Improved Parsimonious Topic Modeling Based on the Bayesian Information Criterion

Hang Wang and David Miller *

Electrical Engineering and Computer Science Department, The Pennsylvania State University, State College, PA 16802, USA; hzw81@psu.edu
* Correspondence: djm25@psu.edu

Received: 14 February 2020; Accepted: 9 March 2020; Published: 12 March 2020

Abstract: In a previous work, a parsimonious topic model (PTM) was proposed for text corpora. In that work, unlike LDA, the modeling determined a subset of salient words for each topic, with topic-specific probabilities, with the rest of the words in the dictionary explained by a universal shared model. Further, in LDA all topics are in principle present in every document. In contrast, PTM gives sparse topic representation, determining the (small) subset of relevant topics for each document. A customized Bayesian information criterion (BIC) was derived, balancing model complexity and goodness of fit, with the BIC minimized to jointly determine the entire model—the topic-specific words, document-specific topics, all model parameter values, and the total number of topics—in a wholly unsupervised fashion. In the present work, several important modeling and algorithm (parameter learning) extensions of PTM are proposed. First, we modify the BIC objective function using a lossless coding scheme with low modeling cost for describing words that are non-salient for all topics—such words are essentially identified as wholly noisy/uninformative. This approach increases the PTM’s model sparsity, which also allows model selection of more topics and with lower BIC cost than the original PTM. Second, in the original PTM model learning strategy, word switches were updated sequentially, which is myopic and susceptible to finding poor locally optimal solutions. Here, instead, we jointly optimize all the switches that correspond to the same word (across topics). This approach jointly optimizes many more parameters at each step than the original PTM, which in principle should be less susceptible to finding poor local minima. Results on several document data sets show that our proposed method outperformed the original PTM model with respect to multiple performance measures, and gave a sparser topic model representation than the original PTM.

Keywords: topic model; Bayesian information criterion; expectation maximization algorithm; medical abstracts

1. Introduction

Topic modeling [1] is a type of statistical modeling for finding the “topics” that occur in a collection of documents. The latent Dirichlet allocation (LDA) [2] is one of the well-known topic models. The LDA topic model assumes that each topic is a probability mass function defined over the given vocabulary, and for each of the documents, every word follows a document specific mixture over the topics. In an improvement of LDA called parsimonious topic modeling (PTM) [3], two shortcomings of LDA are discussed. First, all words have their own probability parameters under every topic in LDA; this strategy uses a huge number of parameters, with the model potentially prone to overfitting. Second, in LDA, every topic is assumed to be present in every document, with a non-zero probability. However, PTM gives a sparser description in the two aspects mentioned above. First, some words are not topic specific for certain topics—they use a universal shared model. Second, in each document, only some of the topics occur with a non-zero probability.
The Bayesian information criterion (BIC) [4,5] is a widely used criterion for model selection. There are two parts in the negative logarithm of the Bayesian marginal likelihood: the likelihood of the data and the model complexity cost. So, we can use BIC to balance data fitting goodness and model complexity. The BIC cost function derived for PTM [3] improves over “vanilla” BIC in two aspects. First, the proposed BIC has differentiated cost terms based on different effective sample sizes for different types of parameters; second, PTM introduces a shared feature representation to decrease the effective feature dimensionality of a topic. Our contribution here over PTM [3] is that we give a cheaper expression to describe wholly uninformative words, which thus encourages an even sparser model. Inspired by the intuition that in text corpora there are a large proportion of words that are not related to any topics, we implemented an optimization method to jointly optimize all the parameters related to a single word under each topic, to encourage the possibility that all topics choose to use a shared (uninformative) representation of a given word. Our work improves on the PTM model in encouraging a sparser and more reasonable representation of topics.

Our model solves both the unsupervised feature selection and model order (number of topics) selection problems. First we fix the hyper-parameter (number of topics) to a large number, and we get an optimized structure of the model by determining the topic-specific words under each topic and which topics occur in each document. Then, we change (reduce) the number of topics (hyper-parameter) by removing some topics. For each value of this hyper-parameter we train the model and compute BIC, and the optimized hyper-parameter (chosen model order) is the one with the lowest BIC value.

2. Notation

Suppose a corpus consists of \( D \) documents and \( N \) unique words, with \( d \in \{1, 2, ..., D\} \) and \( n \in \{1, 2, ..., N\} \) the document and word indices, respectively, and with unique topic indexed by \( j \in \{1, 2, ..., M\} \), \( M \) being the total number of topics (model order). Here are some different definitions used in this paper:

- \( L_d \) is the number of unique words in document \( d \).
- \( w_{id} \in \{1, 2, ..., N\} \), \( i = 1, ..., L_d \) is the \( i \)-th word in document \( d \).
- \( v_{jd} \in \{0, 1\} \) is the topic “switch”; it indicates whether topic \( j \) is present in document \( d \).
- Topic \( j \) is present in document \( d \) if \( v_{jd} = 1 \); otherwise \( v_{jd} = 0 \).
- \( M_d \equiv \sum_{j=1}^{M} v_{jd} \in \{1, 2, ..., M\} \) is the number of topics present in document \( d \).
- \( \alpha_{jd} \) is the proportion for topic \( j \) in document \( d \).

