Proceedings of the 12th INFORMS Workshop on Data Mining and Decision Analytics (Houston - 2017)
C. Iyigun, R. Moghaddass, M. Samoran, eds.

From Correlation to Hierarchy: Practical Topic Modeling via Spectral Inference

Moontae Lee
Computer Science, Cornell University, moontae@cs.cornell.edu

David Bindel
Computer Science, Cornell University, bindel@cs.cornell.edu

David Mimno
Information Science, Cornell University, mimno@cornell.edu

Abstract.

Topic models were originally applied in text analysis for extracting high-level themes from documents, but they work equally well in any setting where users select items from an inventory. Recent work in spectral topic modeling has provided algorithms that operate only on easily-collected summary statistics, rather than exhaustively iterating over the full dataset. The "anchor word" algorithms learn topics by decomposing the co-occurrence between pairs of words into a matrix of topics over words and a matrix of relations between topics. While these algorithms provide transparent inference and provable guarantees in addition to scalability, there are several known issues: inference can be infeasible for large vocabularies and cannot learn quality topics on noisy real data with high sensitivity to learning parameters . In this paper, we solidify the foundations of anchor-based spectral inference and propose practical algorithms that can efficiently tackle each of these problems within the framework of Joint Stochastic Matrix Factorization. These algorithms preserve the provable guarantees and scalability of earlier algorithms, but are more consistent and more stable in identifying quality topics. In addition, this algorithm can also consider and learn meaningful correlations between topics, enabling correlated and hierarchical models. We demonstrate these methods on two text corpora, a corpus of user movie ratings, and a corpus of song playlists.

Keywords: correlated topic modeling, hierarchical text mining, anchor word algorithm, joint stochastic matrix factorization, spectral inference, method of moments.

1 Introduction

Increasing access to massive data streams can be a strategic asset to businesses and industries, but only if they are capable of extracting meaningful patterns. Many of these data involve groups of discrete observations: social media posts consist of words from a vocabulary, shopping carts consists of items from an inventory, network nodes consist of links to other nodes. However discrete

observation is difficult to work with because they are sparse and high-dimensional where groups of observations often combine multiple intentions. Statistical topic modeling is a powerful tool that learns low-dimensional latent structures that can succinctly characterize data. Topic models operate on raw data without requiring any order information or additional human annotation, so users can flexibly apply these models not only to text articles and image streams, but also to customer preferences and social network. For each modality, topics can be common themes that underlie text articles [1], meaningful features/segments that characterize certain image streams [2], hidden clusters of preferences like genres on music/movie consumption [3], and latent communities from network snapshots [4]. For clarity this paper keeps using the standard terms — words, documents, and topics — but the concepts generalize to many applications as enumerated.

Standard probabilistic algorithms for topic models have difficulty scaling to millions and billions of documents. While a number of different topic models have been developed for various applications [5, 6, 7], their training relies mostly on likelihood-based inference such as Variational Bayes (VB) or Markov Chain Monte Carlo (MCMC). In order to learn quality topics, these traditional methods need to iterate through input datasets multiple times until parameters converge, so handling large volumes of data is laborius. The Hathi Trust and their collaborators recently released per-page word counts which consist of 5.1 billion pages from 13.8 million volumes.[1] Giant retailers such as Amazon and Walmart process more than 5 million transactions per day.[2] While probabilistic algorithms are commonly used in practice for their simplicity and relatively tolerable learning cost, scaling to massive datasets requires increasingly complicated engineering.

Spectral inference is a newer alternative to likelihood-based training. Recall that topics are *clusters of frequently co-occurring words*. Instead of operating on the original documents, spectral methods explicitly construct word co-occurrence moments as statistically unbiased estimators by summing over documents. Then they perform moment-matching by decomposing the co-occurrence statistics into specific forms from which the latent topics are revealed without revisiting the original documents. The Anchor Word algorithms [8, 9, 3] factorizes the second-order matrix between pairs of words, matching its posterior moments. Tensor decomposition algorithms [10, 11, 12] factorize the third-order tensor among triples of words, matching its population moments. The co-occurrence statistics are easily calculated through a single, trivially parallelizable pass, resulting in greater scalability. In contrast to the likelihood-based inference such as VB or MCMC, in addition, spectral inference does not suffer from spurious local minima or slow mixing problems, thereby learning transparently with provable guarantees under weak assumptions [8, 10].

[1] `https://wiki.htrc.illinois.edu/display/COM/Extracted+Features+Dataset`
[2] `http://www.economist.com/node/15557443`

This paper focuses on anchor-based inference, which adopts the *separability assumption:* every topic has one specific anchor word that occurs only in the context of that topic. Though it is stronger than the minimal necessary condition for identifying topics [13], most large topic models are proven almost separable [14], and anchor-based inference has many advantages over other approaches. First, no parametric assumption is necessary for prior information of topics. Putting a prior for topic distributions is the crux of successful topic modeling [15]. The separability assumption enables anchor-based topic models to work with arbitrarily correlated topics, flexibly generalizing various traditional choices such as Dirichlet distribution [16] or Logistic-Normal distribution [5]. Second, the formulation naturally learns the correlations between topics in terms of the co-occurrences between the corresponding anchor words [3]. If relaxing this assumption [17, 13], the correlation information becomes less transparent. Third, this assumption clearly divides the inference procedures into 1) anchor finding, 2) topic recovery, and 3) correlation recovery, allowing users to diagnose the origin of inferior performance. Moreover, its second-order nature enjoys efficient time and space complexities comparing to the third-order tensor models.

However, anchor-based inference also has several known problems. Existing anchor finding algorithm does not scale well with the size of vocabulary, requiring a random projection at additional costs of running time and inferior topic quality [18]. The most popular algorithm [9] is capable of learning meaningful topics, only if the number of topics is sufficiently large (e.g., at least more than 50). Even with the enough numbers of topics, topic quality in real data is notably worse than the probabilistic counterparts given by MCMC inference like Gibbs Sampling. This is fundamentally because moment-matching methods are too sensitive to the statistical noise and the mismatch between model and data comparing to likelihood-based training [19]. As a result, the popular algorithm works well on synthetic data which are generated from the model, but the performance highly degrades in real data as they never follow the underlying model.

Joint Stochastic Matrix Factorization (JSMF) resolves this problem by rectifying the co-occurrence statistics based on the geometry of their posteriors. Removing noise and completing missing information, JSMF handles the model-data mismatch, thus enabling users to discover quality topics in every setting [3]. In this paper, we provide theoretical insights for spectral topic modeling and practical algorithms for the efficient implementation of JSMF. The experimental result shows that our model further improves the scalability and the performance from [3]. We also compare the topic correlations learned from our model to the results from the corresponding probabilistic model [5], demonstrating the surprising capabilities of anchor-topic modeling for the first time. We further propose a novel approach for hierarchical topic modeling that maximally reuses the learned topic correlations and the rectification, achieving supertopics from the subtopic co-occurrence.

