Practical Correlated Topic Modeling and Analysis via the Rectified Anchor Word Algorithm

Moontae Lee[1] **Sungjun Cho**[2] **David Bindel**[2] **David Mimno**[2]
[1]University of Illinois at Chicago, Microsoft Research at Redmond
[2]Cornell University
moontae@uic.edu, {sc782,bindel,mimno}@cornell.edu

Abstract

Despite great scalability on large data and their ability to understand correlations between topics, spectral topic models have not been widely used due to the absence of reliability in real data and lack of practical implementations. This paper aims to solidify the foundations of spectral topic inference and provide a practical implementation for anchor-based topic modeling. Beginning with vocabulary curation, we scrutinize every single inference step with other viable options. We also evaluate our matrix-based approach against popular alternatives including a tensor-based spectral method as well as probabilistic algorithms. Our quantitative and qualitative experiments demonstrate the power of Rectified Anchor Word algorithm in various real datasets, providing a complete guide to practical correlated topic modeling.

1 Introduction

Increasing access to massive data streams is useful only if it is equipped with proper tools to discover meaningful patterns. Topic models are capable of learning low-dimensional latent structures from groups of discrete observations, while being flexibly applicable to a wide range of modalities without human annotations. Users can assess consumer profiles by collecting purchase habits (Reisenbichler and Reutterer, 2019), shared sentiments or topics among comments in social networks (Nguyen et al., 2015), hidden genres/preferences on movie or music consumption (Lee et al., 2015), and latent communities from network snapshots (Gerlach et al., 2018). For clarity this paper sticks to the standard terms — words, documents, and topics — but the concepts generalize to various applications beyond these examples.

Traditional algorithms for topic modeling lack scalability. To learn quality topics, probabilistic algorithms such as Variational Inference (VI) or Markov Chain Monte Carlo (MCMC) require multiple passes through the input dataset until convergence, and thus struggle to process millions of documents. Online or stochastic algorithms (Hoffman et al., 2010, 2013) achieve some scalability but at the cost of sacrificing the quality of topics. As a result, the basic Latent Dirichlet Allocation (Blei et al., 2003) is still prevalent for practitioners despite various recent advances (Srivastava and Sutton, 2017; Xun et al., 2017; Xu et al., 2018). However, topics are *likely co-occurring terms* in essence. Spectral methods explicitly construct word co-occurrence moments as statistically unbiased estimators, providing alternatives to the probabilistic algorithms via moment-matching. Once the co-occurrence statistics is built with a single trivially parallelizable pass through the corpus, topic inference no longer needs to revisit individual training documents.

The Anchor Word algorithm (Arora et al., 2012, 2013; Lee et al., 2015) and tensor decomposition algorithms (Anandkumar et al., 2012a,b, 2013) factorize the second and third-order co-occurrence between pairs or triples of words, respectively, matching the corresponding posterior moments. In contrast to VI or MCMC, these spectral algorithms do not suffer from spurious local minima or slow mixing problems, learning consistently with provable guarantees under weak assumptions. However, inference is known to be sensitive to statistical noise, and its quality quickly degrades if the input data does not agree well with the underlying models similar to (Kulesza et al., 2014; Marinho, 2015). As a result, these algorithms have not been popular for real applications despite their great scalability.

This paper aims to provide a complete guide to practical correlated topic modeling. We first explain theoretical insights for spectral topic models based on the framework of Joint Stochastic Matrix Factorization (JSMF) (Lee et al., 2015). Then we

Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, pages 4991–5001,
Hong Kong, China, November 3–7, 2019. ©2019 Association for Computational Linguistics

introduce scalable implementations of the Rectified Anchor Word (RAW) algorithm and various evaluation metrics, investigating the impact of each inference step from vocabulary curation to topic inference. We also analyze quality of topics learned from annotated non-textual datasets as well as unsupervised textual corpora based on their top contributing words to the individual topics. To the best of our knowledge, this paper is the first comparative study that measures both quantitative and qualitative performance across different spectral topic models and their probabilistic counterparts.

The experimental results show that the rectification step in RAW is crucial for overcoming the *model-data mismatch* (Kulesza et al., 2014) but only needs a few iterations. The learned topics substantially outperform tensor-based methods and online VI, being comparable to expensive MCMC. Running RAW on a non-textual music dataset reveals quality genre topics, whereas the probabilistic correlated model (Blei and Lafferty, 2007) often learns overfitted topics that only maximize the co-occurrence of popular songs. To better support the community, we also provide scalable implementations in MATLAB[1] and Python[2] with full algorithmic details in the supplementary material.

2 Spectral Topic Inference

Topic modeling assumes a document representation that is sufficiently simple to allow tractable inference but also realistic to be useful. Each topic k is defined as a distribution $p_{X|Z}(\cdot|k)$ over words where $p_{X|Z}(i|k)$ is a probability to choose a word i given the topic k. Assuming there are N words in the vocabulary and K prepared topics, all topics can be compactly represented by the column-stochastic matrix $\boldsymbol{B} \in \mathbb{R}^{N \times K}$, where each column vector $\boldsymbol{b}_k \in \Delta^{N-1}$ stands for the topic k. Suppose there are M documents in a corpus which are all written by admixing some of these K topics with respect to a certain prior $\mathfrak{f}$. Then topic models explain that each document m with length n_m is written by: 1) Select a topic composition $\boldsymbol{w}_m \in \Delta^{K-1}$ with respect to $\mathfrak{f}$; 2) Write n_m words by repeatedly selecting a topic z from the composition $\boldsymbol{w}_m$ and a word x from the topic $\boldsymbol{b}_z$.

Different models adopt different priors $\mathfrak{f}$ to better explain proper admixing of topics for the given data. For example, LDA assumes $\mathfrak{f} = \text{Dir}(\boldsymbol{\alpha})$ for $\boldsymbol{\alpha} \in \mathbb{R}^K_+$ (Blei et al., 2003). In correlated topic models, $\mathfrak{f} = \text{Logit-Normal}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ (CTM) or Probit-Normal$(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ for $\boldsymbol{\mu} \in \mathbb{R}^{K-1}$, $\boldsymbol{\Sigma} \in \mathbb{R}^{(K-1)\times(K-1)}$ (Blei and Lafferty, 2007; Yu and Fokoue, 2014). These models differ only in explaining the stochastic generation of topic composition: $\boldsymbol{w}_m \sim \mathfrak{f}$. Note that entries in every column vector $\boldsymbol{b}_k$ of $\boldsymbol{B}$ are parameters to recover in our setting, whereas probabilistic topic models often put another parametric prior $\mathfrak{g}(\boldsymbol{\beta})$ from which each $\boldsymbol{b}_k$ is sampled. The form of $\mathfrak{g}$ is not as crucial in learning quality topics as the form of $\mathfrak{f}$ (Asuncion et al., 2012), and can be similarly incorporated in spectral inference by putting additional regularizers when recovering each $\boldsymbol{b}_k$ (Nguyen et al., 2014).