In each topic:

- \( \beta_{jn} \) is the topic-specific probability of word \( n \) under topic \( j \).
- \( \beta_{0n} \) is the shared probability of word \( n \).
- \( u_{jn} \in \{0, 1\} \) indicates whether \( (u_{jn} = 1) \) or not \( (u_{jn} = 0) \) word \( n \) is topic-specific under topic \( j \).
- \( N_j \equiv \sum_{n=1}^{N} u_{jn} \) is the total number of topic-specific words under topic \( j \).
- \( \bar{L}_j \equiv \sum_{d=1}^{L} v_{jd} \) is the sum of the length of the documents for which topic \( j \) is present.

3. Methodology

3.1. “Bag of Words” Model

A bag of words model [6] is commonly used in document classification where the count of each word is used as a feature for class discrimination. Using the bag of words model, we can transform the text corpora into a feature matrix, where each row vector \( x = (x_1, x_2, \ldots, x_D) \) in the matrix is a bag for each document, with the length of the vector the total number of unique words in the whole text corpora. In the row vector, each position represents a single word, and the value in that position is the number of times this word occurs in the document.
3.2. Parsimonious Topic Model (PTM)

We first introduce PTM’s data generation method.

1. For each document \(d = 1, 2, ..., D\)
2. For each word \(i = 1, 2, ..., L_d\)
   
   - Randomly select a topic based on the probability mass function (pmf) \(\{\alpha_{jd}, j = 1, 2, ..., M\}\).
   - Given the selected topic \(j\), randomly generate the \(i\)-th word based on the topic’s pmf over the word space \(\{\beta_{jn}, n = 1, 2, ..., N\}\).

Here \(v_{jd}\) is the topic switch that indicates whether topic \(j\) is present in document \(d\). If \(v_{jd} = 1\), it means that topic \(j\) is present in document \(d\) and \(\alpha_{jd}\) is treated as a model parameter. \(\beta_{jn}\) and \(\beta_{0n}\) are the topic-specific probability of word \(n\) under topic \(j\) and the shared probability of word \(n\), respectively.

Based on the above data generation, we can get the data likelihood of a document dataset \(\chi\) under our model \((H, \Theta)\):

\[
p(\chi|H, \Theta) = \prod_{d=1}^{D} \prod_{i=1}^{L_d} \prod_{j=1}^{M} [\alpha_{jd} v_{jd} \beta_{jn}^{u_{jn}} \beta_{0n}^{1-u_{jn}}].
\] (1)

Here \(u_{jn}\) is the word switch that indicates if word \(n\) is topic-specific under topic \(j\). The model structure parameters, denoted by \(H\{v, u, M\}\), consist of two kinds of switches and the number of topics, \(M\) (model order). Likewise, the model parameters, given a fixed model structure, are denoted by \(\Theta = \{\{\alpha_j\}, \{\beta_{jn}\}, \{\beta_{0n}\}\}\). The model structure together with the model parameters constitutes the PTM model. In PTM the parameters are constrained by the following two conditions:

First, \(\sum_{j=1}^{M} \alpha_{jd} v_{jd} = 1, \forall d\).

Additionally, the word probability parameters \(\{\beta_{jn}, n = 1, ..., N\}\) and \(\{\beta_{0n}, n = 1, ..., N\}\) must satisfy a pmf constraint for each topic:

\[
\sum_{n=1}^{N} (u_{jn} \beta_{jn} + (1-u_{jn}) \beta_{0n}) = 1, \forall j.
\] (3)

Based on the PTM described above, we must determine the model parameters \(\Theta\) and the model structure, \(H\). Assuming the model structure is known, we can estimate the model parameters using the expectation maximization (EM) algorithm. By introducing, as hidden data, random variables that indicate which topic generates each word in each document, we can compute the expected complete data log likelihood and maximize it subject to the two constraints mentioned above. The model selection is more complicated. We need to derive a BIC cost function to balance the model complexity and the data likelihood. For the PTM model a generalized expectation maximization (GEM) [7,8] algorithm was proposed to update the model parameters \(\Theta\) and the model structure \(H\) iteratively. In the following section, we give a derivation of BIC and the GEM algorithm for our modified PTM model.

3.3. Derivation of PTM-Customized Bayesian Information Criterion (BIC)

In this section we derive our BIC objective function, which generalizes the PTM BIC objective [3]. A naive BIC objective has the following form:

\[
BIC = K \log(n) - 2 \log(L).
\] (4)
Here \( \hat{L} \) is the maximized value of the data likelihood of the model with structure \( H \), that is, \( \hat{L} = p(D|H, \hat{\Theta}) \), where \( \hat{\Theta} \) is the collection of parameters that maximize the likelihood function.

Additionally, \( n \) is the number of data points (documents) in the dataset \( \chi \) and \( K \) is the number of free parameters in the model.

