JOINT-STOCHASTIC SPECTRAL INFERENCE FOR ROBUST CO-OCCURRENCE MODELING AND LATENT TOPIC ANALYSIS

A Dissertation

Presented to the Faculty of the Graduate School

of Cornell University

in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

by

Moontae Lee

May 2018

© 2018 Moontae Lee

ALL RIGHTS RESERVED

JOINT-STOCHASTIC SPECTRAL INFERENCE FOR ROBUST
CO-OCCURRENCE MODELING AND LATENT TOPIC ANALYSIS

Moontae Lee, Ph.D.
Cornell University 2018

Co-occurrence information is powerful statistics that can model various discrete objects by their joint instances with other objects. Transforming unsupervised problems of learning low-dimensional geometry into provable decompositions of co-occurrence information, spectral inference provides fast algorithms and optimality guarantees for non-linear dimensionality reduction or latent topic analysis. Spectral approaches reduce the dependence on the original training examples and produce substantial gain in efficiency, but at costs:

- The algorithms perform poorly on real data that does not necessarily follow underlying models.
- Users can no longer infer information about individual examples, which is often important for real-world applications.
- Model complexity rapidly grows as the number of objects increases, requiring a careful curation of the vocabulary.

The first issue is called *model-data mismatch*, which is a fundamental problem common in every spectral inference method for latent variable models. As real data never follows any particular computational model, this issue must be addressed for practicality of the spectral inference beyond synthetic settings. For the second issue, users could revisit probabilistic inference to infer information about individual examples, but this brings back all the drawbacks of traditional approaches. One method is recently developed for spectral inference, but

it works only on tiny models, quickly losing its performance for the datasets whose underlying structures exhibit realistic correlations. While probabilistic inference also suffers from the third issue, the problem is more serious for spectral inferences because co-occurrence information easily exceeds storable capacity as the size of vocabulary becomes larger.

We cast the learning problem in the framework of Joint Stochastic Matrix Factorization (JSMF), showing that existing methods violate the theoretical conditions necessary for a good solution to exist. Proposing novel rectification paradigms for handling the model-data mismatch, the Rectified Anchor Word Algorithm (RAWA) is able to learn quality latent structures and their interactions even on small noisy data. We also propose the Prior Aware Dual Decomposition (PADD) that is capable of considering the learned interactions as well as the learned latent structures to robustly infer example-specific information. Beyond the theoretical guarantees, our experimental results show that RAWA recovers quality low-dimensional geometry on various textual/non-textual datasets comparable to probabilistic Gibbs sampling, and PADD substantially outperforms the recently developed method for learning low-dimensional representations of individual examples.

Although this thesis does not address the complexity issue for large vocabulary, we have developed new methods that can drastically compress co-occurrence information and learn only with the compressed statistics without losing much precision. Providing rich capability to operate on millions of objects and billions of examples, we complete all the necessary tools to make spectral inference robust and scalable competitor to probabilistic inference for unsupervised latent structure learning. We hope our research serves an initial basis for a new perspective that combines the benefits of both spectral and probabilistic worlds.

BIOGRAPHICAL SKETCH

I was born in Seoul, the capital of South Korea, in a family with various scholars for multiple generations. My grandfather used to own his library at home, a wonderful labyrinth with full of books stacked to the ceiling. To deliver a small message from my grandmother, I needed to carefully walk through the maze. He was always seating at the center with his squared manuscripts and fountain pencils, which seemed as old as him. After he had passed away, I later realized this image was the point of my departure and will become the final destination.

The best of luck was that my parents decided to move to a small town outside Seoul, when I was still a boy attending an elementary school. Playing in the mountain forest and sleeping among the shining trees, I learned more than anything from the full glory of green, falling in love with every piece of nature such as the moan of wind, the noise by chance, and the sound of silence. Like Huckleberry Finn, I want to better understand such a wonderful world, perhaps putting myself into the adventures: majoring in Computer Science, Mathematics, and Psychology for the bachelor's degree at Sogang University, and specializing in Artificial Intelligence for the master's degree at Stanford University.

I am now a Ph.D. candidate in the Department of Computer Science at Cornell University. Being advised by two magicians, Prof. David Mimno and David Bindel, my research is to study the modern art of Machine Learning. Every ingredient in natural languages inspires me to develop various computational models for their representations and understanding. I become later interested in modeling biases in human decision making, establishing an ambitious goal to combine the two disparate arts of modeling: *modeling the world* and *modeling the biases*. I wish Machine Learning not only tries to leverage larger amount of data, but also tries to better understand proper context between the data.