Let $\boldsymbol{H} \in \mathbb{R}^{N \times M}$ be the word-document matrix where the m-th column vector $\boldsymbol{h}_m$ indicates the observed term-frequencies in document m. Say $\widetilde{\boldsymbol{H}}$ is the column-normalized $\boldsymbol{H}$ where each column is $\boldsymbol{h}_m / n_m$. Topic compositions of individual documents can also be described compactly by another column-stochastic matrix $\boldsymbol{W} \in \mathbb{R}^{K \times M}$ whose m-th column vector is $\boldsymbol{w}_m \in \Delta^{K-1}$. Then the main learning task of topic models is to find the word-topic matrix $\boldsymbol{B}$ and topic-document matrix $\boldsymbol{W}$ that approximates $\widetilde{\boldsymbol{H}} \approx \boldsymbol{BW}$ with the column-stochastic constraints $\boldsymbol{B} \in \mathcal{CS}^{N \times K}$, $\boldsymbol{W} \in \mathcal{CS}^{K \times M}$. While this Non-negative Matrix factorization (NMF) is identifiable under additional sparsity constraints (Huang et al., 2014), directly applying NMF methods (Lee and Seung, 2001) produces incoherent topics despite small approximation errors (Stevens et al., 2012). $\boldsymbol{H}$ is too noisy and sparse to learn quality topics $\boldsymbol{B}$ and plausible compositions $\boldsymbol{W}$.

2.1 Joint Stochastic Matrix Factorization

Instead of directly decomposing $\widetilde{\boldsymbol{H}}$, JSMF decomposes smaller but aggregated statistics for revealing the latent topics and their correlations. Let $\boldsymbol{C} \in \mathbb{R}^{N \times N}$ be the empirical word co-occurrence matrix where $\boldsymbol{C}_{ij}$ is the joint probability $p_{X_1 X_2}(i, j)$ to observe a pair of words i and j in the corpus. Define the topic co-occurrence matrix $\boldsymbol{A} \in \mathbb{R}^{K \times K}$ where $\boldsymbol{A}_{kl}$ is the joint probability $p_{Z_1 Z_2}(k, l)$ between two latent topics k and l. Then JSMF transforms the topic modeling objective into a second-order NMF: $\boldsymbol{C} \approx \boldsymbol{BAB}^T$, which is algebraically equivalent to $p(X_1, X_2|\boldsymbol{A}; \boldsymbol{B}) = \sum_{z_1} \sum_{z_2} p(X_1|Z_1; \boldsymbol{B}) p(Z_1, Z_2|\boldsymbol{A}) p(X_2|Z_2; \boldsymbol{B})$. The question is how this formulation provides better hints to learn the latent topics $\boldsymbol{B}$ given $\boldsymbol{C}$.

[1] https://github.com/moontae/jsmf-raw

[2] https://github.com/sc782/pyJSMF-RAW

Define $\boldsymbol{x}_1 \in \mathbb{R}^N$ as a random basis vector where only a single component corresponding to one randomly drawn word from the document m is 1. Let $\boldsymbol{p}_m$ be the vector where its i-th component denotes the probability for word i to occur in the document m. Then $\boldsymbol{p}_m = \boldsymbol{B}\boldsymbol{w}_m \in \mathbb{R}^N$, satisfying

$$\boldsymbol{x}_1 \sim \text{Categorical}(\boldsymbol{p}_m) \Rightarrow \mathbb{E}[\boldsymbol{x}_1|\boldsymbol{w}_m] = \boldsymbol{B}\boldsymbol{w}_m.$$

Denote n_m consecutive random draws of words by $\{\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_{n_m}\}$, and let $\boldsymbol{h}_m = \sum_{t=1}^{n_m} \boldsymbol{x}_t$. Then

$$\boldsymbol{h}_m \sim \text{Mult}(n_m, \boldsymbol{p}_m) \Rightarrow \mathbb{E}[\boldsymbol{h}_m|\boldsymbol{w}_m] = n_m\boldsymbol{B}\boldsymbol{w}_m.$$

As explained earlier, assuming that each observed $\boldsymbol{h}_m$ follows this model does not produce statistically meaningful information toward recovering $\boldsymbol{B}$. Since different words in each document m share the same topic composition $\boldsymbol{w}_m$, however, the *cross moments* can provide useful information about co-occurring words even within a single document: $\mathbb{E}[\boldsymbol{h}_m\boldsymbol{h}_m^T|\boldsymbol{w}_m] = \mathbb{E}[\boldsymbol{h}_m|\boldsymbol{w}_m]\mathbb{E}[\boldsymbol{h}_m|\boldsymbol{w}_m]^T + \text{Cov}(\boldsymbol{h}_m|\boldsymbol{w}_m) = n_m(n_m - 1)\boldsymbol{B}\boldsymbol{w}_m\boldsymbol{w}_m^T\boldsymbol{B}^T + n_m{\cdot}\text{diag}(\boldsymbol{B}\boldsymbol{w}_m)$. Hence,

$$\frac{\mathbb{E}[\boldsymbol{h}_m\boldsymbol{h}_m^T|\boldsymbol{w}_m]-n_m{\cdot}\text{diag}(\boldsymbol{B}\boldsymbol{w}_m)}{n_m(n_m-1)}=\boldsymbol{B}\boldsymbol{w}_m\boldsymbol{w}_m^T\boldsymbol{B}^T.$$

Define the co-occurrence $\boldsymbol{C}_m$ for a single document m in terms of the observed $\boldsymbol{h}_m$:

$$\boldsymbol{C}_m = \frac{\boldsymbol{h}_m\boldsymbol{h}_m^T - \text{diag}(\boldsymbol{h}_m)}{n_m(n_m-1)}. \qquad (1)$$

If our observation $\boldsymbol{h}_m$ follows the model, then $\mathbb{E}[\boldsymbol{C}_m|\boldsymbol{w}_m] = \boldsymbol{B}\boldsymbol{w}_m\boldsymbol{w}_m\boldsymbol{B}^T$ by linearity of expectation. Then by the Law of Iterated Expectation,

$$\mathbb{E}[\boldsymbol{C}_m]{=}\mathbb{E}_{\boldsymbol{w}_m}[\mathbb{E}[\boldsymbol{C}_m|\boldsymbol{w}_m]]{=}\boldsymbol{B}\mathbb{E}_{\boldsymbol{w}_m}[\boldsymbol{w}_m\boldsymbol{w}_m^T]\boldsymbol{B}^T.$$

We can now construct the empirical word co-occurrence by averaging $\boldsymbol{C}_m$ across M documents: $\boldsymbol{C} := \frac{1}{M}\sum_{m=1}^M \boldsymbol{C}_m$. Denoting the posterior topic-topic matrix by $\boldsymbol{A}^* := \frac{1}{M}\boldsymbol{W}\boldsymbol{W}^T \in \mathbb{R}^{K\times K}$, it is proven that $\boldsymbol{A}$ is entry-wisely close to both $\boldsymbol{A}^*$ and the population moments $\mathbb{E}_{\boldsymbol{w}\sim\mathfrak{f}}[\boldsymbol{w}\boldsymbol{w}^T]$ when M is sufficiently large (Arora et al., 2012). Thus

$$\begin{aligned}\boldsymbol{C} \approx \mathbb{E}[\boldsymbol{C}] &= \boldsymbol{B}\Big(\frac{1}{M}\sum_{m=1}^M \mathbb{E}_{\boldsymbol{w}_m}[\boldsymbol{w}_m\boldsymbol{w}_m^T]\Big)\boldsymbol{B}^T \\ &= \boldsymbol{B}\mathbb{E}_{\boldsymbol{w}\sim\mathfrak{f}}[\boldsymbol{w}\boldsymbol{w}^T]\boldsymbol{B}^T \approx \boldsymbol{B}\boldsymbol{A}^*\boldsymbol{B}^T \approx \boldsymbol{B}\boldsymbol{A}\boldsymbol{B}^T.\end{aligned}$$

Once we construct the empirical moment $\boldsymbol{C}$ from the input data as an unbiased estimator of the underlying generative process, JSMF enables users to recover the correct $\boldsymbol{B}$ and $\boldsymbol{A}$ up to some precision by matching $\boldsymbol{C}$ to its posterior moments $\boldsymbol{B}\boldsymbol{A}^*\boldsymbol{B}^T$. The **separability assumption**: every topic has one specific *anchor word* that occurs only in the context of that topic, allows the model to satisfy non-negative-rank($\boldsymbol{B}$)=rank($\boldsymbol{B}$)=K, guaranteeing the existence of an identifiable factorization.