However, the Laplace approximation used in deriving this BIC form is only valid under the assumption that the feature space is far smaller than the sample size, and for our topic model, the feature space (word dictionary) is quite large in practice. Moreover, in the naive BIC form, all the parameters incur the same description length penalty (the log of the sample size), but in PTM different types of parameters contribute unequally to the model complexity. So, a new customized BIC is derived for PTM.

The Bayesian approach to model selection is to maximize the posterior probability of the model \( H \) given the dataset \( \chi \). When applying Bayes’ theorem to calculate the posterior probability, we get:

\[
p(H|\chi) = \frac{p(\chi|H)p(H)}{p(\chi)}. \tag{5}\]

Here we define:

\[
I = p(\chi|H) = \int p(\chi|H, \Theta)p(\Theta|H)d\Theta,
\]

where \( p(\Theta|H) \) is the prior distribution of the parameters given the model structure \( H \). Then, we need to use Laplace’s method to approximate \( I \), given the knowledge that for large sample size \( p(\chi|H, \Theta)p(\Theta|H) \) peaks around the maximum point (posterior mode \( \hat{\Theta} \)). We can rewrite \( I \) as:

\[
I = p(\chi|H) = \int \exp(\log(p(\chi|H, \Theta)p(\Theta|H)))d\Theta. \tag{7}\]

We can now expand \( \log(p(\chi|H, \Theta)p(\Theta|H)) \) around the posterior mode \( \hat{\Theta} \) using a Taylor series expansion.

\[
\log(p(\chi|H, \Theta)p(\Theta|H)) \approx \log(p(\chi|H, \hat{\Theta})p(\hat{\Theta}|H)) + (\Theta - \hat{\Theta})^T \nabla_\Theta Q|_{\hat{\Theta}} - \frac{1}{2}(\Theta - \hat{\Theta})^T \tilde{\Sigma}_\Theta (\Theta - \hat{\Theta}), \tag{8}\]

where \( Q \equiv \log(p(\chi|H, \Theta)p(\Theta|H)) \) and \( \tilde{\Sigma}_\Theta \equiv -\Sigma_\Theta \), where \( \Sigma_\Theta \) is the Hessian matrix (i.e., \( \Sigma_{ij} = \frac{\partial^2}{\partial \Theta_i \partial \Theta_j} Q|_{\Theta, \hat{\Theta}} \)).

Since \( Q \) attains its maximum at \( \hat{\Theta} \), \( \nabla_\Theta Q|_{\hat{\Theta}} = 0 \) and \( \tilde{\Sigma}_\Theta \equiv -\Sigma_\Theta \) is negative definite. We can thus approximate \( I \) as follows:

\[
I = p(\chi|H) \approx p(\chi|H, \hat{\Theta})p(\hat{\Theta}|H) \int e^{\frac{1}{2}(\Theta - \hat{\Theta})^T \Sigma_\Theta (\Theta - \hat{\Theta})}. \tag{9}\]

With the above form of the approximation, \( e^{\frac{1}{2}(\Theta - \hat{\Theta})^T \Sigma_\Theta (\Theta - \hat{\Theta})} \) is a scaled Gaussian distribution with mean \( \hat{\Theta} \) and covariance \( \Sigma_\Theta \). Thus:

\[
\int e^{\frac{1}{2}(\Theta - \hat{\Theta})^T \Sigma_\Theta (\Theta - \hat{\Theta})} = (2\pi)^{\frac{k}{2}} |\Sigma_\Theta|^{-\frac{1}{2}}, \tag{10}\]

where \( k \) is the number of parameters in \( \Theta \). So we have the approximation of \( I \):

\[
I = p(\chi|H) \approx p(\chi|H, \hat{\Theta})p(\hat{\Theta}|H)(2\pi)^{\frac{k}{2}} |\Sigma_\Theta|^{-\frac{1}{2}}. \tag{11}\]

BIC is the negative log model posterior: \( BIC = -\log(\hat{I}p(H)) \approx \frac{k}{2} \log(2\pi) + \frac{1}{2} \log(|\Sigma_\Theta|) - \log(p(\chi|H, \hat{\Theta})) - \log(p(\hat{\Theta}|H)) - \log(p(H)). \tag{12} \)
Note that \( p(\Theta|H) \), the prior of the parameters given the structure \( H \) can be assumed to be a uniform distribution (i.e., a constant). So, this term can be neglected. The \( \log(p(x|H, \Theta)) \) term is the data likelihood, and \( k \) is the total number of model parameters. Now we need to approximately calculate \( \frac{1}{2} \log(|\Sigma_\Theta|) \) and \( \log(p(H)) \).

To do so, we assume that \( \Sigma_\Theta \) is a diagonal matrix. We thus obtain:

\[
\frac{1}{2} \log(|\Sigma_\Theta|) \approx \frac{1}{2} \sum_{d=1}^{D} (M_d - 1) \log(L_d) + \frac{1}{2} \sum_{j=1}^{M} \sum_{d=1}^{D} u_{jd} \log(L_j) + \frac{1}{2} \sum_{j=1}^{D} \log(\sum_{j=1}^{M} L_j). \tag{13}
\]

The terms on the right represent the cost of the model parameters \( \{a_{jd}\}, \{\beta_{jn}\}, \) and \( \{\beta_{0n}\} \), respectively. Note that in the naive BIC form, each parameter pays the same cost \( \frac{1}{2} \log(\text{sample size}) \).