2 Foundations of Spectral Topic Inference

Topic modeling assumes a document representation which is sufficiently simple to allow for tractable inference but sufficiently realistic to be useful. Each "topic" k is defined as a distribution $p(x|z{=}k)$ over words where $p(x = i|z = k)$ is a probability to choose a word i given the topic k. Assuming there are N words in the vocabulary and K topics,[3] 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 of the 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 use a different prior $\mathfrak{f}$ to better explain proper admixing of topics for the given data. For example, LDA assumes $\mathfrak{f}{=}\mathrm{Dir}(\boldsymbol{\alpha})$ for $\boldsymbol{\alpha}{\in}\mathbb{R}^K_+$ [16]. In the correlated topic model (CTM) $\mathfrak{f} = \mathcal{LN}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ for $\boldsymbol{\mu} \in \mathbb{R}^{K-1}, \boldsymbol{\Sigma} \in \mathbb{R}^{(K-1)\times(K-1)}$ [5]. In the Pachinko allocation model $\mathfrak{f}$ is not a parametric family, but a DAG-induced distribution, which is not always uniquely identifiable [6]. 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 $\mathfrak{f}$ [15], and can be similarly incorporated in spectral inference by putting additional regularizers when recovering each $\boldsymbol{b}_k$ [20].

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 the document 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 $\boldsymbol{w}_m{\in}\Delta^{K-1}$. The main learning task of topic models is to recover the word-topic matrix $\boldsymbol{B}$ and to infer the topic-document matrix $\boldsymbol{W}$. For certain parametric families such as $\mathfrak{f}{=}\mathrm{Dir}(\boldsymbol{\alpha})$, one can recover the hyperparameter $\boldsymbol{\alpha}$ [8].[4] Say $\widetilde{\boldsymbol{H}}$ is the column-normalized $\boldsymbol{H}$ where each column is $\boldsymbol{h}_m/n_m$. Then the learning task of topic models can be viewed as an approximate Non-negative Matrix Factorization (NMF): $\widetilde{\boldsymbol{H}} \approx \boldsymbol{BW}$, which minimizes $\frac{1}{2}\|\widetilde{\boldsymbol{H}} - \boldsymbol{BW}\|^2_F$ with the column-stochastic constraints $\boldsymbol{B} \in \mathcal{CS}^{N\times K}, \boldsymbol{W} \in \mathcal{CS}^{K\times M}$. While this factorization could be identifiable under some additional sparsity constraints [21], solving it by the NMF methods like [22] produces incoherent topics even if the approximation error is small enough [23]. This is essentially because $\boldsymbol{H}$ itself is too noisy statistics where only a tiny subset of vocabulary appears for each document.

[3] K is considerably smaller than N in the general settings. If $K > N$, it is called *overcomplete* [12].

[4] Note that we here try to distinguish "recover" from "infer". While one can also infer $(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ by Gibbs sampling when $\mathfrak{f} = \mathcal{LN}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, it is unlikely to recover these parameters within a provable precision.

2.1 Joint Stochastic Matrix Factorization

Instead of directly decomposing the giant and noisy $\widetilde{\boldsymbol{H}}$, JSMF decomposes the smaller and aggregated statistics toward 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 $p(x_1 = i, x_2 = j)$ 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 $p(z_1 = k, z_2 = l)$ between two latent topics k and l. Then JSMF transforms topic modeling objective into a second-order non-negative matrix factorization:[5]

$$\boldsymbol{C} \approx \boldsymbol{B}\boldsymbol{A}\boldsymbol{B}^T \iff p(x_1, x_2|A; B) = \sum_{z_1} \sum_{z_2} p(x_1|z_1; B)p(z_1, z_2|A)p(x_2|z_2; B). \tag{1}$$

The question is how this formulation provides better hints to learn the latent topics $\boldsymbol{B}$ from $\boldsymbol{C}$. 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 components means the probability for the 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 draws of a word by $\{\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_{n_m}\}$, and say $\boldsymbol{h}_m = \sum_{t=1}^{n_m} \boldsymbol{x}_t$. Then $\boldsymbol{h}_m \sim \text{Multinomial}(n_m, \boldsymbol{p}_m) \Rightarrow \mathbb{E}[\boldsymbol{h}_m|\boldsymbol{w}_m] = n_m\boldsymbol{B}\boldsymbol{w}_m$. 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: $\mathbb{E}[\boldsymbol{h}_m\boldsymbol{h}_m^T|\boldsymbol{w}_m] = \mathbb{E}[\boldsymbol{h}_m|\boldsymbol{w}_m]\mathbb{E}[\boldsymbol{h}_m|\boldsymbol{w}_m]^T + \text{Cov}(\boldsymbol{h}_m|\boldsymbol{w}_m) = n_m(n_m - 1)\boldsymbol{B}\boldsymbol{w}_m\boldsymbol{w}_m^T\boldsymbol{B}^T + n_m \cdot diag(\boldsymbol{B}\boldsymbol{w}_m)$. Hence,

$$\frac{\mathbb{E}[\boldsymbol{h}_m\boldsymbol{h}_m^T|\boldsymbol{w}_m] - n_m \cdot diag(\boldsymbol{B}\boldsymbol{w}_m)}{n_m(n_m - 1)} = \boldsymbol{B}\boldsymbol{w}_m\boldsymbol{w}_m^T\boldsymbol{B}^T. \quad \text{Define } \boldsymbol{C}_m := \frac{\boldsymbol{h}_m\boldsymbol{h}_m^T - diag(\boldsymbol{h}_m)}{n_m(n_m - 1)} \tag{2}$$

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

If $\boldsymbol{h}_m$ follows our model, then $\mathbb{E}[\boldsymbol{C}_m|\boldsymbol{w}_m] = \boldsymbol{B}\boldsymbol{w}_m\boldsymbol{w}_m\boldsymbol{B}^T$ by the linearity of expectation. Thus $\mathbb{E}[\boldsymbol{C}_m] = \mathbb{E}_{\boldsymbol{w}_m}[\mathbb{E}[\boldsymbol{C}_m|\boldsymbol{w}_m]] = \boldsymbol{B}\mathbb{E}_{\boldsymbol{w}_m}[\boldsymbol{w}_m\boldsymbol{w}_m^T]\boldsymbol{B}^T$ due to the Law of Iterated Expectation. 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 $\boldsymbol{A}^* := \frac{1}{M}\boldsymbol{W}\boldsymbol{W}^T \in \mathbb{R}^{K \times K}$, the result of the second-order decomposition $\boldsymbol{A}$ 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 [8]. Thus once the set of the training documents is properly large, $\boldsymbol{C} \approx \mathbb{E}[\boldsymbol{C}] = \boldsymbol{B}(\frac{1}{M}\sum_{m=1}^{M} \mathbb{E}_{\boldsymbol{w}_m}[\boldsymbol{w}_m\boldsymbol{w}_m^T])\boldsymbol{B}^T = \boldsymbol{B}\mathbb{E}_{\boldsymbol{w} \sim \mathfrak{f}}[\boldsymbol{w}\boldsymbol{w}^T]\boldsymbol{B}^T \approx \boldsymbol{B}\boldsymbol{A}^*\boldsymbol{B}^T \approx \boldsymbol{B}\boldsymbol{A}\boldsymbol{B}^T$. It implies that we can recover the correct $\boldsymbol{B}$ and $\boldsymbol{A}$ up to some precision by matching the

