Topic Modeling via Full Dependence Mixtures

Dan Fisher [1] **Mark Kozdoba** [1] **Shie Mannor** [1 2]

Abstract

In this paper we introduce a new approach to topic modelling that scales to large datasets by using a compact representation of the data and by leveraging the GPU architecture. In this approach, topics are learned directly from the co-occurrence data of the corpus. In particular, we introduce a novel mixture model which we term the Full Dependence Mixture (FDM) model. FDMs model second moment under general generative assumptions on the data. While there is previous work on topic modeling using second moments, we develop a direct stochastic optimization procedure for fitting an FDM with a single Kullback Leibler objective. Moment methods in general have the benefit that an iteration no longer needs to scale with the size of the corpus. Our approach allows us to leverage standard optimizers and GPUs for the problem of topic modeling. In particular, we evaluate the approach on three large datasets, NeurIPS papers, a Twitter corpus, and full English Wikipedia, with a large number of topics, and show that the approach performs comparably or better than the the standard benchmarks.

1. Introduction

A topic model is a probabilistic model of joint distribution in the data, that is typically used as a dimensionality reduction technique in a variety of applications, such as for instance text mining, information retrieval and recommender systems. In this paper we concentrate on topic models in text data. Perhaps the most widely used topic model for text is the Latent Dirichlet Allocation (LDA) model, (Blei et al., 2002). LDA is a fully generative probabilistic model, and is typically learned through a Bayesian approach — by sampling the parameter distribution given the data, via Collapsed Gibbs samplers (Griffiths & Steyvers, 2004), (Steyvers & Griffiths, 2007), variational methods, (Blei et al., 2002), (Foulds et al., 2013), (Hoffman et al., 2013), or other related approaches. A typical learning procedure for Bayesian methods involves an iteration over the entire corpus, where a topic assignment is sampled per each token or document in the corpus. In order to apply these methods to large corpora, a variety of optimized procedures were developed, where speed improvement is achieved either via parallelization or via more economic sampling scheme, per token. An additional level of complexity is added by the fact that LDA has two hyperparameters, the Dirichlet priors of the token distribution in topics, and topic distribution in documents. This may complicate the application of the methods since the choice of parameters may influence the results even for relatively large data sizes. A detailed discussion of LDA optimization is given in Section 2.

In this paper we propose an alternative approach to topic modeling, based on principles that are principally different from the standard LDA optimization techniques. We show that using this approach it is possible to analyze large datasets on a single workstation with a GPU, and to obtain results that are comparable or better than the standard benchmarks. A reference implementation of the algorithm is available at `https://github.com/fisherd3/fdm`.

In the rest of this section we introduce some necessary notation, describe our model and the related loss function, discuss the optimization procedure for this loss, and overview the experimental results of the paper.

1.1. Method Description

We assume that the text data was generated from a pLSA (Probabilistic Latent Semantic Allocation) probability model , (Hofmann, 1999), as follows: Denote by $\mathcal{X}$ the set of distinct tokens in the corpus, and suppose that we are given T topics, μ_t, $t \leq T$, where each topic is a probability distribution on the set of tokens $\mathcal{X}$. Then the pLSA assumption is that each document d is generated by independently sampling tokens from a mixture of topics, denoted ν_d:

$$\nu_d = \sum_t \theta_d(t)\mu_t, \tag{1}$$

where $\theta_d(t) \geq 0$ and $\sum_t \theta_d(t) = 1$ for every document d. Note that we do not specify the generative model for θ_d. In

[1]Technion, Israel Institute of Technology [2]NVIDIA Research. Correspondence to: <dan.fisher.tech@gmail.com>, <markk@technion.ac.il>, <shie@ee.technion.ac.il>.

Proceedings of the 37^{th} International Conference on Machine Learning, Online, PMLR 119, 2020.

this sense, pLSA is a semi-generative model, and is more general than LDA.

Next, for every document in the corpus we construct its token co-occurrence probability matrix, and we take the co-occurrence probability matrix of the corpus to be the average of the document matrices. Let $N = |\mathcal{X}|$ be the dictionary size - the number of distinct tokens in the corpus. Then the co-occurrence matrix $\widehat{M}$ of the corpus is an $N \times N$ matrix, with non-negative entries that sum to 1. Suppose that one performs the following experiment: Sample a document from a corpus at random, and then sample two tokens independently from the document. Then $\widehat{M}_{u,v}$ is the probability to observe the pair u, v in this experiment (up to a small modification, full details are given in Section 3).

Now, if one assumes the pLSA model of the corpus, then it can be shown that the expectation of $\widehat{M}$ should be of the form

$$M_{u,v}(\mu, \alpha) = \sum_{i,j=1}^{T} \alpha_{i,j} \mu_i(u) \mu_j(v), \tag{2}$$

where μ_i are the topics and $\alpha_{i,j} \geq 0$, $\alpha_{i,j} = \alpha_{j,i}$, and $\sum_{i,j} \alpha_{i,j} = 1$ represent the corpus level topic-topic correlations. We refer to the matrices of the form (2) as Full Dependence Mixture (FDM) matrices. This is due to the analogy with standard multinomial mixture models (also known as categorical mixture models), which can be represented in the form (2) but with α restricted to be a diagonal matrix. Multinomial mixture models correspond to the special case where each document contains samples from only one topic, where the topic may be different for different documents. Equivalently, in multinomial mixtures, each θ_d is 0 on all but one coordinates.

In this paper, we consider a set of topics μ_t to be a good fit for the data if there are some correlation coefficients α such that $M(\mu, \alpha)$ is close to $\widehat{M}$, the FDM generated from the data. Specifically, we define the loss by

$$L(\mu, \alpha) = -\sum_{u,v \in \mathcal{X}} \widehat{M}_{u,v} \log M(\mu, \alpha)_{u,v}, \tag{3}$$

and we are interested in minimizing L over all μ, α. Clearly, minimizing L is equivalent to minimizing the Kullback-Leibler divergence between $M(\mu, \alpha)$ and M, viewed as probability distributions over $\mathcal{X} \times \mathcal{X}$.

To gain basic intuition as to why $M(\mu, \alpha)$ that approximates well the empirical matrix $\widehat{M}$ should yield good topics, it is useful to consider again the simple case of the multinomial mixture, and moreover, the specific case where the topics are disjoint. In this case, the matrix $\widehat{M}$ will be block diagonal (up to a reordering of the dictionary), with disjoint blocks that correspond to the topics. Thus, provided enough samples d, the topics can be easily read off from the matrix.

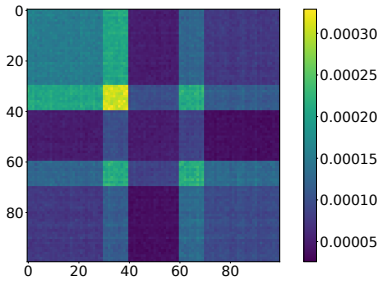


Figure 1. An $\widehat{M}$ matrix example.

An example of a more complicated matrix $\widehat{M}$ is shown in Figure 1. Here the ground set is $\mathcal{X} = \{1, \ldots, 100\}$, and the topics are uniform on intervals $[1, 40]$,$[30, 70]$, and $[60, 100]$ respectively. The documents were generated from the mixture (1), with θ_d sampled from a non-uniform Dirichlet, $\theta_d \sim Dir((2, 1, 1.5))$, which make the topics appear with different frequencies in the data. Although here the relation between the topics and the matrix is more involved, one can still see that the topics could be traceable from the matrix.

In Section 4 we show the asymptotic consistency of the loss (3) for topics which satisfy the classical anchor words ((Donoho & Stodden, 2003), (Arora et al., 2013)) assumption. That is, if the topics satisfy the anchor words assumption, then given enough samples the topics can be uniquely reconstructed from the matrix $\widehat{M}$ by minimizing the loss (3). The anchor words assumption roughly states that each topic has a word that is unique to that particular topic. Note that this word does not have to appear in each document containing the topic, and may in fact have a relatively low probability inside the topic itself. It is known, (Arora et al., 2013), and easy to verify, that natural topics, such as topics produced by learning LDA, do satisfy the anchor words assumption. We refer to Section 4 for further details.