Here we instead use the effective sample size. The effective sample size of \( a_{jd} \) is \( L_d \), parameter \( \beta_{jn} \) has sample size \( L_j \), and the parameter \( \beta_{0n} \) has sample size \( \sum_{j=1}^{M} L_j \).

Another term to be estimated is \( \log(p(H)) = \log(p(v)) + \log(p(u)) \). For \( \log(p(v)) \), in each document \( \delta \), suppose the number of topics follows a uniform distribution, and the switch configuration also follows a uniform distribution over all \( \binom{M}{M_{\delta}} \) switch configurations. We then obtain:

\[- \log(p(v)) = D \log(M) + \sum_{d=1}^{D} \log(\binom{M}{M_{\delta}}). \tag{14}\]

For \( \log(p(u)) \) we propose here a probability model that can jointly estimate \( \log(p(u)) \) and the corresponding parameter cost of \( \beta_{jn}, \beta_{0n} \). For each word \( n \), we define three types of configurations of the word switches \( \{u_{jn}, j = 1, ..., M\} \): (1) each word is topic-specific (i.e., \( \sum_{j=1}^{M} u_{jn} = M \)); (2) all the words are not topic-specific (i.e., \( \sum_{j=1}^{M} u_{jn} = 0 \)); (3) some, but not all, components use the shared distribution (i.e., \( 0 < \sum_{j=1}^{M} u_{jn} < M \)). For cases 1 and 2, there is only one possible configuration of the word switches related to a word (all open or all closed), so the probability associated with this configuration is 1; for case 3 there are \( 2^M \) possible configurations. Assuming these are equally likely under this case \( \log(p(u)) = M \log 2 \). We can then estimate \( - \log(p(u)) \) plus the two terms \( \sum_{j=1}^{M} \sum_{d=1}^{D} u_{jd} \log(L_j) + \sum_{d=1}^{D} \log(\sum_{j=1}^{M} L_j) \) in \( \log(|\Sigma_\Theta|) \). That is, we have

\[
- \log(p(u)) + \frac{1}{2} \sum_{j=1}^{M} \sum_{d=1}^{D} u_{jd} \log(L_j) + \frac{1}{2} \sum_{d=1}^{D} \log(\sum_{j=1}^{M} L_j) \approx \sum_{n=1}^{N} \left( F_1(u_n) \frac{1}{2} \log(\sum_{j=1}^{M} L_j) + F_2(u_n) \frac{1}{2} \sum_{j=1}^{M} \log(L_j) \right) + F_3(u_n) \frac{1}{2} \log(\sum_{j=1}^{M} L_j) \tag{15}\]

Here, \( F_1(u_n) = \begin{cases} 1, & \text{if } \sum_{j=1}^{M} u_{jn} = 0 \\ 0, & \text{otherwise} \end{cases} \), \( F_2(u_n) = \begin{cases} 1, & \text{if } \sum_{j=1}^{M} u_{jn} = M \\ 0, & \text{otherwise} \end{cases} \), \( F_3(u_n) = \begin{cases} 1, & \text{if } \sum_{j=1}^{M} u_{jn} < M \\ 0, & \text{otherwise} \end{cases} \).

Based on the derivation above, the BIC cost function for our modified PTM model is:

\[
\text{BIC} = D \log(M) + \sum_{d=1}^{D} \log(\binom{M}{M_{\delta}}) + \frac{1}{2} \sum_{d=1}^{D} (M_d - 1) \log(L_d) - \frac{1}{2} \sum_{d=1}^{D} (M_d - 1) L_d \log(\frac{L_d}{2\pi}) - \log(p(D|H, \Theta)) - \sum_{n=1}^{N} \left( F_1(u_n) \frac{1}{2} \log(\sum_{j=1}^{M} L_j) + F_2(u_n) \frac{1}{2} \sum_{j=1}^{M} \log(L_j) + F_3(u_n) \frac{1}{2} \log(\sum_{j=1}^{M} L_j) \right). \tag{16}\]
3.4. Generalized Expectation Maximization (EM) Algorithm

The EM \[9\] algorithm is a popular method for maximizing the data log-likelihood. For unsupervised learning tasks, we only have the data points \(\chi\), but we do not have any “labels” for those data points, so during the maximum likelihood estimation (MLE) process we introduce “label” random variables which are called latent variables. The EM algorithm can be described as follows:

E-step: With the parameters fixed, we compute the expectation of the latent variables \(p(Z|\chi, \Theta)\), which gives the class information for each data point. Using the expectation of the latent variables we can compute the expectation of the complete data log-likelihood:

\[
E[L_c] = \sum_Z p(Z|D, \Theta) \log(p(D, Z|\Theta)). \tag{17}
\]