[5] While $\boldsymbol{A}$ is neither correlation nor covariance, we keep calling it *topic correlations* following the naming convention in [8, 9, 3, 13]. Note that the correlations/covariance $\propto p(z_1, z_2) - p(z_1)p(z_2)$, being inducible from $\boldsymbol{A}$.

second-order unbiased estimator $\boldsymbol{C}$ to the posterior moments $\boldsymbol{B}\boldsymbol{A}^*\boldsymbol{B}^T$, which is realized by the JSMF. For some known parametric families like a Dirichlet distribution, furthermore, we can also recover the hyperparameter $\boldsymbol{\alpha}$ by matching the recovered topic-topic matrix $\boldsymbol{A}$ to the parametric second moments of $\mathfrak{f}(\boldsymbol{\alpha})$ rather than performing inference [8]. The separability assumption implies *non-negative rank*$(\boldsymbol{B})$ = *rank*$(\boldsymbol{B}) = K$, guaranteeing the existence of an identifiable factorization.

3 The Rectified Anchor Word Algorithm

The first Anchor Word algorithm [8] works only in theory: many entries in $\boldsymbol{B}$ that should be probabilities are negative due to the purely algebraic estimation through the matrix inversion. While probabilistic inference in [9] fixes some issues, the algorithm works only for large enough number of topics, and the learned topic correlations $\boldsymbol{A}$ still consists of many negative entries whose magnitudes are neither negligible nor interpretable. The Rectified Anchor Word (RAW) algorithm [3] is the first version that can learn quality topics and their correlations in every configuration by rectifying model-data mismatch within the JSMF. Due to the separability assumption, the overall algorithm consists of four clearly divided steps: 1) construct the word co-occurrence matrix $\boldsymbol{C}$ and rectify it; 2) find the set of anchor words S; 3) recover the topics $\boldsymbol{B}$; 4) recover topic correlations $\boldsymbol{A}$ and hyperparameter $\boldsymbol{\alpha}$ if available. We introduce scalable implementations of each step including new methods that further improve the state-of-the-art performance demonstrated in [3].

Step 0: Create $\boldsymbol{C}$. To run the RAW algorithm, we first need to construct the empirical word co-occurrence matrix $\boldsymbol{C}$ as an unbiased estimator of the second-order moments: $\boldsymbol{C} = (1/M)\sum_{m=1}^{M}\boldsymbol{C}_m$ with $\boldsymbol{C}_m$ specified in Equation (2). Due to the efficiency of the anchor-based topic inference, the moment construction often becomes the most expansive step if the data consists of many documents. Note that this step is trivially parallelizable for each document because $\boldsymbol{C}$ is an simple average of $\boldsymbol{C}_m$, and the averaging is the only between-documents computation at the end.

Step 1: Rectify $\boldsymbol{C}$. The typical failure mode of moment-matching is mismatch between the model and the data, so rectifying the co-occurrence estimator is the key to successful inference [3]. Though $\boldsymbol{C}$ is shown to be statistically more stable than $\widetilde{\boldsymbol{H}}$ [8], it does not exhibit the geometric structures of the posterior moments $\boldsymbol{B}\boldsymbol{A}^*\boldsymbol{B}^T$: a low-rank, positive semidefinite ($\mathcal{PSD}$), nonnegative ($\mathcal{NN}$), and normalized ($\mathcal{NOR}$).[6] The rectification step transforms the noisy $\boldsymbol{C}$ into the desirable estimator via alternately projecting to each of the spaces until convergence [3]. It first runs the truncated eigenvalue decomposition, finding only the K largest positive eigenvalues $\boldsymbol{\Lambda}_K$ and the corresponding eigenvectors $\boldsymbol{U}$.[7] After $\mathcal{PSD}_N$-projection by the reconstruction: $\boldsymbol{U}\boldsymbol{\Lambda}_K^+\boldsymbol{U}^T$,

[6] Due to the diagonal penalty in (2) and the variance, $\boldsymbol{C}$ is almost always full-rank and indefinite in real data.

[7] Since $K \ll N$, running the truncated decomposition is incomparably cheaper than the full decomposition.

Algorithm 1 Alternate Projection (AP)

def RECTIFY-C($\boldsymbol{C}, K$)

$\boldsymbol{C}_{\mathcal{NN}} \leftarrow \boldsymbol{C}$
repeat
  $(\boldsymbol{U}, \boldsymbol{\Lambda}_K) = \text{TRUNCATED-EIG}(\boldsymbol{C}_{\mathcal{NN}}, K)$
  $\boldsymbol{\Lambda}_K^+ \leftarrow diag(\max\{diag(\boldsymbol{\Lambda}_K),\ 0\})$
  $\boldsymbol{C}_{\mathcal{PSD}} \leftarrow \boldsymbol{U}\boldsymbol{\Lambda}_K^+\boldsymbol{U}^T$
  $\boldsymbol{C}_{\mathcal{NOR}} \leftarrow \boldsymbol{C}_{\mathcal{PSD}} + \frac{1-\sum_{i,j}\boldsymbol{C}_{\mathcal{PSD}}(i,j)}{N^2}\mathbf{1}\mathbf{1}^T$
  $\boldsymbol{C}_{\mathcal{NN}} \leftarrow \max\{\boldsymbol{C}_{\mathcal{NOR}},\ 0\}$
until the convergence of $\boldsymbol{C}_{\mathcal{NN}}$
return $\boldsymbol{C} \leftarrow \boldsymbol{C}_{\mathcal{NN}}/(\sum_{i,j}\boldsymbol{C}_{\mathcal{NN}}(i,j))$

($diag(\cdot)$ is the Matlab-style operation that maps the input vector to the diagonal matrix or extracts the diagonal vector from the input matrix.)

Algorithm 2 Sparse Implicit Column-pivoted QR

def FIND-S($\overline{\boldsymbol{C}}, K$)

$(\boldsymbol{P},\ \boldsymbol{Q},\ S,\ \boldsymbol{r}) \leftarrow (\overline{\boldsymbol{C}}^T,\ \mathbf{0}^{N\times K}, \varnothing,\ \mathbf{0}^K)$
$\boldsymbol{u} \leftarrow (\|\boldsymbol{p}_1\|_2^2, ..., \|\boldsymbol{p}_N\|_2^2) \in \mathbb{R}^{1\times N}$
for $k = 1$ to K **do**
  $n \leftarrow \text{argmax}_{1\le i\le N}\ u_i$
  $(S,\ \boldsymbol{q}_k,\ r_k) \leftarrow (S \cup \{n\},\ \boldsymbol{p}_n,\ \sqrt{u_n})$
  $\boldsymbol{q}_k \leftarrow (\boldsymbol{q}_k - \sum_{l=1}^{k-1}\langle \boldsymbol{q}_l, \boldsymbol{p}_n\rangle \boldsymbol{q}_l)/r_k$
  $\boldsymbol{u} \leftarrow \boldsymbol{u} - (\boldsymbol{q}_k^T\boldsymbol{P}) \circ (\boldsymbol{q}_k^T\boldsymbol{P})$
end for
return $(S, \boldsymbol{r})$

