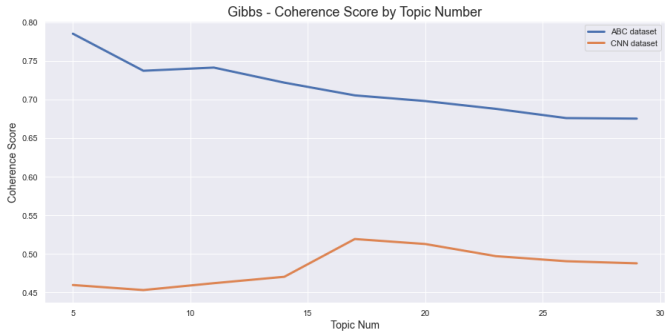

Figure 7: Gibbs Coherence score by topic number

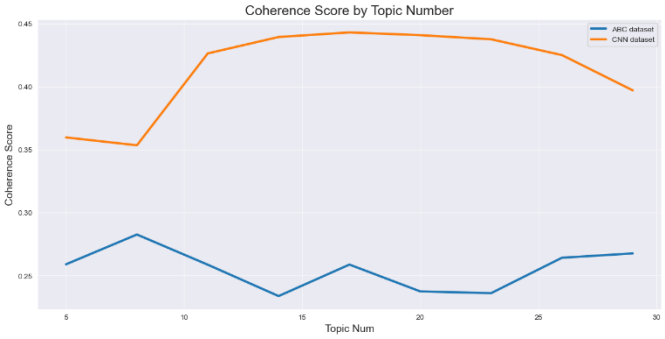

Figure 8: NMF Coherence score by topic number

We will run each of them on the optimal topic number that the Coherence measure found to test our models. We will present the first eight topics from each model on the CNN dataset.

Topic	Top Words
1	amazon social facility virus distancing quarantine
2	mask ventilator production produce patient face
3	plan announce furlough step part executive
4	cut low production global world barrel
5	game life family lot school event
6	order delivery state essential stay local
7	financial stock share investor fund dollar
8	video release live zoom film studio

Table 1: Gibbs model -CNN words in each topic

Topic	Top Words
1	instacart worker shopper custom gig order
2	oil saudi price arabia russia barrel
3	loan payment small busi bank mortgag
4	trump fox news brief hanniti presid
5	airlin flight passeng carrier fli travel
6	app user zoom data googl inform
7	ventil ford devic patient product design
8	store retail close open maci tjx

Table 2: NMF model -CNN words in each topic

In addition, we will examine the runtime of the two models. The time is measured according to the time it took for the models to produce the graphs in figures (4) and (5).

	Gibbs model	**NMF model**
CNN dataset	589 [s]	185 [s]
ABC dataset	745 [s]	630 [s]

Table 3: Running time comparison

6.1. Comparison

Comparing the two models, NMF and Gibbs, according to the Coherence measure, we see that the models presented considerably similar results in the CNN database, which has a small number of documents (300). In contrast, in the ABC database with a large number of documents (300,000), the results in the NMF model decreased significantly compared to the Gibbs model. When we come to analyze the results, a human analysis must be performed in addition to the Coherence measure. We will examine the words that the models found in each topic and see if we can understand what they have in common in the real world. When we look at Table (2), we see that the NMF model presents topics that are easy to identify. For example, we can say that topics 1-6 are: Oil & Gas industry, Economy, Politic, Flights, Technology, Cars. Looking at Table (1), we see that for some rows, it is more difficult for us to find a theme between the words in each topic and define the topic they have in common. In addition, we obtain the result produced by the Gibbs model on the ABC database. The model showed a 0.78 Coherence score, a high-value score compared to the literature, where the results are slightly lower. However, after human analysis, we discovered that there is little to no commonality between the top words in each topic, as presented in table (3). I order to try and explain these results, we would recommend focusing on the balance between topic and vocabulary number [15]. Specifically, the larger the dataset, the greater the number of topics expected, But only if the dataset represents a diverse collection. Therefore, a careful examination and balance have to be made between the reduction of the initial dictionary, in contrast to the ability to capture the majority of important words and heterogeneity of the actual data set. When we look at the optimal number of topics that each model found, we see that the results are close between the models, and these results mainly indicate the structure of each dataset. From table number (3), we see that NMF has a significantly lower runtime for the CNN database with a small number of documents. On the other hand, the runtime increases significantly in the ABC dataset that contains many documents. However, in the Gibbs model, the runtime does not vary significantly between the two types of data sets. These results are consistent with the literature and theory. In the NMF model, an increase in the number of documents causes an increase in the V matrix, which increases the complexity of the problem. Compared to Gibbs, which can better handle a large number of documents, and its runtime depends on the total amount of data (when we assume constant hyperparameters) .