Dedicated to my grandmother, Sungrae Lee (1921-2017),

eternal in my heart.

ACKNOWLEDGEMENTS

Co-occurrence statistics that I have studied for my Ph.D. is mostly of the matrix form called *doubly nonnegative*. My advisor, David Mimno, has magical power that can ever find something bright in every negative. My co-advisor, David Bindel, always opens his door, inverting my negative thoughts into something constructive. Without their doubly nonnegative supports: unconditional advice and unlimited encouragement, I would unlikely finish half-a-decade-long journey to be a doctor. I am also deeply thankful to my minor advisor, Peter Frazier, who has never failed in giving me the best guidance to find the missing pieces.

When I was an orphan (working without any advisor), Lillian Lee and Jon Kleinberg gave me an exceptional care, leading me to a new world of Computational Social Science. I should be grateful to our CS/OR family as well – Claire Cardie and Thorsten Joachims for their faithful advice on teaching and research, Adrian Lewis for inspiring coffee talks, and Rebecca Stewart for her continuous coordination of the Ph.D. program. It was my pleasant honor to collaborate with great researchers in Microsoft Research – Paul Smolensky, Xiaodong He, Jianfeng Gao, and Wen-tau Yih. For master's study at Stanford University, I luckily owed most of my knowledge to excellent scholars – Daphne Koller, Serge Plotkin, and Scott Klemmer. I am also thankful to my undergraduate advisor, Sungyong Park, who has still supported me more than a decade.

Friendship is the greatest gift at Cornell. My brilliant lab mates include Alexandra Schofield, Jack Hassel, Laure Thompson and Maria Antoniak. My first year companions – Paul Upchurch, Ashesh Jain, Efe Gencer, Ozan Irsoy, and Jaeyong Sung. Friends in the same field of Machine Learning and Natural Language Processing – Ruben Sipos, Anshumali Shrivastava, Karthik Raman, Yue Gao, Lu Wang, Igor Labutov, Jason Yosinsky, Chenhao Tan, Arzoo Katiyar,

Vlad Niculae, Justine Zhang, and Ashudeep Singh. My office mates and other friends – Bishan Yang, Eleanor Birrell, Theodoros Gkountouvas, Jonathan Park, Jiyong Shin. Friends in Information Science – Dongwook Yoon, Laewoo Kang, Jaeyeon Kihm, Minsu Park and Sharifa Sultana. Finally my two closest friends who used to be my mentors as well – Adith Swaminathan and Elisavet Kozyri.

During my doctoral journey, I was able to learn profound Indian classical music with Aravind Sivakumar, Vikram Rao, Aparna Mahadevan, Karthik Sridharan, Sumita Mitra, Tirth Pandya, Maithra Raghu and Sarah Tan. Another fortune was to play the tennis regularly with Lawrence Jin, Daniel Lee, Jongrim Ha, Samuel Lee, Hanjong Paik, Younghwa Seok, Jaesun Lee, Yuna Won, Sujin Lee, Sungjoon Park, Jihyun Han, and Yumi Woo. More than thousand words, thanks deeply to my pseudo family – Bori Seo, Dayoung Kim, and Shiyi Li. Thanks gratefully to my dear friends – Dabeum Shin, Handol Jang, Boosung Kim, Dean Park, Hieyoon Kim, Yookyung Noh, Eunyoung Lim, Mingi Jeon, Sungbae Oh, Soojin Lee, Dukshin Lee, Myeongyeol Kim, Ran Kim, Jiyeob Kim, Penny Hu, Rui Lu, Michael Chun, Jeongjin Ku, Zack Lipton, and Patrick Grafe. Also to my wonderful mentees – Seokhyun Jin, Saerom Choi, and Sungjun Cho.

Thanks to my father, Jonggul Lee, for his endless support and my mother, Daebong Jung, for her infinite love. I owe everything to my brother, Haktae Lee, who has taken care me from the beginning to the end of the journey. Thanks to my sister in law, Soyoung Kim, as well. Deeply grateful to my grandparents – Hanki Lee and Haekyung Kim for inheriting me their scholarly legacy and Manki Jung and Sungrae Lee for securing my life and love. Thanks for warm advice from my cousins – Yongtae, Bongtae, and Intae Lee's. Thanks so much to my new family – Kiyeob Hwang, Myungjoo Lee, Chaeyoung Hwang, and Seungjae Hwang. Lastly my friend, my dearest, and my beloved – Sinae Hwang.