($\circ : \mathbb{R}^N \times \mathbb{R}^N \to \mathbb{R}^N$ is the Hadamard Product that yields the same dimensional vector by entry-wise multiplication of the two operand vectors.)

it performs next orthogonal projection to $\mathcal{NOR}_N$ by subtracting mean overage entry-wisely from the desired total, which is 1. This procedure could change some entries into the negative values, being later zeroed out in the subsequent projection to $\mathcal{NN}_N$. While the sequence of projections does not matter, performing $\mathcal{NN}_N$-projection at the end of the loop helps the feasibility.[8]

Step 2: Find S. Once the rectified co-occurrence $\boldsymbol{C}$ is ready, the next step is to find the anchor words. If denoting the set of the K anchor words by $S = \{s_1, ..., s_K\}$, the separability assumption means: $p(z = k'|x = s_k) = 1$ if $k' = k$ and $p(z = k'|x = s_k) = 0$ 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 the separability, $\overline{\boldsymbol{C}}_{ij} = p(x_2 = j|x_1 = i) = \sum_{k'} p(x_2 = j|z_1 = k')p(z_1 = k'|x_1 = i)$. So, $\overline{\boldsymbol{C}}_{s_k,j} = p(x_2 = j|z_1 = k)$. Thus $\overline{\boldsymbol{C}}_{ij} = \sum_k p(z = k|x = i)\overline{\boldsymbol{C}}_{s_k,j}$, implying that every row vector of $\overline{\boldsymbol{C}}$ corresponding to a non-anchor word can be represented by a convex combination (i.e., the coefficient sum $\sum_k p(z = k|x = i) = 1$) of the rows corresponding to the anchor words. Therefore the learning performance depends primarily on the quality of the representatives S, providing users of a clear metric for diagnosis. Since the rectification is proven crucial for finding better anchors [3], it again articulates the importance of the rectification step.

The pivoted QR greedily finds the K best representative vectors by repeatedly: 1) selecting the farthest vector and 2) projecting the remaining vectors to the orthogonal complement. While using the pivoted QR [9] notably expedites the running time from solving a number of LPs [8], that algorithm cannot maintain the sparsity of $\overline{\boldsymbol{C}}$ because it explicitly projects every non-anchor row to

[8] After the loop, we normalize the co-occurrence by dividing all entries by their sum. Such normalization is neither necessary nor changes the co-occurrence much, but it helps us consistently compare the experimental results.

Algorithm 3 ADMM by Douglas-Rachford (DR)

def RECOVER-B($\overline{\boldsymbol{C}}, \boldsymbol{c}, S, \lambda, \gamma$)
1: $(\boldsymbol{U},\ \breve{\boldsymbol{B}},\ \boldsymbol{B}) \leftarrow ((\overline{\boldsymbol{C}}_{S*})^T,\ \boldsymbol{0}^{K\times N},\ \boldsymbol{0}^{N\times K})$
2: $\breve{\boldsymbol{B}}_{*S} \leftarrow \boldsymbol{I}_K$ ($\boldsymbol{I}_K = K \times K$ identity matrix)
3: $\boldsymbol{F} \leftarrow (\gamma \boldsymbol{U}^T\boldsymbol{U} + \boldsymbol{I}_K)^{-1}$
4: **for each** $i \in \{1, ..., N\} \setminus S$ (in parallel) **do**
5: $(\boldsymbol{v},\ \boldsymbol{f}) \leftarrow ((\overline{\boldsymbol{C}}_{i*})^T,\ \gamma \boldsymbol{U}^T\boldsymbol{v})$
6: $\boldsymbol{y}^{(0)} \leftarrow \Pi_{\Delta^{K-1}}\left((\boldsymbol{U}^T\boldsymbol{U})^{-1}(\boldsymbol{f}/\gamma)\right)$
7: $\boldsymbol{q}^{(0)} \leftarrow \boldsymbol{y}^{(0)}$
8: **repeat**
9: $\boldsymbol{p}^{(t)} \leftarrow \boldsymbol{F}(2\boldsymbol{y}^{(t-1)} - \boldsymbol{q}^{(t-1)} + \boldsymbol{f})$
10: $\boldsymbol{q}^{(t)} \leftarrow \boldsymbol{q}^{(t-1)} + \lambda(\boldsymbol{p}^{(t)} - \boldsymbol{y}^{(t-1)})$
11: $\boldsymbol{y}^{(t)} \leftarrow \Pi_{\Delta^{K-1}}(\boldsymbol{q}^{(t)})$
12: **until** the convergence of $\boldsymbol{y}^{(t)}$
13: $\breve{\boldsymbol{B}}_{*i} \leftarrow \boldsymbol{y}^{(t)}$
14: **end for**
15: **for** $(i,\ k) \in \{1, ..., N\} \times \{1, ..., K\}$ **do**
16: $\boldsymbol{B}_{ik} \leftarrow (\breve{\boldsymbol{B}}_{ki}\boldsymbol{c}_i)/(\sum_{i'=1}^{N} \breve{\boldsymbol{B}}_{ki'}\boldsymbol{c}_{i'})$
17: **end for**
18: **return** $\boldsymbol{B}$

($\Pi_{\Delta^{K-1}}(\cdot)$ is the orthogonal projection to the $K-1$ simplex. See the reference for the implementation.)

Algorithm 4 Diagonal Recovery and $\boldsymbol{\alpha}$-learning

def RECOVER-A($\boldsymbol{C}, \boldsymbol{B}, S$)
1: $(\boldsymbol{C}_{SS},\ \boldsymbol{D}) \leftarrow (\boldsymbol{C}(S,S), \boldsymbol{B}(S,*))$
2: $\boldsymbol{A} \leftarrow \boldsymbol{D}^{-1}\boldsymbol{C}_{SS}\boldsymbol{D}^{-1}$
3: **return** $\boldsymbol{A}$

def RECOVER-ALPHA($\boldsymbol{A}$)
1: $\boldsymbol{a} \leftarrow \boldsymbol{A}\boldsymbol{1}$
2: $\overline{\boldsymbol{A}} \leftarrow$ the row-normalized $\boldsymbol{A}$
3: $\overline{\boldsymbol{A}}_0 \leftarrow \overline{\boldsymbol{A}} - diag(diag(\overline{\boldsymbol{A}}))$
4: $\boldsymbol{u} \leftarrow (\boldsymbol{1}^T\overline{\boldsymbol{A}}_0)/(K-1)$
5: $\boldsymbol{v} \leftarrow (diag(\overline{\boldsymbol{A}}) - \boldsymbol{u})^{\dagger} - \boldsymbol{1}^T$
6: $\alpha_0^{(0)} \leftarrow (\sum_k v_k)/K$
7: **repeat**
8: $\nabla\alpha_0 \leftarrow (1 - \alpha_0 - K) + \alpha_0 K \sum_k a_k^2 +$
9: $(\alpha_0 + 1)\sum_k \overline{\boldsymbol{A}}_{kk} - (\alpha_0 + 1)\sum_k (\overline{\boldsymbol{A}}\boldsymbol{a})_k$
10: $\alpha_0^{(t)} \leftarrow (\alpha_0^{(t-1)} - \eta\nabla_{\alpha_0})_+$
11: **until** the convergence of $\alpha_0^{(t)}$
12: **return** $\alpha_0^{(t)} \cdot \boldsymbol{a}$