7. CONCLUSION

In this paper, we reviewed two different methods used for topic modeling, and we examined the models using several ways. By comparing the results we obtained in the Coherence measure and the human analysis, we see a contradiction between the results. That shows the importance of human analysis and the problem of relying on the Coherence measure. In addition, the results are consistent with the literature we have reviewed. Generally, a dataset is unsuitable for topic modeling for short-length documents, too small datasets, or many topics within a collection. Fine-tuning those values ranges is an important task that revolved a combination of human analysis and measure methods (as coherence score). There are guidelines [15] for handling this topic. Sufficiently minimum size of documents (M) is crucial for sufficient modeling. However, Once a viable M value is achieved, further increasing it may not significantly improve performance. It is also recommended to avoid selecting an overly large k value – It might cause inference to become inefficient. It can be said that the Gibbs model works better on a larger database compared to NMF, which manages to present topics that are better humanly discernible, especially on databases that contain a small number of documents. In that sense, each of these models can adapt to different problems.

REFERENCES

1. Onan, A., Korukoglu, S., & Bulut, H. (2016). LDA-based Topic Modelling in Text Sentiment Classification: An Empirical Analysis. *Int. J. Comput. Linguistics Appl.*, *7*(1), 101-119.

2. Darling, W. M. (2011, December). A theoretical and practical implementation tutorial on topic modeling and gibbs sampling. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies* (pp. 642-647).

3. Tong, Z., & Zhang, H. (2016, May). A text mining research based on LDA topic modelling. In *International Conference on Computer Science, Engineering and Information Technology* (pp. 201-210).

4. Griffiths, T. L., & Steyvers, M. (2004). Finding scientific topics. *Proceedings of the National academy of Sciences*, *101*(suppl 1), 5228-5235.

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

6. Resnik, P., & Hardisty, E. (2010). *Gibbs sampling for the uninitiated*. Maryland Univ College Park Inst for Advanced Computer Studies.

7. Suri, P., & Roy, N. R. (2017, February). Comparison between LDA & NMF for event-detection from large text stream data. In *2017 3rd International Conference on Computational Intelligence & Communication Technology (CICT)* (pp. 1-5). IEEE.

8. M'sik, B., & Casablanca, B. M. (2020). Topic modeling coherence: A comparative study between lda and nmf models using covid'19 corpus. *International Journal*, *9*(4).

9. *A.M. Johansen, in International Encyclopedia of Education (Third Edition), 2010, Pages 245-252*

10. *J.L. Tobias, in Encyclopedia of Health Economics, 2014, Pages 146-154*

11. *D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent Dirichlet allocation. Journal of Machine Learning Research, 3:993--1022, 2003.*

12. *TomasHrycej , in Artificial Intelligence Volume 46, Issue 3, December 1990, Pages 351-363*

13. *KDD '08: Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data miningAugust 2008 Pages 569–577*

14. *Dave Binkley, in Understanding LDA for Software Engineering. Loyola University Maryland Baltimore MD, 21210, USA*

15. *ICML'14: Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32June 2014 Pages I-190–I-198*

16. *Michael Röder, Exploring the Space of Topic Coherence Measures*

17. *Hoyer PATRIKHOYER, P. O. (2004). Non-negative Matrix Factorization with Sparseness Constraints. In Journal of Machine Learning Research (Vol. 5).*

18. *O'Callaghan, D., Greene, D., Carthy, J., & Cunningham, P. (2015). An analysis of the coherence of descriptors in topic modeling. Expert Systems with Applications, 42(13), 5645–5657*

19. *Xu, W., Liu, X., & Gong, Y. (2001). Document Clustering Based On Non-negative Matrix Factorization*