The advantage of using the cost $L(\mu, \alpha)$ is that it depends on the corpus *only* through the matrix $\widehat{M}$. Therefore, the size of the corpus does not enter the optimization problem directly, and we are dealing with a fixed size, $N \times N$ problem. This is a general feature of reconstruction through moments approaches, such as for instance (Arora et al., 2013), (Anandkumar et al., 2012). In particular, the number of variables for the optimization is $TN + T^2$, in contrast to variational or Gibbs sampler based methods, which either have T variables for every document, or have a single variable for every token in the corpus, respectively.

1.2. Optimization of the Objective $L(\mu, \alpha)$.

For smaller problems, one may directly optimize the objective (3) using gradient descent methods. However, note that

if one computes $L(\mu, \alpha)$ directly, then one has to compute $M(\mu, \alpha)$, which is a sum of T^2 matrices of size N^2. Indeed, denote by $M^{i,j}$ the $N \times N$ matrix such that its u, v entry is $M^{i,j}_{u,v} = \mu_i(u) \cdot \mu_j(v)$. Then $M(\mu, \alpha) = \sum_{i,j \leq T} \alpha_{i,j} M^{i,j}$. On standard GPU computing architectures, all T^2 of the matrices will have to be in memory simultaneously, which is prohibitive even for moderate values of N, T. To resolve this issue, we reformulate the optimization of L as a stochastic optimization problem in u, v. To this end, note that L is an expectation of the term $\log M(\mu, \alpha)_{u,v}$ over pairs of tokens (u, v), sampled from $\widehat{M}$, where $\widehat{M}$ is viewed as a probability distribution over $\mathcal{X} \times \mathcal{X}$. Formally,

$$L(\mu, \alpha) = \mathbb{E}_{(u,v) \sim \widehat{M}} \log M(\mu, \alpha)_{u,v}. \tag{4}$$

Therefore, given (u, v), one only has to compute the gradient of $M(\mu, \alpha)_{u,v}$, rather than full $M(\mu, \alpha)$ at μ, α – which is a much smaller optimization problem, of size $O(T^2)$, and this can be done for moderate (u, v) batch sizes. This approach makes the optimization of $L(\mu, \alpha)$ practically feasible. The full algorithm, given as Algorithm 2 below, is discussed in detail in Section 3. Note that this approach differs from the standard stochastic gradient descent flow on the GPU, where the batches consist of data samples (documents in the case of text data). Instead, here the data is already summarized as $\widehat{M}$, and the batches consist of pairs of tokens (u, v) that we sample actively from $\widehat{M}$.

1.3. Experimental Results

We evaluate the FDM algorithm on a semi-synthetic dataset where the ground truth topics are known and taken to be realistic (topics learned by LDA on NeurIPS data) and on three real world datasets: the NeurIPS full papers corpus, a very large (20 million tweets) Twitter dataset that was collected via the Twitter API and the full English Wikipedia. For the semi-synthetic dataset the topic quality was measured by comparison to the ground truth topics, while for the real datasets coherence and log-likelihhod on a hold-out set was measured. We compare FDM to a state of the art LDA collapsed Gibbs sampler (termed SparseLDA), and to the topic learning algorithm introduced in (Arora et al., 2013) (termed EP). Additional details on these benchmarks are given in Section 2. For the semi-synthetic models, we find that while, as expected, SparseLDA with true hyperparameters performs somewhat better, FDM performs similarly to a SparseLDA with close but different hyperparameters. All algorithms find a reasonable approximation of the topics, althpough EP performance is notably weaker. On NeurIPS FDM performs similarly to SparseLDA, while on Twitter and Wikipedia data FDM performance is somewhat better. Both algorithms outperform EP.

To summarize, the contributions of this paper are as follows: We introduce a new approach to topic modeling, via the fitting of the empirical FDM $\widehat{M}$ to the topic FDM $M(\mu, \alpha)$ by likelihood minimization, and prove the consistency of the associated loss. We introduce a practical optimisation procedure for this method, and experimentally show that it produces topics that are comparable or better than the state of the art approaches, while using principles that significantly differ from the existing methods.

2. Literature

The subject of optimization in topic models has received significant attention in the literature. We first describe the general directions of this research. Variational methods for the LDA objective were developed in the paper that introduced the LDA model, (Blei et al., 2002). See also (Foulds et al., 2013), (Hoffman et al., 2013). The collapsed Gibbs sampler for LDA was introduced in (Griffiths & Steyvers, 2004), (Steyvers & Griffiths, 2007). Optimizations of the collapsed sampler that exploit the sparsity of the topics in a document were developed in (Yao et al., 2009), (Xiao & Stibor, 2010), (Li et al., 2014) and yielded further performance improvements. Streaming, or online methods for the LDA objective were proposed in (Newman et al., 2009), (Hoffman et al., 2010). We also note that the topic modelling problem, and in particular the pLSA model, is closely related to the Non-Negative Matrix Factorization (NMF) problem, (Huang et al., 2012). In this context, EM type algorithms for topic models were developed in the paper that introduced pLSA, (Hofmann, 1999). Streaming algorithms for NMF in general are also an active field, see for instance (Zhao & Tan, 2016), (Guan et al., 2012), which involve Euclidean costs, but could possibly be adapted to the topic modelling setting. Finally, distributed methods for LDA were introduced in (Newman et al., 2009), (Smola & Narayanamurthy, 2010), (Liu et al., 2011), (Zhai et al., 2012), (Ahmed et al., 2012).

Topic reconstruction from corpus statistics such as the matrix $\widehat{M}$ were previously considered in the theoretical study of topic models. Topic reconstruction from the third moments of the data was proposed in (Anandkumar et al., 2012). While highly theoretically significant, these algorithms require construction an analysis of an $N \times N \times N$ matrix, and thus can not be applied with dictionaries of size of several thousands tokens. An algorithm that is based on the matrix $\widehat{M}$, as in our approach, was given in (Arora et al., 2012) and improved in (Arora et al., 2013). However, despite the fact that both (Arora et al., 2013) and our approach use $\widehat{M}$, the underling principles behind the two algorithms are completely different. The approach of (Arora et al., 2013) is based on the notion of anchor words (see Section 4), and consists of two steps: First, the algorithm attempts to explicitly identify the anchor words from the matrix $\widehat{M}$. Then, the topics are reconstructed using the obtained anchor words.

Due to the structure of the pLSA model, it can be shown that any row of the matrix $\widehat{M}$ can be approximately represented as a convex combination of the rows of $\widehat{M}$ that correspond to the anchor words. Thus, the anchor word identification in (Arora et al., 2013) is done by identifying the approximate extreme points of a set, where the set is the set of rows of $\widehat{M}$. In contrast, our approach does not attempt to find or use anchor words explicitly and conceptually is a much simpler gradient descent algorithm. We optimize in the space of topics directly, by approximating $\widehat{M}$ by an FDM $M(\mu, \alpha)$. It is also worth mentioning that in the consistency proof of Algorithm 2, Theorem 4.2 (although not in the algorithm itself), we use the characterization of the topics as the extreme points of a certain polytope. However, these are not the same extreme points as in (Arora et al., 2013). The extreme points we use for the proof correspond to topics, while the extreme points in (Arora et al., 2013) correspond to conditional probabilities of tokens given anchor words, and are generally very different from the topics themselves. Finally, as mentioned earlier, we use the algorithm form (Arora et al., 2013) as an additional benchmark. While the algorithm of (Arora et al., 2013) is extremely fast, and can be very precise under certain conditions, the quality of the topics found by that algorithm is significantly lower compared to the topics produced by the standard optimized LDA Gibbs sampler.

A variant of the Gibbs sampler for LDA that is adapted to the computation on GPU was recently proposed in (Tristan et al., 2015). We note that similarly to the Gibbs samplers or variational methods, this approach maintains a form of a topic assignment for each document. Therefore, the number of variables that need to be stored grows with the number of documents *and* topics and is $\Omega(D \cdot T)$, where D is the number of documents and T number of topics. This is true despite the remarkable memory optimizations described in (Tristan et al., 2015), which address other matrices used by that algorithm. Observe that for GPUs, this problem is quite severe, since the amount of memory typically available on a GPU is much smaller than the CPU memory. This is in contrast to our approach, where the GPU memory requirement is independent of the number of documents or tokens. To put this in context, the datasets we analyze here, NeurIPS and Twitter (both with $T = 500$) can not be analyzed via the approach of (Tristan et al., 2015) on a high end desktop GPU ($10GB$ memory).