2.2 Tensor Decomposition

The separability assumption is necessary for JSMF because having only up to the second moments is not sufficient by itself to identify latent topics (Anandkumar et al., 2013). While one can release this assumption by adopting the *sufficiently scattered* condition, it maps the factorization into another NP-hard optimization problem (Huang et al., 2016). Alternatively, one can leverage third-order moments to provide sufficient statistics for identifiable topic inference (Anandkumar et al., 2012a,b). In contrast to JSMF, tensor-based algorithms first specify $\mathfrak{f}$ as a tractable parametric prior like the Dirichlet distribution. For example, if $\mathfrak{f} = \text{Dir}(\boldsymbol{\alpha})$ with $\alpha_0 = \sum_k \alpha_k$, then $\mathbb{E}^{(1st)}_{\boldsymbol{w}\sim\mathfrak{f}(\boldsymbol{\alpha})}[\boldsymbol{w}] = \boldsymbol{\alpha}/\alpha_0$, and

$$\mathbb{E}^{(2nd)}_{\boldsymbol{w}\sim\mathfrak{f}(\boldsymbol{\alpha})}[w_k w_l] = \begin{cases} \frac{\alpha_k(\alpha_k+1)}{\alpha_0(\alpha_0+1)} & (k=l) \\ \frac{\alpha_k\alpha_l}{\alpha_0(\alpha_0+1)} & (k \neq l) \end{cases}. \qquad (2)$$

It makes the marginal expectations $\mathbb{E}[\boldsymbol{x}_1]$ and $\mathbb{E}[\boldsymbol{x}_1\boldsymbol{x}_2^T]$ further parametrized by $\boldsymbol{\alpha}$.

$$\begin{aligned}\mathbb{E}[\boldsymbol{x}_1] &= \mathbb{E}_{\boldsymbol{w}_m}[\boldsymbol{x}_1|\boldsymbol{w}_m] = B\mathbb{E}[\boldsymbol{w}_m] = B\boldsymbol{\alpha}/\alpha_0 \\ \mathbb{E}[\boldsymbol{x}_1\boldsymbol{x}_2^T] &= \mathbb{E}_{\boldsymbol{w}_m}[\mathbb{E}[\boldsymbol{x}_1|\boldsymbol{w}_m]\mathbb{E}[\boldsymbol{x}_2|\boldsymbol{w}_m]^T] \\ &= \boldsymbol{B}\mathbb{E}_{\boldsymbol{w}_m}[\boldsymbol{w}_m\boldsymbol{w}_m^T]\boldsymbol{B}^T = \boldsymbol{B}\mathbb{E}^{(2nd)}_{\boldsymbol{w}\sim\mathfrak{f}(\boldsymbol{\alpha})}\boldsymbol{B}^T\end{aligned}$$

Similarly we can represent up to the third moments:

$$\begin{aligned}\mathbb{E}[\boldsymbol{x}_1 \otimes \boldsymbol{x}_2] &= \mathbb{E}^{(2nd)}_{\boldsymbol{w}\sim\mathfrak{f}(\boldsymbol{\alpha})}[\boldsymbol{w}\otimes\boldsymbol{w}](\boldsymbol{B},\boldsymbol{B}), \\ \mathbb{E}[\boldsymbol{x}_1 \otimes \boldsymbol{x}_2 \otimes \boldsymbol{x}_3] &= \mathbb{E}^{(3rd)}_{\boldsymbol{w}\sim\mathfrak{f}(\boldsymbol{\alpha})}[\boldsymbol{w}^{\otimes 3}](\boldsymbol{B},\boldsymbol{B},\boldsymbol{B}).\end{aligned}$$

By assuming $\boldsymbol{w} \sim \text{Dir}(\boldsymbol{\alpha})$, we can fortunately attain closed form expressions of all three population moments only in terms of $\boldsymbol{B}$ and $\boldsymbol{\alpha}$, allowing the non-central second and third moments to be further represented by lower-order moments and α_0 (Anandkumar et al., 2012a). Therefore, once users construct the empirical moments given the training data and choose α_0, tensor decomposition allows us to recover $\boldsymbol{B}$ and $\boldsymbol{\alpha}$ up to some precision by matching the empirical moments to these population moments.[3] But there are several caveats.

First, finding such closed-form moment combinations is not obvious. Normally all higher-order moments are necessary for learning with the general prior $\mathfrak{f}$ (Arabshahi and Anandkumar, 2017).[4] Second, $\mathbb{E}[\boldsymbol{w}^{\otimes 3}]$ should be a diagonal tensor in order to apply the popular CP-decomposition for learning topics $\boldsymbol{B}$. It means that we need to assume **uncorrelated topics** instead of the separability assumption. Whereas most large topics models are proven indeed separable (Ding et al., 2015), users of CP-decomposition can only capture weak negative correlations via the learned $\boldsymbol{\alpha}$ but depending on the user choices of α_0 and $\mathfrak{f} = \text{Dir}$. Tucker decomposition is another option for learning correlated topics, but it instead requires additional sparsity constraints on $\boldsymbol{B}$, demanding notably more parameters to be estimated (Anandkumar et al., 2013). Overall, correlated topic modeling via tensor decomposition is not as flexible as using JSMF even if we factor out the trivial difference in time and space complexities.

3 The Rectified Anchor Word Algorithm

Whereas probabilistic algorithms have an intrinsic capability to fit their models on the data that does not necessarily follow their generative processes, spectral algorithms are susceptible to model-data mismatch (Kulesza et al., 2014). The Rectified Anchor Word (RAW) algorithm (Lee et al., 2015) is the first working formalism that can learn quality topics and their correlations from real data. The overall algorithm consists of five clearly divided steps: 0) construct the word co-occurrence matrix $\boldsymbol{C}$; 1) rectify $\boldsymbol{C}$; 2) find the set of anchor words $\boldsymbol{S}$; 3) recover the topics $\boldsymbol{B}$; 4) recover the topic correlations $\boldsymbol{A}$. Each step has various algorithmic decisions that have been previously unclear. We carefully explore other viable options, providing details on efficient implementations in the supplementary material.

Step 0: Create $\boldsymbol{C}$. For spectral inference, we first construct the empirical word co-occurrence statistics as an unbiased estimator for the underlying generative process: $\boldsymbol{C} = (1/M)\sum_{m=1}^{M} \boldsymbol{C}_m$ with $\boldsymbol{C}_m$ specified in Equation (1). Due to the efficiency of anchor-based inference, the moment construction often becomes the most expensive step for large corpora, but it is trivially parallelizable as the last averaging step is the only computation that couples individual documents.

Instead of using the entire vocabulary, the standard procedure is to remove stop words and prune off rare words based on either corpus frequencies or tf-idf scores. Excluding words appearing on a majority of documents is also known to improve the quality of topics (Schofield et al., 2017a,b). Measuring the impact of vocabulary curation is not so straightforward in probabilistic topic models due to random draws in their algorithms. In contrast, spectral topic models learn topics consistently without any randomness. We later show how to pick the plausible size of vocabulary in the experiment section.