My work was supported through NSF 1652536 and the research grant from the Cornell Graduate School. My sincere apologies to other friends, colleagues, and mentors that I am not able to name above. I deeply thank all of you for being with me in this journey.

TABLE OF CONTENTS

Biographical Sketch . . . iii
Dedication . . . iv
Acknowledgements . . . v
Table of Contents . . . viii
List of Tables . . . x
List of Figures . . . xi

1 Motivational Study and Visualization 1
1.1 Introduction . . . 1
1.2 Related Work . . . 4
1.3 Low-dimensional Embeddings . . . 6
1.4 Experimental Results . . . 8
1.4.1 Anchor-word Selection . . . 11
1.4.2 Quantitative Results . . . 12
1.4.3 Qualitative Results . . . 19
1.5 Conclusion . . . 20

2 Rectification for Robust Low-rank Spectral Topic Inference 24
2.1 Introduction . . . 24
2.2 Requirements for Factorization . . . 27
2.3 Rectified Anchor Words Algorithm . . . 30
2.4 Experimental Results . . . 34
2.5 Analysis of Algorithm . . . 39
2.6 Related and Future Work . . . 42

3 Applications and Comparison with Other Related Models 43
3.1 Introduction . . . 43
3.2 Spectral Topic Inference . . . 46
3.2.1 Joint Stochastic Matrix Factorization . . . 48
3.2.2 Tensor Decomposition . . . 50
3.3 The Robust Rectified Anchor Word Algorithm . . . 52
3.4 Hierarchical Topic Modeling . . . 58
3.5 Experimental Results . . . 59
3.5.1 Quantitative Analysis . . . 60
3.5.2 Qualitative Analysis . . . 63
3.5.3 Hierarchy and Further Analysis . . . 64
3.6 Discussion . . . 66

4 Prior-aware Document-specific Topic Inference 68
4.1 Introduction . . . 68
4.2 Foundations and Related Work . . . 71
4.3 Document-specific Topic Inference . . . 76

4.3.1 Simple Probabilistic Inverse (SPI) . . . 76
4.3.2 Prior-aware Dual Decomposition (PADD) . . . 78
4.3.3 Parallel formulation with ADMM . . . 79
4.4 Experimental Results . . . 82
4.5 Discussion . . . 88

A Appendix for Chapter 2 **90**
A.1 Introduction . . . 90
A.2 Requirements for Factorization . . . 90
A.3 Rectified Anchor Word Algorithm . . . 91
A.4 Experimental Results . . . 96
A.5 Analysis of Algorithm . . . 99
A.6 Related and Future Work . . . 101

Bibliography **105**

LIST OF TABLES

1.1 Statistics for datasets used in experiments 9
1.2 The first 12 anchor words selected by three algorithms for the NYT corpus. 11
1.3 Example *t*-SNE topics and their most similar topics across algorithms. The Greedy algorithm can find similar topics, but the anchor words are much less salient. 23

2.1 Statistics of four datasets. 35
2.2 Each line is a topic from NIPS ($K = 5$). Previous work simply repeats the most frequent words in the corpus five times. 36

3.1 Top 7 words for each of five topics by three models. 65

4.1 Real experiment on Fully-Real (*FR*) corpora. For each entry, a pair of values indicates the corresponding metrics on training/unseen documents. Averaged across all models with different K's. Rand estimates randomly. For two new metrics: Prior-dist and Non-supp, smaller numbers are better. PADD performs the best considering topic compositions learned by Gibbs Sampling as the ground-truth. 86

LIST OF FIGURES

1.1 2D *t*-SNE projection of a Yelp review corpus and its convex hull. The words corresponding to vertices are anchor words for topics, whereas non-anchor words correspond to the interior points. . . 2
1.2 2D PCA projections of a Yelp review corpus and its convex hulls. 7
1.3 3D *t*-SNE projection of a Yelp review corpus and its convex hull. Vertices on the convex hull correspond to anchor words. 8
1.4 Recovery error is similar across algorithms. 13
1.5 Words have higher topic entropy in the greedy model, especially in NYT, resulting in less specific topics. 14
1.6 Greedy topics look more like the corpus distribution and more like each other. 15
1.7 The greedy algorithm creates more coherent topics (higher is better), but at the cost of many overly general or repetitive topics. . 16
1.8 Anchor words have higher probability, and therefore greater salience, in *t*-SNE and PCA models (1 ≈ one third the probability of the top ranked word). 17
1.9 *t*-SNE topics have better held-out probability than greedy topics. 18