The MALLET code, (McCallum, 2002), is widely used as the standard benchmark. This code is based on a collapsed Gibbs sampler for LDA, and implements a variety of optimizations discussed earlier. In particular it exploits sparsity, based on (Yao et al., 2009), parallelization (within a single workstation) based on (Newman et al., 2009), and has an efficient and publicly available implementation.

Finally, we note that, while outside of the scope of this paper, the methods presented here could easily be adapted to distributed, multi-GPU settings, using standard SGD parallelization techniques. This may be achieved either by elementary means, by increasing the batch size and processing it on multiple GPUs in parallel, or via more involved, lock-free methods, such as (Recht et al., 2011).

3. Formal Algorithm Specification

Recall that $|\mathcal{X}| = N$ is the size of the dictionary. For a document d given as a sequence of tokens $d = \{x_1, \dots, x_{l_d}\}$, where l_d is the total number of tokens in d, denote by $c_d \in \mathbb{R}^N$ the count vector of d,

$$c_d(u) = \#\{x_i \mid x_i = u\} \text{ for } u \in \mathcal{X}. \tag{5}$$

Thus, c_d is the bag of words representation of d, and $c_d(u)$ is the number of times u appears in d. With this notation, the construction of the matrix $\widehat{M}$ from the data is described in Algorithm 1.

To motivate this construction, assuming the pLSA model, suppose a document d is sampled from a mixture of topics

$$\nu = \sum_{t \le T} \theta(t)\mu_t. \tag{6}$$

The co-occurrence matrix of the mixture ν is

$$(M_\nu)_{u,v} = \nu(u) \cdot \nu(v) = \sum_{i,j \le T} \theta(i)\theta(j)\mu_i(u)\mu_j(v). \tag{7}$$

Thus, $(M_\nu)_{u,v}$ is the probability of obtaining the pair (u, v) when one samples two token independently from ν. The co-occurrence matrix of the corpus is the average of the corresponding M_ν over all documents d. Observe form (7) that M_ν that has the form of an FDM, (2), and hence the co-occurrence matrix of the full corpus also has this form. Next, note that we do not have access to the matrices M_ν themselves, only to the documents d, which are samples from ν. We therefore estimate M_ν using the tokens of the document. Specifically, the matrix $\widehat{M}_d$ constructed in (9) in Algorithm 1 is an unbiased estimate of M_ν from the tokens in d: We have

$$\mathbb{E}_d \widehat{M}_d \,|\theta = M_\nu, \tag{8}$$

where the expectation is over the documents sampled from ν. We provide the proof of this statement in the supplementary material. Therefore, to obtain an approximation to the co-occurrence matrix of the model, in Algorithm 1 we first compute the estimates $\widehat{M}_d$ for each document, and then average over the corpus.

Once the matrix $\widehat{M}$ is constructed, our goal is to reconstruct the topics μ and the corpus level coefficients α from $\widehat{M}$.

Algorithm 1 Computation of $\widehat{M}$

Input: Corpus: $\mathcal{C} = \{d_1, ..., d_D\}$, a corpus with D documents.

1: For every $d \in \mathcal{C}$ construct $\widehat{M}_d$ such that the entry u, v is:

$$\left(\widehat{M}_d\right)_{u,v} = \begin{cases} \frac{l_d}{l_d-1}\frac{c_d(u)}{l_d}\frac{c_d(v)}{l_d} & \text{if } u \neq v \\ \frac{l_d}{l_d-1}\frac{c_d(u)}{l_d}\frac{c_d(v)}{l_d} - \frac{c_d(u)}{l_d(l_d-1)} & \text{if } u = v \end{cases} \quad (9)$$

2: Set

$$\widehat{M} = \frac{1}{D}\sum_{d\in\mathcal{C}} \widehat{M}_d. \quad (10)$$

Algorithm 2 FDM Optimization

Input: $\widehat{M}$: The empirical FDM
Input: B: Batch size
Input: T: Number of topics

1: Initialize free variables α', μ'_t at random.
2: Set $\alpha = softmax(\alpha'), \mu_t = softmax(\mu'_t)$.
3: **while** L_B not converged **do**
4: Sample B pairs of tokens, $Batch = \{(u_1, v_1), ..., (u_B, v_B)\}$. Each pair is sampled from $\widehat{M}$, $(u_i, v_i) \sim \widehat{M}$.
5: Set $L_B(\mu, \alpha) = \sum_{k=1}^{B} \log \sum_{i,j}^{T} \alpha_{i,j}\mu_i(u_k)\mu_j(v_k)$
6: $(\mu', \alpha') \leftarrow (\mu', \alpha') + \gamma\nabla_{\mu',\alpha'} L_B$
7: **end while**

As discussed in the introduction, the FDM optimization algorithm is a stochastic gradient ascent on the pairs of tokens (u, v) sampled from $\widehat{M}$, given as Algorithm 2. Note that since the parameters α, μ in which we are interested are constrained to be probability distributions, we parametrize them with free variables α', μ'_t via softmax:

$$\alpha_{i,j} = \frac{e^{\alpha'_{ij}}}{\sum_{i',j'\leq T} e^{\alpha'_{i'j'}}} \text{ and } \mu_t(l) = \frac{e^{\mu'_t(l)}}{\sum_{l'\leq N} e^{\mu'_t(l')}}, \quad (11)$$

where $i, j \leq T$, $t \leq T$, and $l \leq N$. Thus, in Algorithm 2, α, μ are functions of α', μ' and the gradient ascent is over α', μ'. Note that any SGD optimizer, including adaptive rate optimizers, may be used in Step 6 of Algorithm 2. We use Adam, (Kingma & Ba, 2015), in the experiments.

4. Consistency

In this section we discuss the consistency of the loss function (3) under the anchor words assumption. Specifically, we show that if the the corpus is generated by a pLSA model with a set of topics $\{\mu_t\}_1^T$ that satisfy the anchor words assumption, then among topics satisfying this assumption, the loss (3) can only be asymptotically minimized by an FDM $M(\mu, \alpha)$ with the true topics μ. It follows that if one minimizes the loss (3) and the resulting topics satisfy the anchor words assumptions (this holds empirically, see below), then in the limit the topics must be the true topics.

The anchor words assumption was introduced in (Donoho & Stodden, 2003) as a part of a set of sufficient conditions for identifiability in NMF. A set $\{\mu_t\}_1^T$ of topics is said to have an *anchor words* property, denoted $(\mathcal{AW})$, if for every $t \leq T$, there is a point $u_t \leq N$ such that $\mu_t(u_t) > 0$ and $\mu_{t'}(u_t) = 0$ for all $t' \neq t$. The point u_t is called the anchor word of the topic μ_t. As mentioned earlier, natural topics tends to have the anchor word property. For instance, topics found by various LDA based methods have anchor words, (Arora et al., 2013). We note that topics found by the FDM optimization algorithm, Algorithm 2, also have anchor words.

We first develop a few equivalent geometric characterizations of anchor words. While some of the arguments used in the proofs are similar in spirit to those in (Donoho & Stodden, 2003),(Arora et al., 2012), the particular notions we introduce have not appeared in the literature previously, and significantly clarify the geometric nature of the anchor word assumption. We thus provide these results here for completeness. Some of the equivalences below play an important role in the proof of Theorem 4.2.

Denote by Δ_{N-1} the probability simplex in $\mathbb{R}^N$. A set of probability measures $\{\mu_t\}_1^T \subset \Delta_{N-1}$ is called *span-maximal* $(\mathcal{SM})$ if $\Delta_{N-1} \cap span\{\mu_t\}_1^T = conv\{\mu_t\}_1^T$. Here *span* and *conv* denote the span and the convex hull of the set. A set $\{\mu_t\}_1^T \subset \Delta_{N-1}$ is *positive* $(\mathcal{P})$ if the following holds: For every linear combination $\sum_t a_t$, if $\sum_t a_t\mu_t \in \Delta_{N-1}$ then $a_t \geq 0$ for all $t \leq T$. In other words, a set is positive if only its linear combinations with non-negative coefficients belong to the simplex. Finally, a set is *maximal* $(\mathcal{M})$ if the following holds: For any set $\{\nu_t\}_1^T \subset \Delta_{N-1}$, if $conv\{\mu_t\}_1^T \subset conv\{\nu_t\}_1^T$, then we must have $\mu_t = \nu_{\pi(t)}$ for some permutation π on $\{1, \ldots, T\}$. Equivalently, a set is maximal if it can not be properly contained in a convex hull of another set of topics of size T.