(Set indexing $(\cdot,\cdot)$ extracts a principle submatrix whose rows/columns correspond to the arguments. The † operation means entry-wise scalar inverse.)

the orthogonal complement for each iteration. Random projections are suggested for the sizable vocabulary, but such projections can no longer maintain the insisted geometric structures of the rectified $\boldsymbol{C}$ and likely degrade the topic quality [18]. The proposed Algorithm 2 requires only $O(NK)$ space to store $\boldsymbol{Q}$ and performs implicit updates on $\boldsymbol{u}$ in $O(nnz(\boldsymbol{C})K)$ times without modifying the input $\overline{\boldsymbol{C}}$. It also leverages the sparsity of the matrix, allowing users to quickly find the set of anchor words without dimensionality reduction techniques such as random projection.

Step 3: Recover $\boldsymbol{B}$. If being provided with the set of the anchor words S and the convex coefficients $\{p(z=k|x=i)\}$, one can easily recover $\boldsymbol{B}$ by applying the Bayes rule. Let $\breve{\boldsymbol{B}}$ be the topic-word matrix in $\mathbb{R}^{K\times N}$ with $\breve{\boldsymbol{B}}_{ki} = p(z=k|x=i)$. Then

$$\boldsymbol{B}_{ik} = p(x=i|z=k) = \frac{p(z=k|x=i)p(x=i)}{\sum_{i'=1} p(z=k|x=i')p(x=i')} = \frac{\breve{\boldsymbol{B}}_{ki}\boldsymbol{c}_i}{\sum_{i'=1} \breve{\boldsymbol{B}}_{ki'}\boldsymbol{c}_{i'}}, \tag{3}$$

where $\boldsymbol{c}_i$ indicates the unigram probability $p(x=i)$ of the word i, which can be evaluated by $\sum_j p(x=i, x=j) = \sum_j \boldsymbol{C}_{ij}$. Hence the key of this step is to learn the topic-word matrix $\breve{\boldsymbol{B}}$ by solving multiple Simplex-constrained Non-negative Least Squares (SNLS) that satisfies $\overline{\boldsymbol{C}}_{ij} = \sum_k \breve{\boldsymbol{B}}_{ki}\overline{\boldsymbol{C}}_{s_k,j}$ for each i. While the exponentiated gradient algorithm (ExpGrad) – used in most of the previous work [9, 3] – quickly converges in practice, tuning the learning rate is overly mysterious, less ensuring the confidence of the results. Instead, we propose another algorithm that

uses Alternating Direction Method of Multipliers (ADMM). Let $\boldsymbol{U}^T$ be the wide submatrix of $\overline{\boldsymbol{C}}$ consisting only of the rows corresponding to the anchor words S. Say $\boldsymbol{v}^T$ is a row vector corresponding to any non-anchor word i. Then Algorithm 3 tries to find $\boldsymbol{y} \in \Delta^{K-1}$ that minimizes $\frac{1}{2}\|\boldsymbol{U}\boldsymbol{y} - \boldsymbol{v}\|_2^2$ by solving SNLS for each i in parallel by Douglas-Rachford (DR) splitting with the rate parameter λ. Since the γ-proximal solution close to the current $\boldsymbol{x}$ is given by $prox_\gamma(\boldsymbol{x}) = (\gamma\boldsymbol{U}^T\boldsymbol{U} + \boldsymbol{I}_K)^{-1}(\boldsymbol{x} + \gamma\boldsymbol{U}^T\boldsymbol{v})$, we can evaluate the first invariant part $\boldsymbol{F} = (\gamma\boldsymbol{U}^T\boldsymbol{U} + \boldsymbol{I}_K)^{-1}$ just once and the second invariant part $\boldsymbol{f} = \gamma\boldsymbol{U}^T\boldsymbol{v}$ only $N - K$ times for different $\boldsymbol{v}$'s.[9]

Step 4: Recover $\boldsymbol{A}$ and $\boldsymbol{\alpha}$. The final step is to recover the topic-topic matrix A and the hyperparameter $\boldsymbol{\alpha}$ if learnable (e.g., $\mathfrak{f}(\boldsymbol{\alpha}) = Dir(\boldsymbol{\alpha})$). Again leveraging the separability assumption, $p(x_1 = s_k, x_2 = s_l) = \sum_{l'}(\sum_{k'} p(x_1 = s_k|z_1 = k')p(z_1 = k', z_2 = l'))p(x_2 = s_l|z_2 = l') = p(x_1 = s_k|z_1{=}k)\sum_{l'} p(z_1{=}k, z_2{=}l')p(x_2{=}s_l|z_2{=}l') = p(x_1{=}s_k|z_1{=}k)p(z_1{=}k, z_2{=}l)p(x_2{=}s_l|z_2{=}l)$. Thus $\boldsymbol{A}_{kl} = p(x_1 = s_k|z_1 = k)^{-1}\boldsymbol{C}_{s_k,s_l}\ p(x_2 = s_l|z_2 = l)^{-1}$. Algorithm 4 concisely performs this derivation in terms of two matrix multiplications at line 2. Therefore in JSMF, the co-occurrence of their anchor words s_k and s_l transparently captures the correlation between a pair of topics k and l. Note that the anchor words are generally rare words (in order to be the vertices of underlying convex hull of the word co-occurrence space) whose co-occurrences are even rarer and noisier. The power of the rectification is thus in correcting and balancing these statistics based on the geometric structures of the posterior moments [3], thereby realizing **correlated topic modeling**.