Step 1: Rectify $\boldsymbol{C}$. Rectifying the co-occurrence estimator is key to successful inference as low-rank spectral learning is highly susceptible to the mismatch between the model and the data (Lee et al., 2015). Though $\boldsymbol{C}$ is shown to be more statistically robust than $\widetilde{\boldsymbol{H}}$ (Arora et al., 2012), its empirical construction from real data hardly exhibits the proper structures of the posterior moments $\boldsymbol{B}\boldsymbol{A}^*\boldsymbol{B}^T$: low-rank ($\mathcal{LR}$), positive semidefinite ($\mathcal{PSD}$), nonnegative ($\mathcal{NN}$), and normalized ($\mathcal{NOR}$).[5] The rectification step transforms the noisy $\boldsymbol{C}$ into a desirable estimator by alternatingly projecting it to individual spaces (Lee et al., 2015). We also discover that cyclic Douglas-Rachford (DR) iterations can properly rectify $\boldsymbol{C}$, but its computational cost is almost twice as expensive as Alternating Projection (AP). Refer to the supplementary material for details.

By running a truncated eigenvalue decomposition, the first step of AP only finds K largest eigenvalues $\boldsymbol{\Lambda}_K$ and the corresponding eigenvectors $\boldsymbol{U}$ at minimal cost. It then projects $\boldsymbol{C}$ to the intersection of $\mathcal{PSD}_N$ and $\mathcal{LR}_K$ by reconstructing $\boldsymbol{U}\boldsymbol{\Lambda}_K^{+}\boldsymbol{U}^T$. The next step orthogonally projects $\boldsymbol{C}$ to $\mathcal{NOR}_N$ by subtracting the mean average entry-wisely from the desired total, 1.0. The negative entries are then zeroed out in the subsequent projection to $\mathcal{NN}_N$. While the order of projec-

[3] JSMF does not ask users to specify α_0, flexibly and transparently modeling arbitrary pairwise correlations between topics by the co-occurrence between pairs of the corresponding anchor words.

[4] They recently discover that having up to third-order moments suffices to perform CP-decomposition when $\mathfrak{f}$ is a class of Normalized Infinitely Divisible (NID) prior.

[5] Due to the diagonal penalty in (1) for the unbiased construction and the variance of the generative process, $\boldsymbol{C}$ is almost always full-rank and indefinite in finite real data.

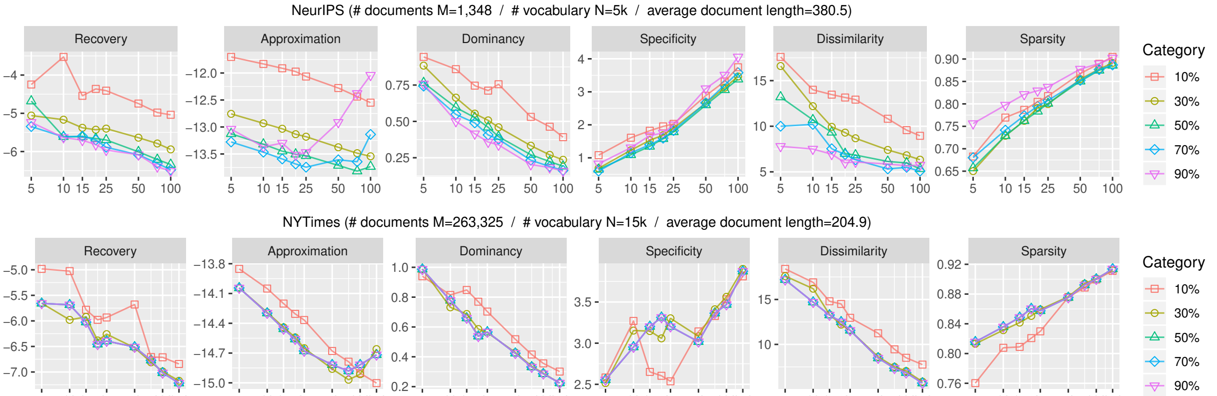

Figure 1: Vocabulary pruning assessed with AP+ADMM. A threshold of 50% implies that all words occurring in more than half of the documents are pruned. X-axis: the number of topics. Columns 1, 2, 3: lower is better / 4, 5, 6: higher is better.

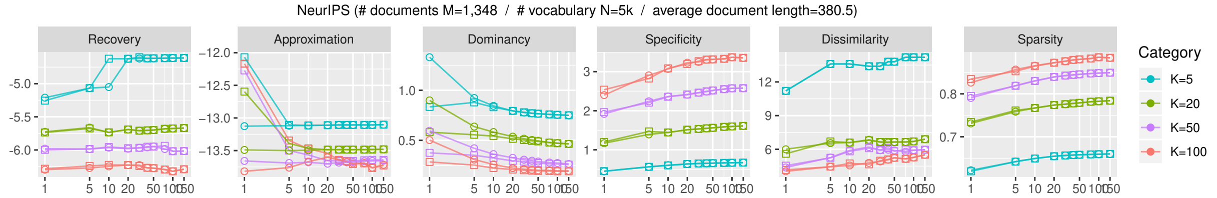

Figure 2: Alternating Projection (AP) vs Douglas-Rachford (DR). X-axis: the number of iterations in rectification. ⊟ AP+ADMM and ⊖ DR+ADMM mostly agree each other and converge within 15-20 iterations. 5 iterations are often enough.

tions in each iteration does not matter, performing a $\mathcal{NOR}_N$-projection after the loop ends helps with feasibility.[6] Note that tensor-based methods have a similar step called *whitening*, which runs a full SVD to transform the third-order moments into an orthogonal tensor for CP-decomposition.

Step 2: Find $\boldsymbol{S}$. Say $\boldsymbol{C}$ now indicates the rectified co-occurrence. Then the next step is to find the anchor words. Denoting the set of K anchor words by $\boldsymbol{S} = \{s_1, ..., s_K\}$, the separability assumption suggests: $p_{Z_1|X_1}(k'|s_k) = 1$ if $k' = k$ and $p_{Z_1|X_1}(k'|s_k) = 0$ if $k' \neq k$. Let $\overline{\boldsymbol{C}}$ be the row-normalized version of $\boldsymbol{C}$. Then by the conditional independence between a pair of words given one of their topics ($X_1 \perp X_2 | Z_1$ or Z_2) and separability, $\overline{\boldsymbol{C}}_{ij} = p_{X_2|X_1}(j|i) = \sum_{k'} p_{X_2|Z_1}(j|k')p_{Z_1|X_1}(k'|i)$. Thus $\overline{\boldsymbol{C}}_{ij} = \sum_k p_{Z_1|X_1}(k|i)\overline{\boldsymbol{C}}_{s_k,j}$, implying that every row vector of $\overline{\boldsymbol{C}}$ corresponding to a non-anchor word can be represented by a convex combination: $\sum_k p_{Z_1|X_1}(k|i) = 1$ of the rows $\{\overline{\boldsymbol{C}}_{s_k}\}$ corresponding to the anchor words $\{s_k\}$. Therefore, the inference quality depends primarily on the choice of the anchor words S, providing a clear metric for diagnosis. Note that rectification is also crucial for finding better anchors (Lee et al., 2015).