Proposition 4.1. *For a set of linearly independent topics $\{\mu_t\}_1^T \subset \Delta_{N-1}$, the properties $(\mathcal{SM})$, $(\mathcal{P})$, $(\mathcal{AW})$, and $(\mathcal{M})$ are equivalent.*

The proof is given in supplementary material Section C. Note in addition that $(\mathcal{AW})$ implies that $\{\mu_t\}_1^T \subset \Delta_{N-1}$ are linearly independent. We now discuss the relation between between $\{\mu_t\}_1^T$ and an FDM matrix $M_{u,v} = \sum_{i,j=1}^{T} \alpha_{ij}\mu_i(u)\mu_j(v)$. In particular, we describe how $\{\mu_t\}_1^T$ can be recovered from the image of M (as an operator, $M : \mathbb{R}^N \to \mathbb{R}^N$) under $(\mathcal{AW})$. Note first that the image of M satisfies $Im(M) \subseteq span\{\mu_t\}_1^T$. Indeed,

for a fixed v, a column of M can be written as a linear combination, $M_{\cdot,v} = \sum_i \left(\sum_j \alpha_{i,j}\mu_j(v)\right)\mu_i(\cdot)$. Moreover, if the matrix $\alpha = \alpha_{ij}$ has full rank, $rank(\alpha) = T$, and if $\{\mu_t\}_1^T$ are linearly independent, then $Im(M) = span\{\mu_t\}_1^T$. Therefore, when $rank(\alpha) = T$, given M we can recover $span\{\mu_t\}_1^T$ as $Im(M)$. Now, if $\{\mu_t\}_1^T$ satisfies $(\mathcal{AW})$, then by Proposition 4.1 it also satisfies $(\mathcal{SM})$. It then follows that $conv\{\mu_t\}_1^T$ can be recovered as $conv\{\mu_t\}_1^T = Im(M) \cap \Delta_{N-1}$. Finally, if we know $conv\{\mu_t\}_1^T$, we can recover $\{\mu_t\}_1^T$, since these are simply the extreme points of that polytope, and every polytope is uniquely characterized by its extreme points. This relation between $\{\mu_t\}_1^T$ and M is at the basis of the consistency result below.

We state the consistency for the probabilistic setting: We complement the pLSA model to be a full generative model by assuming that topic distribution θ_d is sampled independently for each document d from some probability distribution $\mathcal{T}$ on Δ_{T-1}. If $\mathcal{T}$ is a Dirichlet distribution, symmetric or asymmetric, this corresponds to an LDA model. Other examples include models with correlated topics, (Blei & Lafferty, 2007), or hierarchical topics, (Li & McCallum, 2006), among many others. The only requirement on the topic distribution is the following: Let $\Theta_{i,j} = \mathbb{E}\theta_d(i)\theta_d(j)$ be the expected topic-topic co-occurrence matrix corresponding to the sampling scheme. We require that Θ is full rank. This assumption holds in all the examples above.

Theorem 4.2 (Consistency)**.** *Consider a generative pLSA model (1), over topics $\{\mu_i\}_1^T$ which satisfy $(\mathcal{AW})$, and where θ_d are sampled independently from a fixed distribution on Δ_{T-1}, with topic-topic expected co-ocurrence matrix Θ. Set $M_{u,v} := M(\mu,\Theta) = \sum_{i,j=1}^T \Theta_{i,j}\mu_i(u)\mu_j(v)$. Then with probability* 1*,*

$$\lim_{D\to\infty} \widehat{M}_{u,v} \log M_{u,v} = \sum_{u,v} M_{u,v} \log M_{u,v}. \qquad (12)$$

Conversely, let $\{\mu'_t\}_{t=1}^T \neq \{\mu_t\}_{t=1}^T$ be a different set of topics satisfying $(\mathcal{AW})$. There is a $\gamma = \gamma\left(\{\mu_i\}_1^T, \{\mu'_i\}_1^T, \kappa(\Theta)\right) > 0$, such that for any *FDM M' over $\{\mu'_i\}_1^T$, with probability* 1*,*

$$\lim_{D\to\infty} \widehat{M}_{u,v} \log M'_{u,v} \leq \sum_{u,v} M_{u,v} \log M_{u,v} - \gamma. \qquad (13)$$

This result is proved in supplementary material Section D. The probability in Theorem 4.2 is over the samples from the generative model, through the random variables $\widehat{M}$ (note that $\widehat{M}$, defined (10), depends on D). The theorem states that the loss (3) is minimized at the true parameters $\{\mu_i\}_1^T$ and Θ. Indeed, denote $L_0 = \sum_{u,v} M_{u,v} \log M_{u,v}$. Note that the cost $L(\mu,\Theta)$ is a random variable, as it depends on $\widehat{M}$. Then Eq. (12) states that $L(\mu,\Theta)$ converges to L_0, while Eq. (13) is equivalent to stating that for any other set of topics, $\{\mu'_i\}_1^T$, we have $L(\mu',\alpha) > L_0 + \gamma$ for any α. The gap γ will depend on how well $\{\mu'_i\}_1^T$ approximates the true topics $\{\mu_i\}_1^T$.

5. Experiments

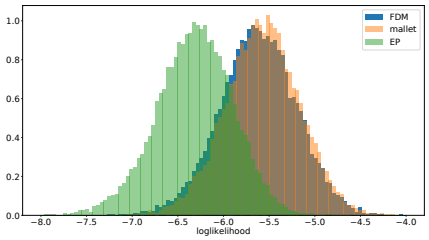
In the following sections we discuss experiments on semi-synthetic data, on the NeurIPS papers corpus, and on Twitter and Wikipedia datasets; see Section 1.3 for an overview. For non synthetic data the performance is evaluated by log-likelihood on test set, and by the coherence measure of the topics. The coherence results are discussed in Section 5.5.

5.1. Synthetic Data

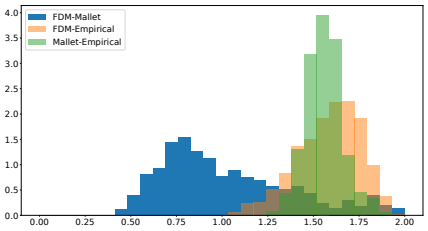
To approximate real data in a synthetic setting, we used $T = 500$ topics learned by SparseLDA optimization on the NeurIPS papers corpus (Section 5.2) as the ground truth topics. The dictionary size in this case is $N = 10500$ tokens. The synthetic documents were generated using the LDA model: For each document, a topic distribution θ_d was sampled per document from a symmetric Dirichlet with the standard concentration parameter $\alpha = 1/T$, and 30 tokens were sampled from the θ_d mixture of the ground truth topics. The corpus size was $D = 100000$ documents. Note that for a dictionary of size $N = 10500$ and non-uniform topics, this is not a very large corpus.

To reconstruct the topics, we compared three algorithms: (i) SparseLDA – a sparsity optimized parallel collapsed Gibbs sampler for LDA, implemented in the MALLET framework (see Section 2 for details), which was run with 4 parallel threads. (ii) FDM, Algorithm 2. (iii) The topic learning algorithm from (Arora et al., 2013), to which we refer as EP (Extreme Points) in what follows. SparseLDA was run with 4 threads. All algorithms were run 5 times, until convergence, on the fixed dataset. Note that the EP algorithm does not have a random initialization, but uses a random projection as an intermediate dimensionality reduction step. Therefore different runs might be affected by restarts (although very mildly, in practice). We used $M = 1000$ as the dimension of the random projection – a value that was specified in (Arora et al., 2013) as the practical value for the dictionary sizes of the order we use here. Hardware specifications are given in the supplementary material. All the algorithms were run with the true number of topics T as a parameter.

The SparseLDA algorithm was run in two modes: With the true hyperparameters, $\alpha = 1/T$, corresponding to the true α of the corpus, and with topic sparsity parameter $\beta = 1/N$, a standard setting which was also used to learn the ground



(a) Distribution of log-likelihoods on test set. Larger (closer to zero) values are better. FDM (blue), SparseLDA (orange), EP (green).



(b) Distribution of distances in an optimal Matching. FDM to SparseLDA (blue), FDM to empirical (orange), SparseLDA to empirical(green).