Suppose that the recovered $\boldsymbol{A}$ is close to the second moments of $\mathrm{Dir}(\boldsymbol{\alpha})$, $\mathbb{E}_{\boldsymbol{w}\sim\mathrm{Dir}(\boldsymbol{\alpha})}[\boldsymbol{w}\boldsymbol{w}^T]$. Then its row sum vector $\boldsymbol{a}$ becomes $\boldsymbol{\alpha}/\alpha_0$ (i.e., first moments of $\mathrm{Dir}(\boldsymbol{\alpha})$), meaning we can easily recover the $\boldsymbol{\alpha}$ up to the scalar. Let $\overline{\boldsymbol{A}}$ be the row-normalized $\boldsymbol{A}$. In theory the diagonal entries of $\overline{\boldsymbol{A}}$ should always be bigger than the off-diagonal entries in the same column by $1/(\alpha_0 + 1)$. As the real data never satisfies the model, we evaluate the average $\boldsymbol{u}$ of the off-diagonal entries and compute the K candidates for $1/(\alpha_0+1)$. Then the vector $\boldsymbol{v}$ stores the K corresponding candidates for α_0, and we start fitting the learned $\overline{\boldsymbol{A}}$ to the row-normalized version of the second moments $\mathbb{E}_{\boldsymbol{w}\sim\mathrm{Dir}(\boldsymbol{\alpha})}[\boldsymbol{w}\boldsymbol{w}^T]$ by finding $\alpha_0 > 0$ that minimizes the Frobenius norm of their difference: $\sum_{k=1}^K(\frac{\alpha_0 a_k+1}{\alpha_0+1} - \overline{\boldsymbol{A}}_{kk})^2 + \sum_{k\neq l}(\frac{\alpha_0 a_j}{\alpha_0+1}) - \overline{\boldsymbol{A}}_{kl})^2$. We verify that the optimal α_0 is quickly attained inside the candidate interval, and agrees well with the result of the exhaustive line-search within the offset of 10^{-3}. While our algorithm outperforms the previous α-recovery method proposed by [8], we do not compare the learned $\boldsymbol{\alpha}$ with other inference-based algorithms like the fixed-point iteration [24].[10]

[9]Note first that this inversion is performed only for small $K \times K$ matrix rather than $N \times N$. Note second that Algorithm 3 (at line 6) projects the least square solution by the normal equation to the simplex Δ^{K-1} in order to speculate a reasonable initialization. Whereas this procedure aggravates the performance of the ExpGrad due to its multiplicative nature, it benefits the ADMM-DR to achieve sparser solutions without putting another prior $\mathfrak{g}(\boldsymbol{\beta})$ [20].

[10]Whereas the JSMF can capture arbitrary topic correlations, fitting to Dirichlet can only model weakly negative correlations. Indeed we are solving a highly over-determined system to find α_0, loosing rich correlation information.

4 Hierarchical Topic Modeling

Topics help users organize documents, but as the number of topics grows, it begins to be important to organize the topics themselves. One option is to arrange topics in hierarchies [25, 6, 7]. As the correlations $\boldsymbol{A}$ learned by the RAW algorithm have the same positive semidefinite and joint-stochastic structures as the original matrix $\boldsymbol{C}$, one might want to further factorize $\boldsymbol{A}$ in order to learn a smaller number of "supertopics." This approach should only be effective if there are non-trivial off-diagonal entries in $\boldsymbol{A}$, since otherwise the matrix would have no more interesting low-dimensional structure, and indeed this is true of non-rectified algorithms [8, 9]. Rectification can effectively balance the diagonal entries (if they meaningfully exist), thus transforming the topic co-occurrence into a further decomposable low-rank matrix. Therefore we can recursively apply the RAW algorithm on the recovered $\boldsymbol{A}$, learning supertopics of the current topics.

Suppose that the initial run of the RAW algorithm factorizes $\boldsymbol{C}{=}\boldsymbol{C}_1{\approx}\boldsymbol{B}_1\boldsymbol{A}_1\boldsymbol{B}_1^T$ with K_1 subtopics. Define $\boldsymbol{C}_{t+1}$ as the rectified $\boldsymbol{A}_t$, and then the next run factorizes $\boldsymbol{C}_{t+1}{\approx}\boldsymbol{B}_{t+1}\boldsymbol{A}_{t+1}\boldsymbol{B}_{t+1}^T$, resulting K_{t+1} supertopics ($K_{t+1} < K_t$). The recursive applications allow users to achieve a level-wise DAG of hierarchical topics where the lowest level ($t = 0$) corresponds to the observed words, the next level ($t = 1$) indicates the subtopics, and the upper level describes their ($t = 2$) supertopics, and so on. The learned $\boldsymbol{A}_t$ explains topic correlations within each level t, whereas the learned $\boldsymbol{B}_t$ analyzes the weights between two consecutive levels $t-1$ and t. Most interestingly, we may attain better K_{t+1} anchors with the cleaner topics at the upper levels comparing to the direct application with K_{t+1} topics because of the continuous noise balancing via the intermediate rectifications.

5 Experimental Results

We evaluate our algorithms on two standard textual datasets: NIPS full papers (NIPS) and New York Times news articles (NYTimes). We also adopt two other preference-based datasets: Movielens 10m reviews (Movies)[11] and Yes.com complete playlists (Songs)[12]. While the two textual datasets do not consist of any topic-specific meta data, but we can retrieve genre information for Movies and Songs.[13] Songs dataset is particularly useful because songs in each playlist are likely chosen coherently based on genre-specific themes, whereas people often watch and review newly released movies rather than consuming only similar genres. We process training documents following [3] for fair comparison.[14] Basic statistics of each dataset are available in Figure 1.

[11] `https://grouplens.org/datasets/movielens/10m/`

[12] `http://csinpi.github.io/lme/data_page.html`

[13] We scrape the genre of each song by matching the artist and the title from `http://www.discogs.com`.

[14] Remove 347 English stop words. Prune rare words based on tf-idf scores. Discard short documents with fewer than 5 tokens after vocabulary curation.