While using the pivoted QR (Arora et al., 2013) substantially expedites the running time against solving a number of LPs (Arora et al., 2012), its explicit projection of each non-anchor row to the current orthogonal complement quickly damages the sparsity of $\overline{\boldsymbol{C}}$. As a result, random projections are suggested for sizable vocabulary (Arora et al., 2013), but such projections do not maintain the joint-stochasticity of the rectified $\boldsymbol{C}$, degrading the topic quality (Lee and Mimno, 2014). Instead, we develop a sparse implicit pivoted QR that requires only $\mathcal{O}(NK)$ space to store the basis rows and performs implicit updates in $\mathcal{O}(nnz(\boldsymbol{C})K)$ time without modifying any entry in the input $\overline{\boldsymbol{C}}$.

Step 3: Recover $\boldsymbol{B}$. Provided with the set of anchor words S and the convex coefficients $\breve{\boldsymbol{B}}_{ki} = \{p_{Z_1|X_1}(k|i)\}$, one can easily recover $\boldsymbol{B}$ by applying Bayes' rule: $\boldsymbol{B}_{ik} = (\breve{\boldsymbol{B}}_{ki}c_i)/(\sum_{i'=1}^{N} \breve{\boldsymbol{B}}_{ki'}c_{i'})$, where $c_i := p_{X_1}(i)$ is the unigram probability of the word i, which is equal to $\sum_j \boldsymbol{C}_{ij}$. Hence the core of this step is to find the coefficient matrix $\breve{\boldsymbol{B}}$ by solving a Simplex-Constrained Least Squares (SCLS) that satisfy $\overline{\boldsymbol{C}}_{ij} = \sum_k \breve{\boldsymbol{B}}_{ki}\overline{\boldsymbol{C}}_{s_k,j}$ for each i. While the exponentiated gradient algorithm (ExpGrad) used in previous work (Arora et al., 2013; Lee et al., 2015) converges quickly, it is unclear how to tune the learning rate, weakening confidence in the learned topics. We propose another algorithm based on Alternating Direction Method

[6] Avoiding negative entries is useful because RAW is not a purely algebraic algorithm but uses probabilistic conditions.

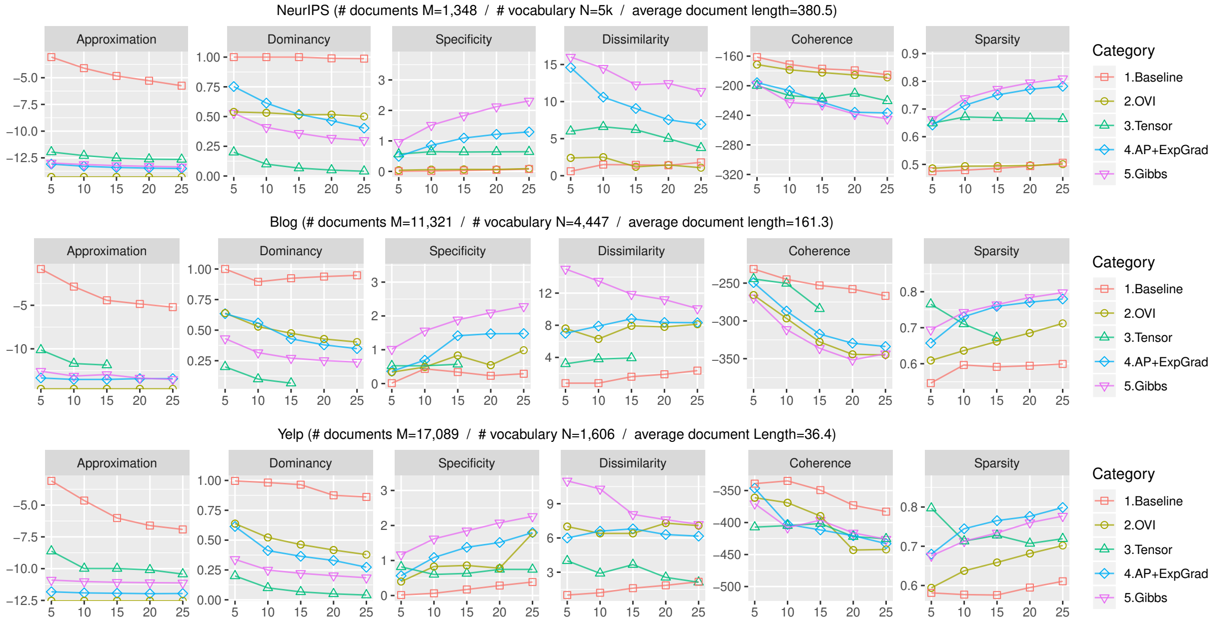

Figure 3: Quantitative results from various methods. Tensor (CP-decomposition (Anandkumar et al., 2012a)) performs better than the Baseline (Anchor Word algorithm with ExpGrad without any rectification (Arora et al., 2013)) and OVI (Online Variational Inference (Hoffman et al., 2010)), but much poorer than the AP+ExpGrad (AP-rectified Anchor Word algorithm (Lee et al., 2015)) and Gibbs (Collapsed Gibbs Sampling (Yao et al., 2009)). Surprisingly the tensor algorithm does not show consistent behavior for increasing number of topics in X-axis. Closer to Gibbs is generally better in Y-axis.

of Multipliers (ADMM), which is not sensitive to different parameter settings. Note that we also provide the Active-Set method that can solve SCLS within machine precision in our implementation, but our practical choice is ADMM due to the much higher cost of running the Active-Set method.

Step 4: Recover $\boldsymbol{A}$. The final step is to recover the topic correlation matrix $\boldsymbol{A}$, which summarizes the latent topic compositions $\boldsymbol{W}$ by $\boldsymbol{A} = (1/M)\boldsymbol{W}\boldsymbol{W}^T$. Instead of learning $\boldsymbol{W}$, Anchor Word algorithms learn the correlations $\boldsymbol{A}$ by again leveraging the separability assumption: $\sum_{l'}\big(\sum_{k'} p_{X_1|Z_1}(s_k|k')p_{Z_1Z_2}(k',l')\big)p_{X_2|Z_2}(s_l|l')$ $= p_{X_1|Z_1}(s_k|k)\big(\sum_{l'} p_{Z_1Z_2}(k,l')p_{X_2|Z_2}(s_l|l')\big)$ $= p_{X_1|Z_1}(s_k|k)p_{Z_1Z_2}(k,l)p_{X_2|Z_2}(s_l|l)$. Thus $p_{Z_1Z_2}(k,l)=p^{-1}_{X_1|Z_1}(s_k|k)p_{X_1X_2}(s_k,s_l)p^{-1}_{X_2|Z_2}(s_l|l)$, which can be simplified to $\boldsymbol{A} = \boldsymbol{B}_{S*}^{-1}\boldsymbol{C}_{SS}\boldsymbol{B}_{S*}^{-1}$. Therefore the co-occurrence of the anchor words s_k and s_l transparently captures the correlation between the pair of topics k and l. Note that the anchor words are generally rare words — in order to be the vertices of an underlying convex hull of the word co-occurrence space — whose co-occurrences are even rarer and noisier. The rectification step in JSMF effectively balances these entries (Lee et al., 2015), thereby realizing robust and transparent correlated topic inference.