Figure 2. NeurIPS Papers Experiment

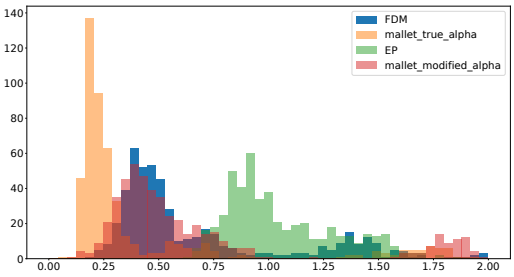


Figure 3. Distribution of the distances to ground truths, in an optimal matching, for a typical instance of each method.

truth topics. To model a situation where the true hyperparameters are unknown, we also evaluated SparseLDA with a modified hyperparameter $\alpha = 10/T$, and same β. Note that this is a relatively mild change of the hyperparameter.

The quality of the learned topics was measured by calculating the optimal matching ℓ_1 [1] distances to the ground truth topics. That is, given the topics returned by the model, ν_t, $1 \leq t \leq T$ and the ground truth topics μ_t, we compute

$$err = \frac{1}{T} \min_{\tau} \sum_{t=1}^{T} \left| \mu_t - \nu_{\tau(t)} \right|_1 , \tag{14}$$

where τ is the matching — a permutation of the set $\{1, \ldots, T\}$. The optimal matching τ was computed using the Hungarian algorithm.

The results are given in Table 1, which shows for each algorithm the average error and the standard deviation over the different runs. To put the numbers in perspective, the typical ℓ_1 distance between two ground truth topics is around 1.75. Thus all algorithms learned at least some approximation of the ground truth. By visual inspection, a topic at distance 0.6 from a given ground truth topic tends to capture the mass at the correct tokens, but the amount of mass deviates somewhat from the correct one.

We observe that SparseLDA with the ground truth α attained best performance. This is not surprising, since the algorithm is based on the generative model that is the true generative model of the corpus, and was provided with the true hyperparameters. Both of these constitute a considerable prior knowledge. The performance of the EP algorithm was relatively low, which is likely due to the fact that the corpus size was not sufficiently large for that algorithm, with this set of topics.

Finally, FDM and SparseLDA with the modified hyperparameter attained similar performance. Interestingly, SparseLDA and FDM tend to err slightly differently. In Figure 3 for a set of topics found by each algorithm we show the histogram of the quantities $\left| \mu_t - \nu_{\tau(t)} \right|_1$, i.e. the distribution of the distances within the matching. SparseLDA tends to completely miss about 50 to 80 out of 500 topics, while FDM is slightly less precise on the topics that it does approximate well. This figure is remarkably consistent across the different runs of the algorithms.

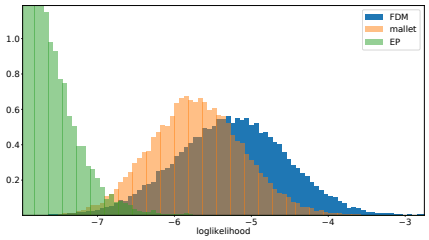
Table 1. Synthetic Corpus Matching Distances

Model	Average ℓ_1
FDM	0.66 ± 0.005
SparseLDA, $\alpha = \alpha^*$	0.41 ± 0.0004
EP	1.05 ± 0.005
SparseLDA, $\alpha = 10\alpha^*$	0.66 ± 0.01

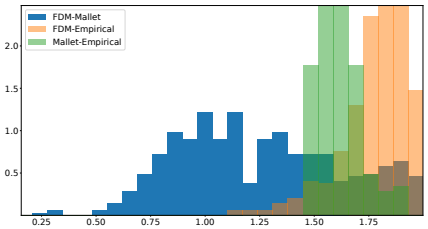
5.2. NeurIPS Dataset

The NeurIPS dataset, (NeurIPSPapersCorpus, 2016), consists of all NeurIPS papers from 1987 to 2016. Each document was taken to be a single paragraph. Stop words, numbers and tokens appearing less than 50 times were removed. All tokens were stemmed and documents with less than 20 tokens are removed. The preprocessed dataset con-

[1] $|\mu - \nu|_1 = \sum_{u \in \mathcal{X}} |\mu(u) - \nu(u)|$



(a) Distribution of log-likelihoods on test set. Larger (closer to zero) values are better. FDM (blue), SparseLDA (orange), EP (green).



(b) Distribution of distances in an optimal Matching. FDM to SparseLDA (blue), FDM to empirical (orange), SparseLDA to empirical(green).

Figure 4. Twitter Experiment

tained roughly $D = 251000$ documents over the dictionary of around $N = 10500$ unique tokens. 20% of the documents were taken at random as a hold-out (test) set. The log-likelihood of the documents in the hold-out set was used as the performance measure. The computation of the (log) likelihood on the hold out set given topics is standard, with full details provided in supplementary material Section E. The following models were trained: FDM, SparseLDA ($\alpha = 1/T, \beta = 1/N$) and EP ((Arora et al., 2013)) with $T = 500$ topics. All models were run until convergence.

The mean hold-out log-likelihoods for each method are shown in Table 2, and a histogram of the distribution of the holdout log-likelihoods for a single run of each algorithm is shown in Figure 2a. We observe that the performance of SparseLDA and FDM are practically identical, and both perform better than EP.

To obtain some insight into the relation between the models, Figure 2b shows the histogram of the optimal matching distances between the topics learned by a fixed run of SpraseLDA and a fixed run of FDM (blue). For scale, the distances of SparseLDA and FDM to a fixed topic, the empirical distribution of the corpus, are also shown. It appears that both models find somewhat similar topics.

Table 2. Test set average Log-likelihoods

Model	NeurIPS	Twitter	Wiki
FDM	$\mathbf{-5.62}$	$\mathbf{-5.26}$	$\mathbf{-6.28}$
SparseLDA	$\mathbf{-5.57}$	-5.70	$-6.68(-6.45)$
EP	-6.31	-8.5	-6.84

5.3. Twitter Data

We first describe the collection and processing of the Twitter corpus. The tweets were collected via the Tweeter API, (TwitterAPI). The data contains about 16M (million) (after pre-processing, see below) publicly available tweets posted by 60K users during roughly the period of 1/2014 to 1/2017. The tweets were preprocessed to remove numbers, none-ASCII characters, mentions (Twitter usernames preceded by an @ character) and URLs. All tokens were stemmed, and stop words and tokens with less than 3 characters were removed. The most common 200 tokens and rare tokens (less than 1000 appearances in the corpus) were removed. Following this, tweets shorter than 4 tokens were also removed. This resulted in a corpus of about $D = 16M$ tweets over a dictionary of slightly more than $N = 15000$ unique tokens. Each tweet was considered as a separate document, and typical tweets have 4 to 8 tokens.

The experiment setting was similar to the NeurIPS papers corpus. 20% of the documents were taken as a hold-out set, and the holdout loglikelihood of SparseLDA, FDM and EP topics was evaluted. All algorithms were run to convergence, for about 12 hours for each run.

The resulting hold-out log-likelihoods are given in Table 2, and the distribution of the log-likelihoods in shown in Figure 4a. We observe that in this case the performance of FDM is better than that of SparseLDA, while EP does not produce a good approximation of the dataset.

5.4. Wikipedia

We use the full English Wikipedia corpus, as archived on 04/2020. In Wikipedia the articles are naturally divided into sections, and each section was taken as a single document. 20% of articles (that is, all sections from these articles) were taken as test set. Similarly to the Twitter data, the text was stemmed and stop words were removed. Then the dictionary was restricted to $N = 50000$ most common tokens. After this preprocessing the total number of tokens in the train set was 1.3 billion, across approximately 11 million, $D = 11 \cdot 10^6$, documents.

In all models we have used $T = 1000$ topics, and similarly to the other datasets the LDA models were trained with the standard hyperparameters $\alpha = 1/T, \beta = 1/N$.