M-step: We update the parameters \(\Theta\) to find the maximum value of the expected complete data log-likelihood. By doing the E-step and M-step iteratively, the expected complete data log-likelihood strictly increases and typically converges to a local optimum of the incomplete data log-likelihood (or of the expected BIC cost function). However, note that in PTM there are not only the model parameters \(\Theta\) but also the model structure \(H\) over which we need to optimize. The original EM algorithm cannot be applied here because one cannot get jointly optimal closed form estimates of both the model parameters \(\Theta\) and the structure parameters \(H\) to maximize \(E[L_c]\). However, a generalized expectation maximization (GEM) \[7,8\] algorithm is proposed, which alternately jointly optimizes \(E[L_c]\) over \(\Theta\) and then over subsets of the structure parameters, \(H\) given fixed \(\Theta\).

Our GEM algorithm is specified for fixed model order, \(M\). First we introduce the hidden data \(Z\): \(Z_{id}\) is an \(M\)-dimensional binary vector, with a “1” indicating the topic of origin for the word \(w_{id}\). For example, if the element \(Z_{id}^{(j)} = 1\) and other elements of \(Z_{id}\) are all equal to zero, topic \(j\) is the topic of origin for word \(w_{id}\).

Our GEM algorithm strictly descends in the BIC cost function Equation (16). It consists of an E-step followed by an M-step that minimizes the expected BIC cost function. These steps are given as follows: In the E-step, first we compute the expectation of the hidden data \(Z\):

\[
p(Z_{id}^{(j)} = 1|w_{id}; \Theta, H) = \frac{p(w_{id}|Z_{id}^{(j)}, \Theta, H)p(Z_{id}^{(j)}|\Theta, H)}{\sum_{Z_{id}} p(w_{id}|Z_{id}^{(j)}, \Theta, H)p(Z_{id}^{(j)}|\Theta, H)}
\tag{18}

= \frac{\alpha_{id}^{\nu_{id}} \beta_{0n}^{1-u_{|w_{id}}} \beta_{1n}^{1-u_{|w_{id}}} \sum_{l=1}^M \alpha_{id}^{\nu_{id}} \beta_{0n}^{1-u_{|w_{id}}} \beta_{1n}^{1-u_{|w_{id}}} }{\sum_{l=1}^M \alpha_{id}^{\nu_{id}} \beta_{0n}^{1-u_{|w_{id}}} \beta_{1n}^{1-u_{|w_{id}}} }.
\]

With the expectation of the hidden data, we can compute the expected complete data log-likelihood using Equation (17). By replacing the \(\log(p(D|H, \Theta))\) term in BIC with the expected complete data log-likelihood(\(E[L_c]\)), we get the expected complete data BIC.

In the generalized M-step, based on the expectation of the complete data BIC we computed in the E-step, we update the model structure \(H\) and the model parameters \(\Theta\). First we optimize the model parameters given fixed model structure. Then we optimize the model structure given fixed model parameters, both steps taken to minimize the expected BIC. These steps are alternated until convergence.

When updating the model parameters, note that the only term in BIC that is related to the model parameters is the data likelihood term, so we can just maximize the expected complete data
log-likelihood computed in the E-step to choose the model parameters. Taking those two constraints (Equations (2) and (3)) into consideration, we have our Lagrangian objective function.

\[ L = \sum_{d=1}^{D} \sum_{i=1}^{L_d} (v_{id} \log(P(Z_{id}^{(j)} = 1|w_{id}; \Theta, H)(\log(\alpha_{jd}) + u_{jw_{id}} \log(\beta_{jw_{id}}) + (1 - u_{jw_{id}}) \log(\beta_0w_{id})))) - \sum_{d=1}^{D} \sum_{j=1}^{M} \lambda_{d}(\sum_{i=1}^{L_d} \alpha_{jd} v_{id} = 1) - \sum_{j=1}^{M} \sum_{i=1}^{L_d} (u_{jn}\beta_{jn} + (1 - u_{jn})\beta_0n) - 1, \]  

(19)

where \( \mu_j \) and \( \lambda_d \) are Lagrange multipliers. By computing the partial derivative of each parameter type and setting those derivatives to zero, we can get the optimized model parameters, satisfying necessary optimality conditions as:

\[ \alpha_{jd} = \frac{\sum_{i=1}^{L_d} p(Z_{id}^{(j)} = 1|w_{id}; \Theta, H)v_{id}}{\sum_{i=1}^{L_d} p(Z_{id}^{(j)} = 1|w_{id}; \Theta, H)v_{id}}, j = 1, ..., M, d = 1, ..., D, \]  

(20)

\[ \beta_{jn} = \frac{u_{jn} \sum_{d=1}^{D} \sum_{i=1}^{L_d} p(Z_{id}^{(j)} = 1|w_{id}; \Theta, H)v_{id}}{\mu_j}, j = 1, ..., M, n = 1, ..., N. \]  

(21)