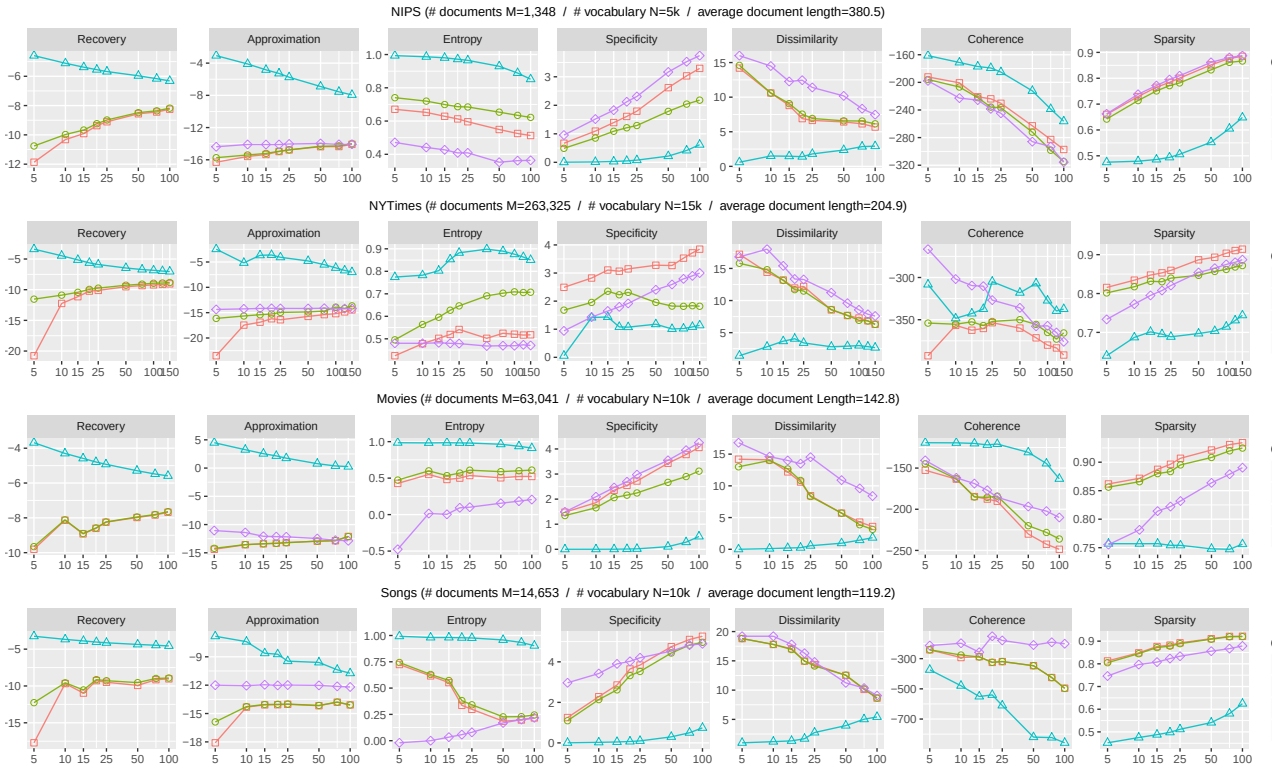

Figure 1: Experiment with the various numbers of topics, K (x-axis). △ Baseline: the popular algorithm without rectification [9], ○ ExpGrad: the previous work with AP-rectification [3], □ ADMM-DR: this paper, ◇ Gibbs: Gibbs sampling. The results show 1) the power of the rectification and 2) the improvement by our ADMM-DR on overall metrics. It is more comparable to probabilistic Gibbs Sampling than ExpGrad.

5.1 Quantitative analysis

After constructing $\boldsymbol{C}$ (Step 0), the Baseline method [9] jumps to the anchor-finding (Step 2) without any rectification so that we can demonstrate the power of managing model-data mismatch. However we do not use any random projection or pseudo-inverse recovery of $\boldsymbol{A}$ given in [9] in order to prevent further degradation of learning quality. For methods within the framework of the JSMF, we execute 150 iterations of Alternating Projection (AP) for the rectification (Step 1). Since our new anchor-finding algorithm does not change any result, but only improves time/space complexity, solving SNLS (Step 3) contrasts our work from the previous work [3]. For the exponentiated gradient (ExpGrad), we set the learning rate as 50.0, which is the best-known from [3]. For our ADMM with DR splitting (ADMM-DR), we set $\lambda = 1.9$, the widely known best, and $\gamma = 3.0$ as the algorithm is not sensitive within $\gamma \in [1.0, 5.0]$. For likelihood-based inference, we use Gibbs Sampling (MCMC) with 1,000 iterations as it is more stable than Variational Bayes (VB).

We measure various metrics based on [3]. However we transform Recovery and Approximation errors to logarithms of $\frac{1}{N}\sum_i \|\overline{\boldsymbol{C}}_i - \sum_k \breve{\boldsymbol{B}}_{ki}\overline{\boldsymbol{C}}_{s_k}\|_2$ and $\|\boldsymbol{C} - \boldsymbol{B}\boldsymbol{A}\boldsymbol{B}^T\|_F$ to compare ADMM-DR against ExpGrad.[15] We also add two new metrics: Entropy $(\frac{1}{N}\sum_i \frac{H(z|x=i)}{\log_2 K})$ [18] and Sparsity

[15] We measure these errors with the rectified $\boldsymbol{C}$ than the original as the inference continues with the rectified version.

$(\frac{1}{K}\sum_k \frac{\sqrt{N}-(\|\boldsymbol{b}_k\|_1/\|\boldsymbol{b}_k\|_2)}{\sqrt{N}-1})$ [26]. Smaller entropy is better because topic distribution given a word is better concentrated in a few number of topics than being uniformly spread. Values of sparsity closer to 1.0 are better because our model does not have a sparsity-insisting prior $\mathfrak{g}(\boldsymbol{\beta})$ on each topic (i.e., each column of $\boldsymbol{B}$). Specificity $(\frac{1}{K}\sum_k KL(p(x|z=k)\|p(x)))$ 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 $(\frac{1}{K}\sum_k \sum_{x_1\neq x_2}^{x_1,x_2\in Top_{20}} \log \frac{D_2(x_1,x_2)+\epsilon}{D_1(x_2)})$ penalizes any pair of top words in each topic that do not appear together in the training documents.[16] Larger values are better for all three metrics.

Figure 1 shows that the Baseline method works notably worse than other methods and is far behind the trend of Gibbs sampling, reconfirming [3]. AP+ExpGrad and AP+ADMM-DR generally agree on many metrics, but ADMM-DR produces more specific and sparse topics with smaller entropies. ADMM-DR also improves inference quality by decreasing Recovery and Approximation errors especially when K is small. For running time, Sparse Implicit Column-pivoted QR takes in average 0.71 shorter times than the explicit anchor finding given in [9]. For topic recovery, ADMM-DR takes 1.92 more times than ExpGrad if using the same maximum number of 500 iterations.

5.2 Qualitative analysis for hierarchy and correlations

For hierarchy analysis, in the NIPS dataset, we compare across three different settings: 1) single JSMF with $K = 5$ (JSMF-5); 2) recursive JSMF with $K = 25$ then $K = 5$ (JSMF-25:5); 3) single LDA with $K = 5$ (Gibbs-5), manually sorted to align with JSMF-5. Table 1 shows the most prominent 7 words out of top 20 words for each topic similar to [3]. As expected, JSMF-5 and Gibbs-5 are fairly comparable. Whereas the five supertopics from JSMF-25:5 show different partitions: T3 is about machine learning theory and T4 is about probabilistic models, JSMF-5 and Gibbs-5 mix these themes in their respective T4s.

Table 1: Top 7 words for each of five topics by three models.

Recursive JSMF with $K = 25$ then $K = 5$ (JSMF-25:5)
T0: neuron dynamic signal gradient matrix control solution
T1: action policy optimal reinforcement control states reward
T2: object hidden layer image representation recognition cell
T3: bound threshold theorem class dimension polynomial proof
T4: gaussian density likelihood noise mixture component prior
Single JSMF with $K = 5$ (JSMF-5)
T0: neuron circuit cell synaptic signal layer activity
T1: control action dynamic optimal policy controller reinforcement
T2: recognition layer hidden word speech image net
T3: cell field visual direction image motion orientation
T4: gaussian noise hidden approximation matrix bound examples
Single Probabilistic LDA (Gibbs-5)
T0: neuron cell visual signal response field activity
T1: control action policy optimal reinforcement dynamic robot
T2: recognition image object feature word speech features
T3: hidden net layer dynamic neuron recurrent noise
T4: gaussian approximation matrix bound component variables

[16] Topic coherence could be deceptive if a model learns many duplicated topics containing the frequent words [13].