4 Experimental Results

We evaluate our models of interest on two standard textual corpora: NeurIPS and NYTimes. Full papers in NeurIPS are generally longer but share a smaller vocabulary (12k), whereas massive news articles in NYTimes have medium length with the largest vocabulary (103k). Due to high complexities of tensor decomposition, we prepare two small textual datasets: Blog and Yelp. They consist of political blogs (Eisenstein et al., 2011) and business reviews (Lee and Mimno, 2014) with the smallest vocabulary (4.4k, 1.6k), respectively. In addition, we adopt two non-textual preference datasets: Movies and Songs. They include 10m movie reviews[7] and music playlists from Yes.com.[8] In contrast to textual datasets, we can retrieve genre information for Movies and Songs.[9] You can find the exact statistics of each dataset in our figures.

Evaluating topic models with held-out likelihoods or perplexities only is often misleading (Passos et al., 2011; Lee and Mimno, 2014). Instead we follow six metrics used on (Lee et al.,

[7] Movies has a vocabulary of 10.7k movies. https://grouplens.org/datasets/movielens/10m/

[8] Songs has a vocabulary of 75.3k songs. http://csinpi.github.io/lme/data_page.html

[9] Genre information are already annotated as tags in these two non-textual datasets. If missing, we scrape the information from IMDB and Discogs.com, respectively.

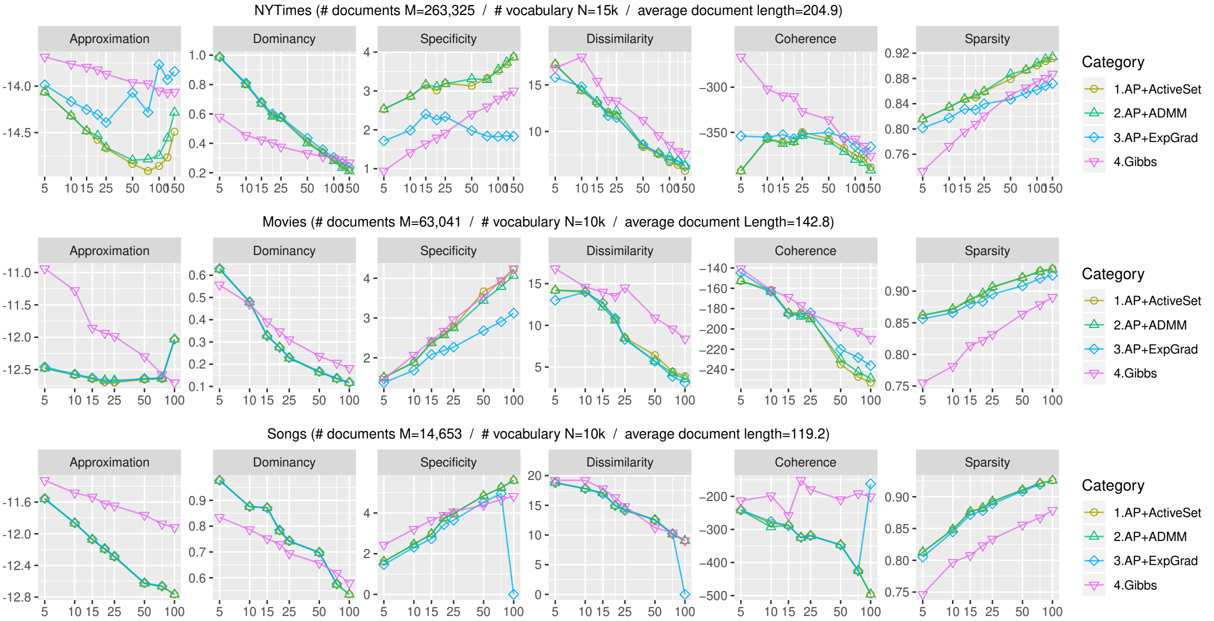

Figure 4: ExpGrad vs ADMM vs ActiveSet. Our △ AP+ADMM and ⊖ AP+ActiveSet algorithms outperform the previous state-of-the-art rectified algorithm ◇ AP+ExpGrad, being more comparable to probabilistic ▽ Gibbs sampling. Columns 1, 2: lower is better / 3, 4, 6: higher is better / 5: closer to Gibbs is better.

2015) for fair and comprehensive evaluations. **Recovery** ($\frac{1}{N}\sum_i \|\overline{\boldsymbol{C}}_i - \sum_k \breve{\boldsymbol{B}}_{ki}\overline{\boldsymbol{C}}_{s_k}\|_2$) evaluates how successfully anchor words reconstruct the word co-occurrence space. **Approximation** ($\|\boldsymbol{C} - \boldsymbol{BAB}^T\|_F$) measures the closeness between the learned factorization and the unbiased co-occurrence statistics. Note that they are measured against *the original* $\boldsymbol{C}$ rather than the rectified one, and are visualized in logarithms of base 1.8 for readability. **Dominancy** ($\frac{1}{K}\sum_k \boldsymbol{A}_{kk}$) is the average self-correlations, indirectly gauging the loss of correlations between different topics. **Specificity** ($\frac{1}{K}\sum_k \mathrm{KL}(\boldsymbol{b}_k \| \sum_i \boldsymbol{C}_{*i})$) measures the average KL-distance of each topic from the unigram distribution of the corpus. **Dissimilarity** counts the mean number of top words in each topic that do not belong to the top 20 words of other topics. **Coherence** ($\frac{1}{K}\sum_k \sum_{x_1 \neq x_2}^{x_1,x_2 \in Top_k} \log \frac{D_2(x_1,x_2)+\epsilon}{D_1(x_2)}$) penalizes any pair of top words in each topic that do not appear together in the training documents.[10] While we report Coherence, the metric could be deceptive if a model learns many duplicated topics whose top words are mostly frequent words (Huang et al., 2016). Thus following the trends of Gibbs generally implies better performance. We newly add **Sparsity** ($\frac{1}{K}\sum_k \frac{\sqrt{N}-(\|\boldsymbol{b}_k\|_1/\|\boldsymbol{b}_k\|_2)}{\sqrt{N}-1}$) (Hoyer, 2004) to gauge the average sparsity of the topics.

Vocab pruning: We experiment different document frequency cut-offs. Figure 1 shows that removing all the words that occur more than 10% the documents is too aggressive, thus showing inconsistent behavior as the number of topics grows. In contrast, using 90% cut-off saves too many words. We process vocabulary identically to (Lee et al., 2015) for fair comparison, discarding the words that appear in more than 50% of the documents. The title on top of each figure indicates the size of the pruned vocabulary with the specific statistics.

Quantitative analysis: After constructing $\boldsymbol{C}$ (Step 0), the Baseline method (Arora et al., 2013) skips to finding the anchor words (Step 2) without any rectification. While we use the exponentiated gradient (ExpGrad) method adopted in the previous work (Arora et al., 2013), we do not perform any random projection or pseudo-inverse recovery of $\boldsymbol{A}$ in order to prevent further degradation of learning quality. For methods within the framework of JSMF, we execute 150 iterations of AP or DR rectifications (Step 1) , which is equivalent to (Lee et al., 2015). However, Figure 2 shows that running only 5 iterations of AP or DR sufficiently rectifies $\boldsymbol{C}$, and 15-20 iterations yields almost identical results to 150 iterations.

Whereas our new anchor-finding method (Step 2) only improves the time/space complexity, the method choice for SCLS (Step 3) actually affects

[10] $D_2(x_1, x_2)$ means the number of training documents where two words x_1 and x_2 jointly appear. $D_1(x_2)$ counts the number of training documents that include the word x_2.