We compute \( \mu_j \) by multiplying both sides of Equation (21) by \( u_{jn} \), summing over all \( n \), and applying the distribution constrains on topic \( j \). This gives:

\[ \mu_j = \frac{\sum_{n=1}^{N} u_{jn} \sum_{d=1}^{D} \sum_{i=1}^{L_d} p(Z_{id}^{(j)} = 1|w_{id}; \Theta, H)v_{id}}{1 - \sum_{j=1}^{M} (1 - u_{jn})\beta_0n}, \forall j. \]  

(22)

For the shared parameters, we only estimate them once via global frequency counts at initialization and hold them fixed during the GEM algorithm. That is, we set:

\[ \beta_{0n} = \frac{\sum_{d=1}^{D} \sum_{i=1}^{L_d} 1}{\sum_{d=1}^{D} L_d}, \forall n. \]  

(23)

When updating the model structure, we implement an iterative loop in which all the topic switches \( u \) are visited one by one. If the current change reduces BIC, we accept the change; otherwise we keep the switch unchanged. Note that in updating the word switches \( v \), we update all the word switches for a single word jointly to see if it is optimal to choose all the switches related to a single word to be closed (i.e., to specify that this word is completely uninformative). This process is repeated over all the switches until there is no decrease in BIC or until we reach a pre-defined maximum number of iterations. We first update the word switches \( u \) until convergence; then we update the topic switches \( v \) until convergence. Then we go back to the E-step. Note that when updating the word switches, for each single search of all the switches related to one word, we have three configurations. All switches are closed, all switches are open, or some are closed and some are open. We compute the minimized BIC for each configuration, and then choose the configuration that has the lowest BIC value.

3.5. Selecting the Model Order

The optimization process discussed above is under the assumption that the model order is known. Model order selection is based on applying the optimization process for different model orders, in a top-down fashion. We initialize the model with a specific number of topics (\( M_{\text{max}} \), chosen to upper bound the number of topics expected to be present in the corpus) and reduce the number by a predefined step \( \Delta \). For the model trained at each order, we remove the \( \Delta \) topics with the smallest mass. This process is applied iteratively until the predefined minimum order is reached. We then retain the model (and model order) with the smallest BIC cost.
For our model, the only “hyper-parameters” are $M_{\text{max}}$ and $\Delta$. We can expect the best performance by choosing $\Delta = 1$. For $M_{\text{max}}$, if this is set too small, it will underestimate the model order. If it is set very large, the learning and model order selection will require more computation. In principle, choosing any value of $M_{\text{max}}$ above the ground truth $M^*$ or the BIC minimizing $\hat{M}$ should be reasonable to find a good solution (the bottom line is that our method requires no true hyper-parameters—the only tradeoff is more computation by choosing $\Delta = 1$ and sufficiently large $M_{\text{max}}$).

4. Experiments and Results

In this section we compare the original PTM, LDA, and the new PTM method. All methods were used to solve the unsupervised density modeling problem, with no knowledge of class labels. Hence, the comparison of the methods is fair, and we evaluated multiple measures that assess the quality of the learned models as a function of the number of components in the model. None of the methods set hyper-parameters (except LDA, for which $M$ was set to the the maximum).

Performance measurements: BIC was compared between the PTM model and our revised PTM model. For both models, the held-out log-likelihood and the class label purity were also compared. In [3] the performance of the PTM model and the LDA model were compared, and PTM was found to outperform LDA. Here we only include the label purity of the LDA model on different datasets.

When computing the held-out likelihood, we used the method described in [10,11] to compare model fitness on a held-out test set. We divided the documents in the test set into two parts: the observed part and the held-out part. First we computed the topic proportions based on the observed part, then we computed the held-out log-likelihood based on the held-out part:

$$\sum_{d=1}^{D_{\text{test}}} \sum_{i=1}^{L_{\text{held-out}}} \log \left( \sum_{j=1}^{M} E_q[\alpha_{jd}]E_q[\beta_{jn}] \right).$$

(24)

In our model $E_q[\alpha_{jd}]$ is directly the topic proportions $\alpha_{jd}$. $E_q[\beta_{jn}]$ is $u_{jn}\beta_{jn} + (1 - u_{jn})\beta_{0n}$.

We evaluated PTM, modified-PTM, and LDA on three datasets as next discussed.

4.1. Reuters-21578

The Reuters-21578 dataset is a collection of documents from Reuters news in 1987. There are in total 7674 documents from 35 categories. After stemming and stop word removal, there were 17,387 unique words. There were 5485 documents in the training set and 2189 documents in the test set.

4.2. 20-Newsgroups

The 20-Newsgroup dataset is a collection of 18,821 newsgroup documents from 20 classes. There were 53,976 words after stemming and stop word removal. It was split into a 11,293-document training set and a 7528-document test set.

4.3. Ohsumed

The Ohsumed dataset includes medical abstracts from the MeSH categories of the year 1991. It consists of 34,389 documents, each assigned to one or multiple labels of the 23 MeSH disease categories. Each document has on average 2.26 labels. The dataset was divided into 24,218 training and 10,171 test documents. There were 12,072 unique words in the corpus after applying standard stop word removal and stemming.