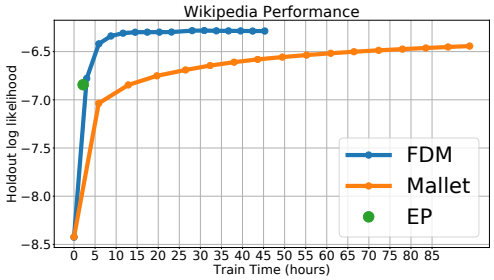


Figure 5. Wikipedia Test Likelihoods as Function of Train Time

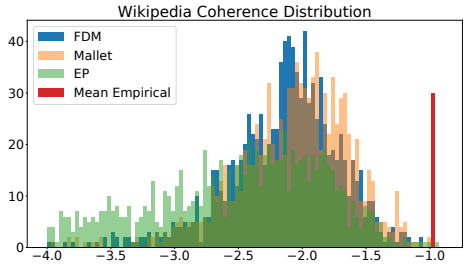


Figure 6. Wikipedia Distribution of Topic Coherence

The test set log-likelihoods are given in Table 2, and shown as a function of train time in Figure 5. While absolute training times depend strongly on hardware, in this case both FDM and Mallet were run on the same system, using CPUs and GPU of the same generation. See Section F for full hardware details.

As shown in Figure 5, EP provides non-trivial results relatively quickly [2]. However, these results are significantly weaker than what FDM and Mallet provide. FDM converges to its best log-likelihood, -6.28, after about 10 hours, while MALLET, after 24 hours achieves -6.68, and -6.45 after 80 hours, which still does not match the FDM performance.

5.5. Coherence

In addition to the holdout log-likelihood, another topic quality measure that is often used in the literature is topic coherence, (Mimno et al., 2011).

The coherence of a topic μ with respect to the corpus is defined as follows: Let $C(v)$ denote the number of documents containing at least one instance of the token v, and let $C(v, w)$ denote the number of documents containing at least one instance of both v and w. For a topic μ, denote by $u_1, \ldots, u_M$ the M tokens with the highest weights in the topic, ordered by weight. Thus for instance u_1 is the token with the highest weight $\mu(u_1)$. The (normalized) coherence of μ is then given by

$$Coh(\mu; M) = \frac{1}{\binom{M}{2}} \sum_{m=2}^{M} \sum_{l=1}^{m-1} \log \frac{C(u_m, u_l) + 1}{C(u_l)}.$$

Coherence values are normally negative, and typically values closer to 0 are considered to be better. All coherences below are computed at the standard value of $M = 10$ top words. For a set of topics produced by any given model, we compute the average coherence of the topics. The results for all models and datasets we consider are given in Table 3. In particular FDM and SparseLDA achieve similar coherence score, with SparseLDA's slightly higher.

When discussing coherence, it is crucial to note that coherence typically *decreases* with training. Moreover, for a given corpus, define the empirical mean topic $\bar{\mu}$ as $\bar{\mu}(v) = C(v)/\sum_u C(u)$. That is, this is the topic in which each token has the same frequency as in the whole corpus. While this is in many senses a trivial topic, it usually has high coherence compared to other, more specific and useful topics. The coherence of this topic for each corpus is shown in the last row of Table 3. Since most initialization algorithms initialize topics such that they are close to μ, this explains the decrease of coherence during training. Due to this reason, while higher values of coherence might be desirable, such values are only meaningful when log-likelihood values are also high.

The distribution of coherences for the Wikipedia topics for FDM, Mallet and EP are shown in Figure 6.

Table 3. Average Topic Coherence (at 10 topwords)

Model	NeurIPS	Twitter	Wiki
FDM	−2.62	−4.51	−2.15
SparseLDA	−2.44	−4.36	−2.07
EP	−2.19	−7.35	−2.72
Mean Empirical	**−1.34**	**−4.02**	**−0.98**

6. Conclusions

In this paper a introduced a new topic modeling approach, FDM topic modeling, which is based on matching, via KL divergence, of the token co-occurrence distribution induced by the topics to the co-occurrence distribution of the corpus. We have shown the asymptotic consistency of the approach under the anchor words assumption and presented an efficient stochastic optimization procedure for this problem. This algorithm enables the approach to leverage GPU computation and efficient SGD optimizers. Our empirical evaluation shows that FDM produces topics of good quality and improves over the performance of SparseLDA.

[2] Note that EP is not an iterative method. Its results can not be further improved by using more time.

References

Ahmed, A., Aly, M., Gonzalez, J., Narayanamurthy, S., and Smola, A. J. Scalable inference in latent variable models. In *Proceedings of the fifth ACM international conference on Web search and data mining*, pp. 123–132. ACM, 2012.

Anandkumar, A., Foster, D. P., Hsu, D. J., Kakade, S. M., and kai Liu, Y. A spectral algorithm for latent dirichlet allocation. In *NIPS*. 2012.

Arora, S., Ge, R., and Moitra, A. Learning topic models – going beyond svd. FOCS '12, 2012.

Arora, S., Ge, R., Halpern, Y., Mimno, D., Moitra, A., Sontag, D., Wu, Y., and Zhu, M. A practical algorithm for topic modeling with provable guarantees. In *ICML*, 2013.

Bhatia, R. *Matrix Analysis*. Graduate Texts in Mathematics. Springer New York, 1997.

Blei, D. M. and Lafferty, J. D. A correlated topic model of science. *The Annals of Applied Statistics*, 2007.

Blei, D. M., Ng, A. Y., and Jordan, M. I. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3, 2002.

Cover, T. M. and Thomas, J. A. *Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing)*. Wiley-Interscience, 2006.

Donoho, D. and Stodden, V. When does non-negative matrix factorization give a correct decomposition into parts? In *Proceedings of the 16th International Conference on Neural Information Processing Systems*, NIPS'03, 2003.

Foulds, J., Boyles, L., DuBois, C., Smyth, P., and Welling, M. Stochastic collapsed variational bayesian inference for latent dirichlet allocation. In *KDD*, 2013.

Griffiths, T. L. and Steyvers, M. Finding scientific topics. *Proceedings of the National Academy of Sciences of the United States of America*, 101, 2004.

Guan, N., Tao, D., Luo, Z., and Yuan, B. Online nonnegative matrix factorization with robust stochastic approximation. *IEEE Transactions on Neural Networks and Learning Systems*, 2012.

Hoffman, M., Bach, F. R., and Blei, D. M. Online learning for latent dirichlet allocation. In *Advances in Neural Information Processing Systems 23*. 2010.

Hoffman, M. D., Blei, D. M., Wang, C., and Paisley, J. Stochastic variational inference. *J. Mach. Learn. Res.*, 14 (1), 2013.

Hofmann, T. Probabilistic latent semantic indexing. SIGIR '99, 1999.

Huang, Z., Zhou, A., and Zhang, G. Non-negative matrix factorization: A short survey on methods and applications. In *Computational Intelligence and Intelligent Systems*. Springer Berlin Heidelberg, 2012.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

Li, A. Q., Ahmed, A., Ravi, S., and Smola, A. J. Reducing the sampling complexity of topic models. In *KDD*, 2014.

Li, W. and McCallum, A. Pachinko allocation: Dag-structured mixture models of topic correlations. In *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06. Association for Computing Machinery, 2006.

Liu, Z., Zhang, Y., Chang, E. Y., and Sun, M. Plda+: Parallel latent dirichlet allocation with data placement and pipeline processing. *ACM Trans. Intell. Syst. Technol.*, 2011.

McCallum, A. K. Mallet: A machine learning for language toolkit. http://mallet.cs.umass.edu, 2002.

Mimno, D., Wallach, H. M., Talley, E., Leenders, M., and McCallum, A. Optimizing semantic coherence in topic models. In *EMNLP*, 2011.

NeurIPSPapersCorpus. Neuripspaperscorpus. https://www.kaggle.com/benhamner/nips-papers, 2016. Accessed: 1/8/2019.

Newman, D., Asuncion, A., Smyth, P., and Welling, M. Distributed algorithms for topic models. *J. Mach. Learn. Res.*, 2009.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Recht, B., Re, C., Wright, S., and Niu, F. Hogwild: A lock-free approach to parallelizing stochastic gradient descent. In *Advances in Neural Information Processing Systems 24*. 2011.

Smola, A. and Narayanamurthy, S. An architecture for parallel topic models. *Proc. VLDB Endow.*, 2010.

Steyvers, M. and Griffiths, T. *Probabilistic Topic Models*. 2007.

Tristan, J.-B., Tassarotti, J., and Steele, G. Efficient training of lda on a gpu by mean-for-mode estimation. In *ICML*, 2015.

TwitterAPI. Twitter api. https://developer.twitter.com/en/docs/tweets/timelines/api-reference/get-statuses-user_timeline.html. Accessed: 1/8/2019.

Xiao, H. and Stibor, T. Efficient collapsed gibbs sampling for latent dirichlet allocation. In *ACML*, 2010.