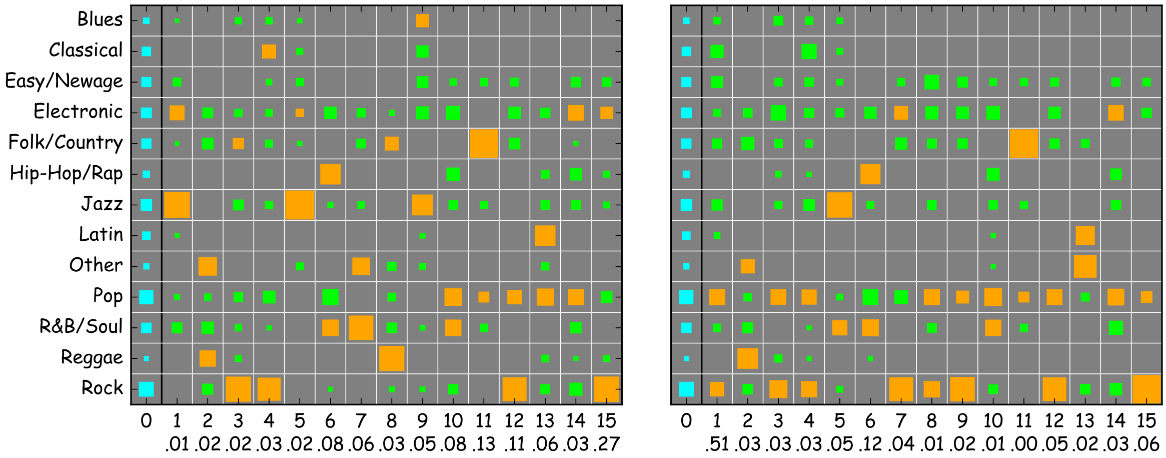

Figure 5: AP+ADMM (left) vs CTM (right). The column 0 shows the genre distribution of the entire corpus. Each column 1-15 stands for k-th topic where two most prominent genres are of orange colors. The size of each box is proportional to the relative intensity. Fractional value below each topic number on the X-axis indicates the marginal probability $p_{z_1}(k)$ of the latent topic k. AP+ADMM learns better topics that capture more coherent information about music genres.

the quality of topics. For ExpGrad, we set the learning rate as 50.0, which is the best-known from (Lee et al., 2015). For our ADMM, we set $\lambda = 1.9$ and $\gamma = 3.0$, but we also find that the algorithm is not at all sensitive to different settings. For probabilistic inference, we adopt a sufficiently mixed collapsed Gibbs Sampling (Gibbs) from the standard Mallet library[11], using 1,000 iterations after discarding the initial 200 burn-in samples. We also run Online Variational Inference (OVI) (Hoffman et al., 2010) in the standard Gensim package. Finally, we run CP-decomposition for the Tensor algorithm.[12]

Figure 3 shows that Tensor decomposition outperforms Baseline and OVI, but evaluation metrics fluctuate as the number of topics increases. We also observe that the topic distribution given a word becomes closer to uniform over growing number of topics. In contrast, AP+ExpGrad exhibits consistent behaviors as expected in spectral inference, being most comparable to Gibbs. Though SCLS (Step 3) is a convex problem, Figure 4 shows that AP+ADMM and AP+ActiveSet improve Specificity and Sparsity over AP+ExpGrad, making the learned topics even more comparable to Gibbs. We choose ADMM as our main optimization method especially because it is not only insensitive to its parameters, but also notably faster than ActiveSet. Users of ExpGrad must search through less intuitive learning rates for optimal performance, which can be different for each dataset.

Qualitative analysis: We first verify the learned topics in the NeurIPS dataset. As given in Figure 6, AP+ADMM and Gibbs learn comparable topics, while the topics learned from OVI and Tensor are not sufficiently separated: OVI repeats 'cell' and 'neuron' in different topics. Similarly, 'neuron' and 'layer' contribute to nearly every topic in Tensor regardless of hidden differences in their themes.

When running Variational CTM (Blei and Lafferty, 2007) with default parameters, the resulting topics do not show distinguishable genre associations. Most topics involve Pop and Rock, emulating the overall genre distribution of the corpus as illustrated in Figure 5. In contrast, our AP+ADMM captures three Jazz topics (T1: Electronic, T5: Pure, T9: Blues style) and four specific Rock topics (T3: Folk Rock, T4: Rock n Roll, T12: Pop style, T15: Alternative Rock). While both models discover Reggae and Latin genres, CTM's associate more with generic Other genres, whereas AP+ADMM's associate more with Folk/Country or Pop.

In addition, CTM puts spuriously high marginal topic probability on T1, which is the topic closest to the corpus genre distribution. While it can highly contribute to maximizing the likelihood of the data, an unseen playlist would most likely be classified as a mixture of Pop and Rock even if it only contains a couple of Pop or Rock songs. This also happens in Movies, explaining why we prefer using various metrics than merely measuring the held-out likelihood or perplexity. Genre association in Movies turns out to be less clear than in Songs. While songs within a playlist are likely to

[11] Gibbs is run on Java Mallet that implements time- and memory-efficient sampling with optimized multicore controls.

[12] https://github.com/FurongHuang/TensorDecomposition4TopicModeling

OVI (Hoffman et al., 2010)	**AP+ADMM (This paper)**
cell layer object neuron node	neuron circuit synaptic cell layer
layer neuron node output rbf	control action dynamic optimal controller
image cell filter neuron vector	recognition layer hidden word speech
neuron cell object map activity	cell field visual direction image
cell object recognition layer vector	gaussian noise hidden approximation matrix

Tensor (Anandkumar et al., 2012a)	**Gibbs (Yao et al., 2009)**
cell neuron field visual direction	neuron cell visual signal response
layer hidden neuron field approximation	control action policy optimal reinforcement
object image layer recognition field	recognition image object feature word
neuron layer hidden threshold synaptic	hidden net layer dynamic neuron
hidden noise gaussian layer approximation	gaussian approximation matrix bound component

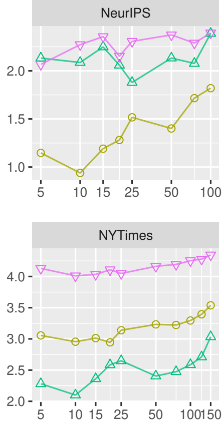

Figure 6: (Left) Each line consisted of top 5 words represents a topic from NeurIPS ($K = 5$). Both OVI and Tensor tend to repeat top words across different topics, whereas AP+ADMM discovers distinctive and meaningful topics similar to Gibbs. (Right) Scalability of the methods measured in log_{10}(seconds). Compared to ⊖ OVI and ▽ Gibbs, our △ AP+ADMM can infer quality topics while being more scalable to large corpora.

share a genre-specific theme, people often watch and review recently released movies rather than coherently consume movies within related genres. Thus topics inferred from Movies consist of year-specific topics as well as Fantasy or Sci-Fi.

5 Discussion

Runtime analyses on RAW alongside other methods demonstrate its strong scalability. In our experiment, the tensor algorithm takes 5 hours for learning 5 topics on NeurIPS and 48 days for 25 topics on Yelp. It runs indefinitely for 20-25 topics on Blog, which explains two missing data points in Figure 3. Topic learning with RAW takes less than a minute on these toy datasets and less than an hour on the largest NYTimes when using 15 iterations of AP with ADMM.