Note that for the Ohsumed dataset, each document may be associated with multiple labels. We computed the label purity for this dataset as follows: We first associated to each topic a multinomial distribution on the class labels. We learned these label distributions for each topic by frequency counting over the ground truth class labels of all documents, weighted by topic proportions:
\[ p_j(c) = \frac{\sum_{d=1}^{D} \sum_{i=1: l_{id}=c}^{|C_d|} \alpha_{jd} v_{jd}}{\sum_{d=1}^{D} \sum_{i=1}^{|C_d|} \alpha_{jd} v_{jd}}, \forall j, c, \] (25)

where \( p_j(c) \) is the class proportion for class \( c \), for topic \( j \). Here \( l_{id} \) is the \( i \)-th class label in document \( d \) and \( |C_d| \) is the number of class labels in document \( d \). For labeling a text document, we then computed the probability of each class label based on the topic proportions in that document, that is, \( \sum_{j=1}^{M} \alpha_{jd} p_j(c) \), and assigned the labels that had probability higher than a threshold value \( T \):

\[ \hat{C}_d = \{ c : \sum_{j=1}^{M} \alpha_{jd} p_j(c) > T \}. \] (26)

We changed the threshold \( T \) and measured the area under the precision/recall curve (AUC). Note that here precision is the number of true discovered labels divided by the total number of ground-truth labels. Recall is the number of correctly classified labels divided by the total number of labels assigned to documents by our classifier.

4.4. Discussion

Our results on the three text corpora are shown in Figures 1–3. Here we only include comparison with LDA on label purity. The comparison of LDA with the original PTM on held-out log-likelihood is in [3], and demonstrates that original PTM gives better results. Here, we see that the new PTM method convincingly outperformed original PTM (and hence LDA). Previous work showed that the original PTM method outperformed LDA with respect to two performance measurements: held-out log-likelihood and label purity [3]. This was attributed to the fact that PTM models are sparser than LDA models. However, the modified-PTM method dominated the original PTM (and LDA) with respect to all three of these measures, on all three datasets. Note in particular the large gain in held-out log-likelihood on Reuters-21578, in all measures on 20-Newsgroups, and with respect to BIC and held-out log-likelihood on Ohsumed.

![Figure 1](image.png)

**Figure 1.** Performance comparison between the original parsimonious topic model (PTM), our modified PTM, and latent Dirichlet allocation (LDA) on the Reuters-21578 dataset. The measurements were the Bayesian information criterion (BIC), held-out log-likelihood (HL), label purity (LP), and the proportion of words with all-closed word switches (All-closed). LDA is only shown for label purity.
**Figure 2.** Performance comparison between the original PTM, our modified PTM, and LDA on the 20-Newsgroup dataset. The measurements were BIC, held-out log-likelihood (HL), label purity (LP), and the proportion of the words with all-closed word switches (All-closed). LDA is only shown for label purity.

**Figure 3.** Performance comparison between the original PTM, our modified PTM, and LDA on the Ohsumed dataset. The measurements were BIC, held-out log-likelihood (HL), AUC, and the proportion of the words with all-closed word switches (All-closed). LDA is only shown for label purity.

The modified PTM minimizes BIC by selecting a much larger (richer) set of topics than the original PTM. This is achieved by the low description length penalty associated with making words completely uninformative (all closed), which leads to many more words being deemed completely uninformative compared to the original PTM. This is a key advantage of the new PTM method. The original PTM method offers no incentive within the BIC cost function to decide that a word is wholly uninformative about (irrelevant to) the topics that are present in the document. The new method provides substantial
incentive for such determination with great reduction in model description complexity gained by such a choice. This allows the model to “afford” having many more topics than in the original PTM method. Especially, close to 80% of words were closed under modified PTM compared to approximately 40% for PTM for Reuters (at the selected model order), approximately 80% compared to approximately 50% are closed for 20-Newsgroups, and approximately 70% compared to approximately 30% were closed for Ohsumed. That is, modified PTM chose higher model orders (number of topics) with fewer topic-specific words and more wholly uninformative words than the original PTM. Choosing more topics was seen to yield better performance for several measures (label purity and held-out log-likelihood).

4.5. Computational Complexity

In both the PTM model and our modified model, we need to optimize over the parameters $\Theta$ and the model structure parameters $\{v, u\}$.

Consider learning the parameters $\Theta$ when $M$ is fixed and with $D$ documents. The computational complexity of the PTM model and our modified model are both $O(M_D D L_D)$, where $M_D$ and $L_D$ are the number of topics present in a document and the length of a document, respectively. In LDA all topics are present in all documents, and the computational complexity is $O(M D L_D)$—that is, the total computational complexity of the LDA model. However, in our model and the PTM model we also need to optimize over the model structure parameters $\{v, u\}$.

For the word switches $\{u_{jn}\}$, in the PTM model, updating them involves an iterative loop over all $N$ words in $M$ topics. Thus, the complexity is $O(M N)$. In our modified PTM model for each of the $N$ words, we need to check two more cases—that is, case 1) and case 2) discussed before Equation (15). So, the computational complexity is $O((M + 2)N) = O(M N)$.