Yao, L., Mimno, D., and McCallum, A. Efficient methods for topic model inference on streaming document collections. In *KDD*, 2009.

Zhai, K., Boyd-Graber, J., Asadi, N., and Khouja, J. Mr. lda: a flexible large scale topic modeling package using variational inference in mapreduce. In *WWW*, 2012.

Zhao, R. and Tan, V. Y. F. Online nonnegative matrix factorization with outliers. In *ICASSP*, 2016.

Topic Modeling via Full Dependence Mixtures - Supplementary Material

A. Outline

In this supplementary material we provide the proofs of the results stated in the main text. In addition, details on the hardware used in the experiments are given in Section F.

B. The Unbiased Estimate $\widehat{M}_d$

In this section we discuss the construction of the matrix $\widehat{M}_d$, defined in (9) in Algorithm 1. Specifically, we show the unbiased estimate property of $\widehat{M}_d$, (8).

First, let us introduce some notation. For any two vectors $a, b \in \mathbb{R}^N$, denote by $a \otimes b \in \mathbb{R}^{N \times N}$ the outer product of a, b, an $N \times N$ matrix given by:

$$(a \otimes b)_{u,v} = a(v) \cdot b(u). \tag{15}$$

For any probability distribution μ on $\mathcal{X}$, $(\mu \otimes \mu)_{u,v}$ is simply the probability of obtaining the pair u, v when sampling independently twice from μ.

For a document d, recall that c_d, defined in (37), is the token counts vector of the document d and l_d is the total number of tokens in d. Set $\hat{d} = \frac{1}{l_d} c_d$ to be the empirical probability distribution on $\mathcal{X}$ corresponding to d.

As described in the main text, assuming the pLSA model, each document d is an i.i.d sample from some mixture of topics

$$\nu_d = \sum_t \theta_d(t) \mu_t. \tag{16}$$

Let us fix some mixture

$$\nu = \sum_t \theta(t) \mu_t. \tag{17}$$

The co-occurrence matrix for the mixture is by definition

$$(M_\nu)_{u,v} = \nu(u) \cdot \nu(v) = \sum_{i,j \leq T} \theta(i)\theta(j)\mu_i(u)\mu_j(v). \tag{18}$$

Note that with our notation, we have

$$M_\nu = \nu \otimes \nu. \tag{19}$$

Moreover, given a document d, note that the empirical co-ocurrence matrix $\widehat{M}_d$ may be written as

$$\widehat{M}_d = \frac{l_d}{l_d - 1} \hat{d} \otimes \hat{d} - \frac{1}{l_d - 1} Diag(\hat{d}). \tag{20}$$

Here $Diag(x)$ is a diagonal $N \times N$ matrix with diagonal entries given by the vector $x \in \mathbb{R}^N$.

We first compute the expectation of $\hat{d} \otimes \hat{d}$ in the following Lemma.

Lemma B.1. *Let $d = \{x_1, \ldots, x_{l_d}\}$ be an i.i.d sample from a distribution ν. Then*

$$\mathbb{E}\hat{d} \otimes \hat{d} = \frac{l_d - 1}{l_d} \nu \otimes \nu + \frac{1}{l_d} Diag(\nu). \tag{21}$$

Proof. Consider the coordinate u, v of the matrix $\mathbb{E}\hat{d} \otimes \hat{d}$.

$$\left(\mathbb{E}\hat{d} \otimes \hat{d}\right)_{u,v} = \mathbb{E}\hat{d}(u) \cdot \hat{d}(v) = \tag{22}$$

$$\frac{1}{l_d^2}\mathbb{E}\left(\sum_{i=1}^{l_d} \mathbf{1}_{\{x_i=u\}}\right) \cdot \left(\sum_{j=1}^{l_d} \mathbf{1}_{\{x_j=v\}}\right) = \tag{23}$$

$$\frac{1}{{l_d}^2}\sum_{i,j=1}^{l_d} \mathbb{E}\left(\mathbf{1}_{\{x_i=u\}} \cdot \mathbf{1}_{\{x_j=v\}}\right).$$

The results is now obtained by by considering separately the cases $i = j$, $i \neq j$, $u = v$, $u \neq v$. Indeed, choose for instance fixed $u \neq v$. For $i \neq j$ we have

$$\mathbb{E}\left(\mathbf{1}_{\{x_i=u\}} \cdot \mathbf{1}_{\{x_j=v\}}\right) = \nu(u) \cdot \nu(v). \tag{24}$$

For $i = j$, since $u \neq v$,

$$\mathbb{E}\left(\mathbf{1}_{\{x_i=u\}} \cdot \mathbf{1}_{\{x_i=v\}}\right) = 0. \tag{25}$$

Since there are $l_d^2 - l_d$ pairs i, j with $i \neq j$, we thus have overall that

$$\left(\mathbb{E}\hat{d} \otimes \hat{d}\right)_{u,v} = \frac{1}{{l_d}^2}\sum_{i,j=1}^{l_d} \mathbb{E}\left(\mathbf{1}_{\{x_i=u\}} \cdot \mathbf{1}_{\{x_j=v\}}\right) = \frac{l_d(l_d-1)}{l_d^2}\nu(u) \cdot \nu(v) = \frac{l_d-1}{l_d}\nu(u) \cdot \nu(v) = \left(\frac{l_d-1}{l_d}\nu \otimes \nu\right)_{u,v}. \tag{26}$$

The diagonal case, $u = v$, is handled similarly. □

It follows therefore that if d is a document constructed by sampling l_d tokens from ν, then $\hat{d} \otimes \hat{d}$ is not an unbiased estimate of $\nu \otimes \nu$. One can however easily fix this by subtracting the diagonal and renormalizing. Indeed, from Lemma B.1 we have

$$\begin{aligned}\nu \otimes \nu &= \frac{l_d}{l_d-1}\mathbb{E}\hat{d} \otimes \hat{d} - \frac{1}{l_d-1}Diag(\nu) \\ &= \frac{l_d}{l_d-1}\mathbb{E}\hat{d} \otimes \hat{d} - \frac{1}{l_d-1}\mathbb{E}Diag(\hat{d}) \\ &= \mathbb{E}\left(\frac{l_d}{l_d-1}\hat{d} \otimes \hat{d} - \frac{1}{l_d-1}Diag(\hat{d})\right) \\ &= \mathbb{E}\widehat{M}_d.\end{aligned} \tag{27}$$

That is, $\widehat{M}_d$ is an unbiased estimate of $\nu_d \otimes \nu_d$.

C. Proof of Proposition 4.1

Proof. $(sM) \implies (P)$: Suppose $\mu = \sum_t a_t\mu_t \in \Delta_{N-1}$. Since also have $\mu \in span\{\mu_t\}_1^T$, by span-maximality it follows that there are $b_t \geq 0$, with $\sum_t b_t = 1$ such that $\mu = \sum b_t\phi_t$. Now, by linear independence, there is a unique representation of μ. Thus $a_t = b_t$. Conversely, to show $(P) \implies (sM)$, suppose $\mu = \sum_t a_t\mu_t \in \Delta_{N-1} \cap span\{\mu_t\}_1^T$. By (P) we have $a_t \geq 0$. Note also that $1 = \sum_u \mu(u) = \sum_t \sum_u a_t\mu_t(u) = \sum_t a_t$. Thus $\mu \in conv\{\mu_t\}_1^T$.

We now show $(P) \iff (\mathcal{AW})$. First assume $(\mathcal{AW})$. Let u_t be the anchor words. Choose some $\mu = \sum_t a_t\mu_t \in \Delta_{N-1}$. Then by definition of the anchor words, $\mu_t(u_t) = \sum_t a_t\mu_t(u_t) = a_t\mu_t(u_t)$. Since $\mu_t(u_t) > 0$, and $\mu \in \Delta_{N-1}$ this implies $a_t \geq 0$. For the converse, we show $\neg(\mathcal{AW}) \implies \neg(P)$. Assume $\neg(\mathcal{AW})$. This means that there is a topic μ_t such that for every token u for which $\mu_t(u) > 0$, there is another topic, μ_{t_u}, such that $\mu_{t_u}(u) > 0$. In particular, set $\tilde{\mu} = \frac{1}{T-1}\sum_{t' \neq t} \mu_{t'}$. It follows then that μ_t is absolutely continuous with respect to μ. That is, for every u for which $\mu_t(u) > 0$, we have $\tilde{\mu}(u) > 0$. Using this property, it is clear that for a small enough $\varepsilon > 0$, we have $\tilde{\mu}(u) - \varepsilon\mu_t(u) \geq 0$ for all u. Thus $\frac{1}{1-\varepsilon}\left(\tilde{\mu}(u) - \varepsilon\mu_t(u)\right) \in \Delta_{N-1}$. However, this expression has a strictly negative coefficient at μ_t, thus contradicting (P).