2.1 2D visualizations show the low-quality convex hull found by Anchor Words [9] (left) and a better convex hull (middle) found by discovering anchor words on a rectified space (right). 25
2.2 JSMF applications, with anchor-word equivalents. 26
2.3 The JSMF event space differs from LDA's. JSMF deals only with pairwise co-occurrence events and does not generate observations/documents. 27
2.4 The algorithm of [9] (first panel) produces negative cluster co-occurrence probabilities. A probabilistic reconstruction alone (this paper & [8], second panel) removes negative entries but has no off-diagonals and does not sum to one. Trying after rectification (this paper, third panel) produces a valid joint stochastic matrix. 33
2.5 Experimental results on real dataset. The x-axis indicates $logK$ where K varies by 5 up to 25 topics and by 25 up to 100 or 150 topics. Whereas the Baseline algorithm largely fails with small K and does not infer quality B and A even with large K, Alternating Projection (AP) not only finds better basis vectors (Recovery), but also shows stable and comparable behaviors to probabilistic inference (Gibbs) in every metric. 38

3.1 Matrix vs Tensor. Tensor algorithm performs better than Baseline Anchor Word algorithm [9], but much poorer than the Rectified Anchor Word algorithm: ExpGrad [51] and Gibbs. Surprisingly, tensor algorithm does not show consistent behavior for increasing numbers of topics in X-axis. Close to Gibbs is generally better in Y-axis. 61

3.2 ADMM-DR vs ExpGrad. Our ⊖ ADMM-DR algorithm outperforms the previous state-of-the-art rectified algorithm ⧋ ExpGrad, being more comparable to probabilistic ⟐ Gibbs sampling. ⊟ Baseline algorithm without rectification works consistently poor due to model-data mismatch. Panel 1, 2, 3: lower is better / 4, 5, 7: higher is better / 6: closer to Gibbs is better. 62
3.3 ADMM-DR (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. The number below each topic indicates the marginal probability $p_z(k)$. ADMM-DR topics capture more about genres. 63
3.4 **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 with the ADMM-DR and the CTM, respectively. The middle panel represents 5 supertopics by recursive running of the same JSMF on top of 25 subtopics given in the second row. 66

4.1 LDA asserts a topic composition $\boldsymbol{w}_m$ for each document m. Dir($\boldsymbol{\alpha}$) provides prior information for the entire corpus. 70
4.2 JSMF asserts a *joint* distribution $\boldsymbol{A}_m$ over topic pairs for each document m. $\boldsymbol{A}$ serves as a prior for the entire corpus. 71
4.3 Artificial experiment on Semi-Synthetic (*SS*) corpus with highly sparse topics and little correlation. X-axis: # topics K. Y-axis: higher numbers are better for the left three columns, lower numbers are better for the right four. SPI performs the best with $K \geq 25$. 83
4.4 Realistic experiment on Semi-Real (*SR*) corpus with non-trivial topic correlations. X-axis: # topics K. Y-axis: higher numbers are better for the left three columns, lower numbers are better for the right four. PADD is consistent and comparable to Gibbs Sampling. 84

A.1 Full experimental results on real dataset. The x-axis indicates $logK$ where K varies by 5 up to 25 topics and by 25 up to 100 or 150 topics. 97
A.2 Full experimental results on synthetic dataset. The x-axis indicates $logK$ where K varies by 5 up to 25 topics and by 25 up to 100 or 150 topics. 98
A.3 Gaps between the learned and the truth parameters. 99
A.4 Locally linear convergence of AP. 100

CHAPTER 1

MOTIVATIONAL STUDY AND VISUALIZATION

The anchor words algorithm performs provably efficient topic model inference by finding an approximate convex hull in a high-dimensional word co-occurrence space. However, the existing greedy algorithm often selects poor anchor words, reducing topic quality and interpretability. Rather than finding an approximate convex hull in a high-dimensional space, we propose to find an exact convex hull in a visualizable 2- or 3-dimensional space. Such low-dimensional embeddings both improve topics and clearly show users why the algorithm selects certain words.