Two plots in Figure 6 further verify that RAW with AP+ADMM is approximately 4 times faster than OVI (in addition to learning better topics), and 40 times faster than Gibbs (but showing comparable results across various numbers of topics) on the NYTimes dataset. While OVI runs faster on NeurIPS, the superior quality of topics inferred with RAW far outweighs the additional cost. As running the entire pipeline of RAW takes less than 5 minutes in NeurIPS, OVI is not as competitive as RAW.[13] Lastly, the Variational CTM takes 15 minutes to learn 15 topics on Songs, but 6 hours for 50 topics. In contrast, our RAW method takes less than 10 minutes to find 50 topics on Songs.

6 Conclusion

By removing the dependency on the training documents, spectral topic modeling provides scalable formalisms for finding compact high-level structures in sparse and discrete data such as text and user-preference. The Rectified Anchor Word (RAW) algorithm enjoys its transparent and consistent behaviors, working seamlessly on various types of textual and non-textual real datasets. In particular, our AP+ADMM algorithm outperforms the previous AP+ExpGrad (Lee et al., 2015), being more comparable to Gibbs sampling and less sensitive to parameters.

Through this paper, we closely investigate each step of inference with various algorithmic decisions. Proper pruning of vocabulary is shown necessary, and rectification is proven crucial for reliable topic inference under the model-data mismatch. Joint Stochastic Matrix Factorization (JSMF) with the rectification better models arbitrary pairwise topic correlations at lower cost than probabilistic correlated topic model and tensor decomposition. We hope that our work, built upon the theoretical insights on spectral inference, provide a complete guide to correlated topic modeling for both researchers and practitioners.

Acknowledgements

This work was completed when the first author was visiting Microsoft Research at Redmond. We thank both the department of Information and Decision Sciences in UIC Business School and the Deep Learning group in Microsoft Research. They provided useful resources and thoughtful advice.

[13] Note that slight fluctuations in Figure 6 are due to the load-balancing from the job queue on our high-performance computing cluster. For precision, we draw these two panels by averaging the running times from 10 different trials.

References

Anima Anandkumar, Dean P. Foster, Daniel Hsu, Sham Kakade, and Yi-Kai Liu. 2012a. A spectral algorithm for latent Dirichlet allocation. In *NIPS*.

Animashree Anandkumar, Daniel J. Hsu, Majid Janzamin, and Sham Kakade. 2013. When are overcomplete topic models identifiable? uniqueness of tensor tucker decompositions with structured sparsity.

Animashree Anandkumar, Sham M Kakade, Dean P Foster, Yi-Kai Liu, and Daniel Hsu. 2012b. Two svds suffice: Spectral decompositions for probabilistic topic modeling and latent dirichlet allocation.

F. Arabshahi and A. Anandkumar. 2017. Spectral methods for correlated topic models. *AISTATS*.

S. Arora, R. Ge, and A. Moitra. 2012. Learning topic models – going beyond SVD. In *FOCS*.

Sanjeev Arora, Rong Ge, Yonatan Halpern, David Mimno, Ankur Moitra, David Sontag, Yichen Wu, and Michael Zhu. 2013. A practical algorithm for topic modeling with provable guarantees. In *ICML*.

Arthur U. Asuncion, Max Welling, Padhraic Smyth, and Yee Whye Teh. 2012. On smoothing and inference for topic models. *CoRR*, abs/1205.2662.

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

D. Blei, A. Ng, and M. Jordan. 2003. Latent Dirichlet allocation. *JMLR*.

Weicong Ding, Prakash Ishwar, and Venkatesh Saligrama. 2015. Most large topic models are approximately separable. In *ITA, 2015*, pages 199–203. IEEE.

Jacob Eisenstein, Duen Horng Chau, Aniket Kittur, and Eric P. Xing. 2011. Topicviz: Semantic navigation of document collections. *CoRR*, abs/1110.6200.

Martin Gerlach, Tiago P. Peixoto, and Eduardo G. Altmann. 2018. A network approach to topic models. *Science Advances*, 4(7).

Matthew D. Hoffman, David M. Blei, and Francis Bach. 2010. Online learning for latent dirichlet allocation. In *NIPS*.

Matthew D Hoffman, David M Blei, Chong Wang, and John Paisley. 2013. Stochastic variational inference. *The Journal of Machine Learning Research*, 14(1):1303–1347.

Patrik O. Hoyer. 2004. Non-negative matrix factorization with sparseness constraints. *JMLR*.

K. Huang, N. D. Sidiropoulos, and A. Swami. 2014. Non-negative matrix factorization revisited: Uniqueness and algorithm for symmetric decomposition. *IEEE Transactions on Signal Processing*.

Kejun Huang, Xiao Fu, and Nikolaos D. Sidiropoulos. 2016. Anchor-free correlated topic modeling: Identifiability and algorithm. In *NIPS*.

Alex Kulesza, N Raj Rao, and Satinder Singh. 2014. Low-rank spectral learning. In *AISTATS*.

Daniel D. Lee and H. Sebastian Seung. 2001. Algorithms for non-negative matrix factorization. In *NIPS*.

Moontae Lee, David Bindel, and David Mimno. 2015. Robust spectral inference for joint stochastic matrix factorization. In *NIPS*.

Moontae Lee and David Mimno. 2014. Low-dimensional embeddings for interpretable anchor-based topic inference. In *EMNLP*. Association for Computational Linguistics.

Zita Marinho. 2015. Moment-based algorithms for structured prediction.

Thang Nguyen, Yuening Hu, and Jordan Boyd-Graber. 2014. Anchors regularized: Adding robustness and extensibility to scalable topic-modeling algorithms. In *ACL*.

Thien Hai Nguyen, Kiyoaki Shirai, and Julien Velcin. 2015. Sentiment analysis on social media for stock movement prediction. *Expert Systems with Applications*, 42(24):9603 – 9611.

Alexandre Passos, Hanna Wallach, and Andrew McCallum. 2011. Correlations and anticorrelations in lda inference. In *NIPS*.

Martin Reisenbichler and Thomas Reutterer. 2019. Topic modeling in marketing: recent advances and research opportunities. *Journal of Business Economics*, 89(3):327–356.

Alexandra Schofield, Måns Magnusson, and D Mimno. 2017a. Understanding text pre-processing for latent dirichlet allocation. In *Proceedings of the 15th conference of the European chapter of the Association for Computational Linguistics*, volume 2, pages 432–436.

Alexandra Schofield, Måns Magnusson, and David Mimno. 2017b. Pulling out the stops: Rethinking stopword removal for topic models. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 432–436.

Akash Srivastava and Charles A. Sutton. 2017. Autoencoding variational inference for topic models. In *ICLR*.

Keith Stevens, Philip Kegelmeyer, David Andrzejewski, and David Buttler. 2012. Exploring topic coherence over many models and many topics. In *EMNLP-CoNLL*.

Hongteng Xu, Wenlin Wang, Wei Liu, and Lawrence Carin. 2018. Distilled wasserstein learning for word embedding and topic modeling. In *Advances in Neural Information Processing Systems 31*, pages 1716–1725.

Guangxu Xun, Yaliang Li, Wayne Xin Zhao, Jing Gao, and Aidong Zhang. 2017. A correlated topic model using word embeddings. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, IJCAI'17, pages 4207–4213.

Limin Yao, David Mimno, and Andrew McCallum. 2009. Efficient methods for topic model inference on streaming document collections. In *KDD*.

Xingchen Yu and Ernest Fokoue. 2014. Probit normal correlated topic model. In *Open Journal of Statistics*, pages 879–888.