Finally, we show $(sM) \implies (M)$ and $(M) \implies (AW)$.

$(sM) \implies (M)$:
Assume (sM). We show (M'). Since $conv\{\mu_t\}_1^T \subset conv\{\nu_t\}_1^T$ and since μ_t are linearly independent, we have $span\{\mu_t\}_1^T = span\{\nu_t\}_1^T$. By (sM), this implies $conv\{\mu_t\}_1^T = conv\{\nu_t\}_1^T$. Since any polytope defines its extreme points uniquely, we obtain the conclusion of (M').

For $(M) \implies (AW)$, we equivalently show $\neg(\mathcal{AW}) \implies \neg(M')$. Assume $\neg(\mathcal{AW})$. Consider the distribution $\widetilde{\mu}_t = \frac{1}{1-\varepsilon}(\tilde{\mu}(u) - \varepsilon\mu_t(u))$ constructed earlier. Clearly we may write $\mu_t = \alpha\widetilde{\mu}_t + \beta\tilde{\mu}$ with $\alpha+\beta=1$, and $\alpha, \beta > 0$, strictly positive. It follows that $conv\{\mu_t\}_1^T \subsetneq conv\left(\{\widetilde{\mu}_t\} \cup \{\mu_{t'}\}_{t'\neq t}\right)$, thus implying (M'). □

D. Proof of Theorem 4.2

In this section we use the notation introduced in Section B.

Proof. It follows from Lemma B.1 and (27) that conditioned on θ_d, we have $\mathbb{E}\widehat{M}_d|\ \theta_d = \sum_{i,j=1}^T \theta_d(i)\theta_d(j)\mu_i \otimes \mu_j$. Therefore by definition,

$$\mathbb{E}\widehat{M}_d = \mathbb{E}\sum_{i,j=1}^T \theta_d(i)\theta_d(j)\mu_i \otimes \mu_j = \sum_{i,j=1}^T \Theta_{i,j}\mu_i(u)\mu_j(v) = M. \tag{28}$$

Since M_d are bounded independent variables in $\mathbb{R}^{N\times N}$, and $\widehat{M} = \frac{1}{D}\left(\sum_{i=1}^D \widehat{M}_{d_i}\right)$, by the Law of Large Numbers we have $\widehat{M} \to M$ with probability 1, which in turn implies (12).

Conversely, let

$$M' = \sum_{i,j} \beta_{i,j}\mu'_i \otimes \mu'_j \tag{29}$$

be an FDM based on the topics $\{\mu'_i\}_1^T$. Set $V = span\{\mu_i\}_1^T$ and $V' = span\{\mu'_i\}_1^T$. Clearly $Im(M') \subseteq V'$. On the other hand, since Θ has full rank, $\kappa(\Theta) > 0$, we have $Im(M) = V$ and $\kappa(M) > 0$. Next, by Lemma 4.1, topics with $(\mathcal{AW})$ property are identified by their span. Since $\{\mu'_i\}_1^T \neq \{\mu_i\}_1^T$, this implies $V \neq V'$. Set

$$\gamma' := \inf_{M' \ s.t.\ Im(M')\subseteq V'} \|M - M'\|_{op} > 0, \tag{30}$$

where the norm is (say) the operator norm. The fact that $\gamma > 0$ is crucial and follows, for instance, from the Davis-Kahan "sin" Perturbation Theorem, (Bhatia, 1997, Section VII.3). Indeed, assume to the contrary that $\gamma = 0$. This would imply that there is a sequence M'_n, with $Im(M'_n) \subseteq V'$, converging to M. By Davis-Kahan Theorem, the eigen-spaces of M'_n must converge to those of M. However, since $\kappa(M) > 0$ and since $V \neq V'$ (and hence $\sin(V, V') > 0$, see (Bhatia, 1997)), this is impossible.

It remains to observe that since all finite dimensional norms are equivalent, $\gamma > 0$ implies $\|M - M'\|_1 \geq \gamma'' > 0$ where $\|\cdot\|_1$ is the coordinatewise ℓ_1 norm. Next, Pinsker's Inequality, (Cover & Thomas, 2006), yields a lower bound on the KL-divergence,

$$0 < \gamma'' \leq \|M - M'\|_1 \leq \sqrt{2D_{KL}(M|M')}, \tag{31}$$

where

$$D_{KL}(M|M') = \sum_{u,v} M_{u,v}\log\frac{M_{u,v}}{M'_{u,v}} \tag{32}$$

$$= \sum_{u,v} M_{u,v}\log M_{u,v} - \sum_{u,v} M_{u,v}\log M'_{u,v}. \tag{33}$$

To summarize,

$$\gamma''' := \inf_{M' \ s.t.\ Im(M')\subseteq V'} D_{KL}(M'|M) > 0. \tag{34}$$

Finally, we obtain

$$\lim_{D\to\infty} \sum_{u,v} \widehat{M}_{u,v} \log(M'_{u,v}) = \sum_{u,v} M_{u,v} \log(M'_{u,v}) \tag{35}$$

$$\leq \sum_{u,v} M_{u,v} \log(M_{u,v}) - \gamma''', \tag{36}$$

where (35) follows from the first part of the proof and (36) follows form (34) and (33). □

E. Holdout Likelihood Computation

For a document d given as a sequence of tokens $d = \{x_1, \ldots, x_{l_d}\}$, where l_d is the total number of tokens in d, recall from the main text that we denote by $c_d \in \mathbb{R}^N$ the count vector of d,

$$c_d(u) = \#\{x_i \mid x_i = u\} \text{ for } u \in \mathcal{X}. \tag{37}$$

The empirical distribution of d, denoted by $\hat{d}$ is the normalized count vector, $\hat{d} = \frac{1}{l_d} \cdot c_d \in \mathbb{R}^N$.

Given the topics returned by the model, $\{\mu_i \mid i = 1, .., T\}$ for every document d we first compute the topics assignment θ_d as follows:

$$\theta_d = \operatorname*{argmin}_{\theta} D_{KL}(\hat{d}, m(\theta)) \tag{38}$$

$$= \operatorname*{argmax}_{\theta} \sum_{u\in\mathcal{X}} \hat{d}(u) \log\{m(\theta)(u)\} \tag{39}$$

$$= \operatorname*{argmax}_{\theta} \frac{1}{l_d} \sum_{x_i \in d} \log\{m(\theta)(x_i)\}, \tag{40}$$

where $m(\theta)$ is the mixture generated by the topics and the assignment θ, $m(\theta) = \sum_t \theta(t)\mu_t$, and D_{KL} is the Kullback-Leibler divergence. Thus θ_d is the assignment such that the mixture $m(\theta_d)$ best approximates the document in KL divergence. Equivalently, θ_d is the assignment such that the mixture $m(\theta_d)$ gives the highest likelihood to the document d. This is the standard definition of likelihood for pLSA models.

Note that (38) is a convex problem in θ, and can be solved efficiently and in parallel over the documents. Solving (38) is a standard step in most pLSA and Non-Negative Matrix Factorization methods, and existing efficient implementations may be used. See for instance (Pedregosa et al., 2011).

Next, given θ_d we compute the document likelihood L_d as $L_d = \sum_{u\in\mathcal{X}} \hat{d}(u) \log m(\theta_d)(u)$ and take the overall likelihood of the holdout set of documents to be the average of L_d over all documents in the set.

F. Hardware Specifications

The LDA models were trained using an Intel Core i7-8700K processor (3.7GHz), using the collapsed Gibbs sampler algorithm implemented in the MALLET package, (McCallum, 2002). MALLET was run with $threads = 4$ option, therefore using 4 threads on 4 physical cores, and no other tasks were run on the system at the same time. The FDM models were trained using NVIDIA GeForce GTX 1080 Ti GPU, and the optimization was implemented using TensorFlow 1.9.0 and Adam SGD optimizer with 0.001 learning rate.