For the topic switches $\{v_{jd}\}$, both our model and the PTM model need computations of order $O(M D L_D)$ to update each switch $v_{jd}$. We have in total $M D$ switches to be updated. So the computational complexity of updating the topic switches is $O(M M_D D L_D)$ for both PTM and our modified model. The computational complexity comparison is shown in Table 1.

We recorded the execution time for the PTM model and the revised PTM model on each dataset (in Table 1). We ran the experiment on a machine with an Intel Core i5, 2.3 GHz processor. The execution time of our method was slightly less than that of the PTM model, which may be because in our modified model, for the word switches related to a single word, when it comes to the point that all the switches are closed, it is very likely that no update will be done in the future iterations. We did an experiment with the LDA model using the Python package, but our modified PTM model is implemented in C. So, the comparison between our model and LDA may be unfair. A comparison of the execution time between LDA and PTM models is reported in [3].

|                  | Modified PTM       | PTM       | LDA       |
|------------------|---------------------|-----------|-----------|
| Complexity       | $O(M M_D D L_D + M N)$ | $O(M M_D D L_D + M N)$ | $O(M D L_D)$ |
| Execution Time (min) | 20-Newsgroup 387 493  | 39 44    | 424 659   |
|                  | Reuters-21578       |           |           |
|                  | Ohsumed             |           |           |
4.6. Limitations of Our Work

One limitation of our model is that the computational complexity is higher than for the LDA model. Another limitation is the highly non-convex optimization, with discrete and continuous parameters and no guarantee of finding the global minimum.

5. Conclusions

In this paper, we proposed two improvements on the PTM method. One is improving the modeling by giving a cheaper expression to describe the wholly uninformative words. The other is improving the learning—we optimized all the parameters related to a single word under each topic, which encourages a sparser model. Our improvements led to consistent and substantial gains in modeling accuracy with respect to multiple performance measures, compared with both the original PTM method and the well-known LDA model. While we demonstrated significant gains on three document corpora, there is increased computational complexity for our method compared to LDA. Future work could aim to develop a fully Bayesian version of our approach, based on posteriors on topic and word switches, rather than on deterministic (binary) switch parameters. Future work could also investigate extensions that model word order dependency, but somehow in a low-complexity fashion.

Author Contributions: Conceptualization, D.M.; Data curation, H.W.; Funding acquisition, D.M.; Methodology, D.M.; Project administration, D.M.; Software, H.W.; Supervision, D.M.; Validation, H.W.; Writing – review & editing, D.M. All authors have read and approved the final published version.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

Reference
1. Wallach, H.M. Topic modeling: beyond bag-of-words. In Proceedings of the 23rd International Conference on Machine Learning, Pittsburgh, PA, USA, 25–29 June 2006; pp. 977–984. Available online: http://dirichlet.net/pdf/wallach06topic.pdf (accessed on 10 March 2020).
2. Blei, D.M.; Ng, A.Y.; Jordan, M.I. Latent dirichlet allocation. J. Mach. Learn. Res. 2003, 3, 993–1022.
3. Soleimani, H.; Miller, D.J. Parsimonious topic models with salient word discovery. IEEE Trans. Knowl. Data Eng. 2014, 27, 824–837. [CrossRef]
4. Schwarz, G. Estimating the dimension of a model. Ann. Stat. 1978, 6, 461–464. [CrossRef]
5. Bhat, H.S.; Kumar, N. On the derivation of the Bayesian Information Criterion. Available online: https://faculty.ucmerced.edu/hbhat/BICderivation.pdf (accessed on 10 March 2020).
6. Zhang, Y.; Jin, R.; Zhou, Z.H. Understanding bag-of-words model: A statistical framework. Int. J. Mach. Learn. Cybern. 2010, 1, 43–52. [CrossRef]
7. Dempster, A.P.; Laird, N.M.; Rubin, D.B. Maximum likelihood from incomplete data via the EM algorithm. J. R. Stat. Soc. Series B. Stat. Methodol. 1977, 39, 1–22.
8. Meng, X.L.; Van Dyk, D. The EM algorithm—An old folk-song sung to a fast new tune. J. R. Stat. Soc. Ser. B. Stat. Methodol. 1997, 59, 511–567. [CrossRef]
9. Do, C.B.; Batzoglou, S. What is the expectation maximization algorithm? Nat. Biotechnol. 2008, 26, 897. [CrossRef] [PubMed]
10. Hoffman, M.D.; Blei, D.M.; Wang, C.; Paisley, J. Stochastic variational inference. J. Mach. Learn. Res. 2013, 14, 1303–1347.
11. Teh, Y.W.; Newman, D.; Welling, M. A collapsed variational Bayesian inference algorithm for latent Dirichlet allocation. In Proceedings of the Advances in Neural Information Processing Systems, Vancouver, BC, Canada, 3–6 December 2007; pp. 1353–1360.

© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).