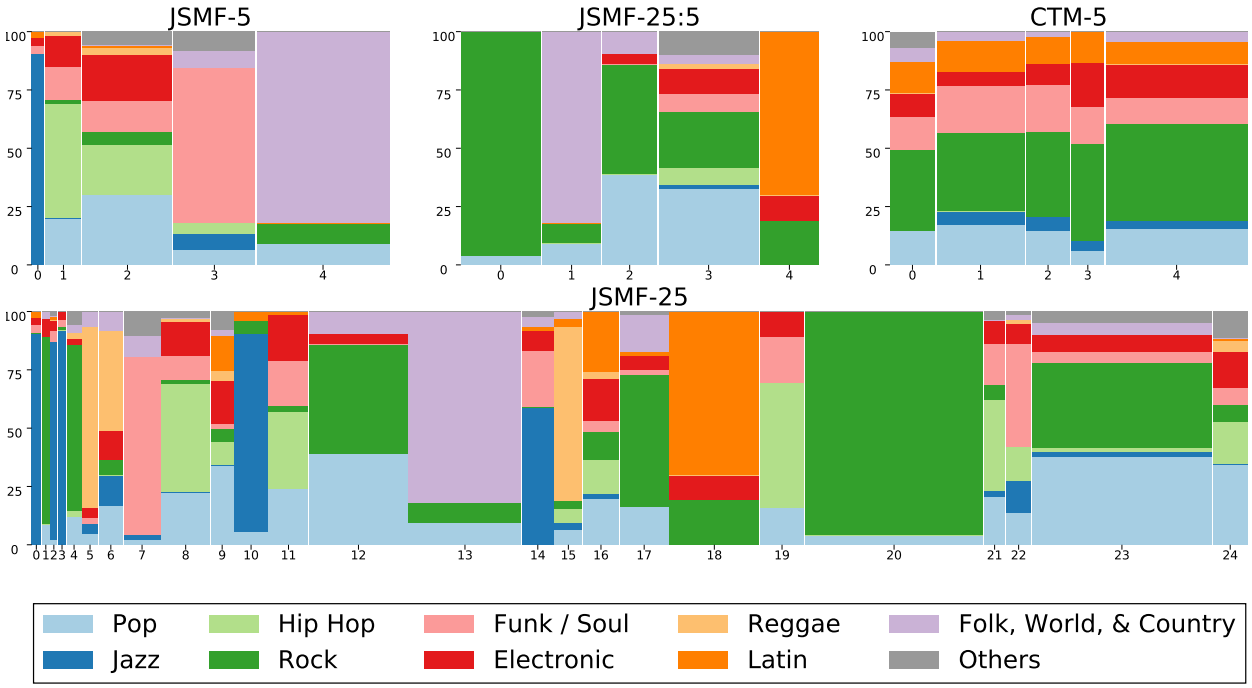

Figure 2: Second row. 25 subtopics on Songs dataset. Given 20 top songs of each topic, the stacked bar chart indicates the percentages of the most popular 9 genres. The width of each topic is proportional to the marginal likelihood of the topic $p(z = k) = \sum_l A_{kl}$. **First row.** The leftmost and the rightmost panels show 5 topics from independent running of the JSMF and the CTM, respectively. The middle panel represents 5 supertopics by recursive running of the JSMF on top of 25 subtopics given in the second row.

Evaluating correlated topic models is not easy due to the potential subjectivity in analysis. If models are capable of considering and learning topic correlations, the genres of top "words" (i.e., songs) in each topic are more likely to align with human classifications. The standard probabilistic topic model is CTM [5], which uses Logistic-Normal priors with pairwise covariance between topics. When we run the variational CTM-5 [5] with the default parameters, the resulting topics do not have distinguishable genre associations as illustrated in Figure 2. This failure may be the result of spurious correlations as pointed out in [27]. However, the simple JSMF-5 captures Jazz (T0), Funk (T3), and Folk (T4) genres as independent topics with two other relatively mixed topics. Indeed JSMF-25 shows rather isolated topics of Jazz (T0, T2, T3, T10), Funk (T7), Raggae (T5, T6, T15), Latin (T18), and Rock (T20), whereas Pop is often mixed with every other genre. The five topics from the recursive run (JSMF-25:5) differ from JSMF-5: they discover Rock (T0) and Latin (T4) instead of Jazz and Funk. This could be because Jazz and Funk may be more distinctive than Rock and Latin, but they are marginally much less probable as shown in JSMF-25.

6 Conclusions

Spectral topic modeling provides an useful way to find compact high-level structures in sparse and discrete data such as text and user-preference. While the inference whose complexity does not depend on the size of data provides great scalability for processing massive data, it has been

less popular due to the lack of the ability to handle model-data mismatch on real data. Revisiting the Rectified Anchor Words (RAW) algorithm within the framework of the Joint Stochastic Matrix Factorization (JSMF), we present scalable implementation of each inference step and propose novel algorithms for anchor-fining and ADMM-based topic recovery. The quantitative analysis proves that our ADMM-based algorithm outperforms the previous exponentiated gradients [9, 3] in various types of data, being more comparable to probabilistic Gibbs. We also show how the JSMF considers the co-occurrence of the anchor words toward learning quality topics and their correlations. The qualitative analysis demonstrates that our model is capable of learning high-quality topic correlations aligned better with human annotations than the probabilistic model that is specifically designed to capture those correlations. Achieving quality topic correlations further supports hierarchical topic modeling that finds different and cleaner supertopics. Its simplicity can greatly benefit practitioners and industry-level data processing. In future, improving the complexity of co-occurrence construction and rectification will make the approach even more scalable.

References

[1] T. L. Griffiths and M. Steyvers. Finding scientific topics. *National Academy of Sciences*, 2004.

[2] Chong Wang, David M. Blei, and Fei-Fei Li. Simultaneous image classification and annotation. In *CVPR*, 2009.

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

[4] Yan Liu, Alexandru Niculescu-Mizil, and Wojciech Gryc. Topic-link LDA: Joint models of topic and author community. In *ICML*, 2009.

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

[6] W. Li and A. McCallum. Pachinko allocation: DAG-structured mixture models of topic correlations. In *ICML*, pages 633–640, 2007.

[7] David Mimno, Wei Li, and Andrew McCallum. Mixtures of hierarchical topics with pachinko allocation. ICML, 2007.

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

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

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

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

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

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

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

[15] Arthur Asuncion, Max Welling, Padhraic Smyth, and Yee Whye Teh. On smoothing and inference for topic models. In *UAI*, 2009.

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

[17] Trapit Bansal, Chiranjib Bhattacharyya, and Ravindran Kannan. A provable svd-based algorithm for learning topics in dominant admixture corpus. In *NIPS*. 2014.

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

[19] Zita Marinho. Moment-based algorithms for structured prediction. 2015.

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

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

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

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

[24] Thomas Minka. Estimating a dirichlet distribution, 2000.

[25] David M. Blei, Thomas Griffiths, Michael Jordan, and Joshua Tenenbaum. Hierarchical topic models and the nested chinese restaurant process. In *NIPS*, 2003.

[26] Patrik O. Hoyer. Non-negative matrix factorization with sparseness constraints. *JMLR*, 2004.

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