1.1 Introduction

Statistical topic modeling is useful in exploratory data analysis [15], but model inference is known to be NP-hard even for the simplest models with only two topics [72], and training often remains a black box to users. Likelihood-based training requires expensive approximate inference such as variational methods [15], which are deterministic but sensitive to initialization, or Markov chain Monte Carlo (MCMC) methods [34], which have no finite convergence guarantees. Recently Arora et al. proposed the Anchor Words algorithm [9], which casts topic inference as statistical recovery using a *separability assumption*: each topic has a specific anchor word that appears only in the context of that single topic. Each anchor word can be used as a unique pivot to disambiguate the corresponding topic distribution. We then reconstruct the word co-occurrence pattern of each non-anchor words as a convex combination of the co-occurrence

patterns of the anchor words.

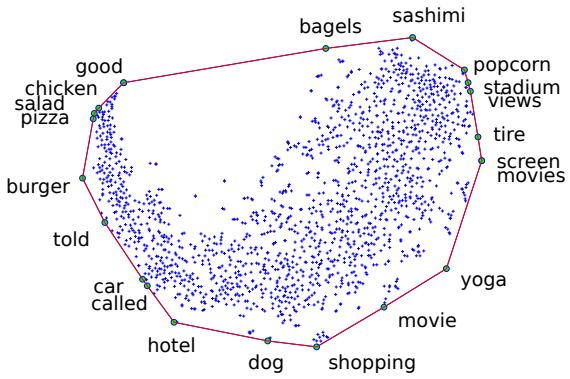

Figure 1.1: 2D *t*-SNE projection of a Yelp review corpus and its convex hull. The words corresponding to vertices are anchor words for topics, whereas non-anchor words correspond to the interior points.

This algorithm is fast, requiring only one pass through the training documents, and provides provable guarantees, but results depend entirely on selecting good anchor words. [9] propose a greedy method that finds an approximate convex hull around a set of vectors corresponding to the word co-occurrence patterns for each vocabulary word. Although this method is an improvement over previous work that used impractical linear programming methods [8], serious problems remain. The method greedily chooses the farthest point from the current subspace until the given number of anchors have been found. Particularly at the early stages of the algorithm, the words associated with the farthest points are likely to be infrequent and idiosyncratic, and thus form poor bases for human interpretation and topic recovery. This poor choice of anchors noticeably affects topic quality: the anchor words algorithm tends to produce large numbers of nearly identical topics.

Besides providing a separability criterion, anchor words also have the poten-

tial to improve topic interpretability. After learning topics for given text collections, users often request a label that summarizes each topic. Manually labeling topics is arduous, and labels often do not carry over between random initializations and models with differing numbers of topics. Moreover, it is hard to control the subjectivity in labelings between annotators, which is open to interpretive errors. There has been considerable interest in automating the labeling process [60, 49, 24]. [24] propose a measure of *saliency*: a good summary term should be both distinctive specifically to one topic and probable in that topic. Anchor words are by definition optimally distinct, and therefore may seem to be good candidates for topic labels, but greedily selecting extreme words often results in anchor words that have low probability.

In this work we explore the opposite of Arora et al.'s method: rather than finding an approximate convex hull for an exact set of vectors, we find an exact convex hull for an approximate set of vectors. We project the $N \times N$ word co-occurrence matrix to visualizable 2- and 3-dimensional spaces using methods such as t-SNE [79], resulting in an input matrix up to 3600 times narrower than the original input for our training corpora. Despite this radically low-dimensional projection, the method not only finds topics that are as good or better than the greedy anchor method, it also finds highly salient, interpretable anchor words and provides users with a clear visual explanation for why the algorithm chooses particular words, all while maintaining the original algorithm's computational benefits.

1.2 Related Work

Latent Dirichlet allocation (LDA) [15] models D documents with a vocabulary N using a predefined number of topics by K. LDA views both $\{\boldsymbol{B}_k\}_{k=1}^K$, a set of K topic-word distributions for each topic k, and $\{\boldsymbol{W}_m\}_{m=1}^M$, a set of M document-topic distributions for each document m, and $\{\mathbf{z}_m\}_{m=1}^M$, a set of topic-assignment vectors for word tokens in the document m, as randomly generated from known stochastic processes. Merging $\{\boldsymbol{B}_k\}$ as k-th column vector of $N \times K$ matrix $\boldsymbol{B}$, $\{\boldsymbol{W}_m\}$ as m-th column vector of $K \times M$ matrix W, the learning task is to estimate the posterior distribution of latent variables $\boldsymbol{B}$, $\boldsymbol{W}$, and $\{\mathbf{z}_m\}$ given $N \times M$ word-document matrix $\widetilde{\boldsymbol{H}}$, which is the only observed variable where m-th column corresponds to the empirical word frequencies in the training documents m.

[9] recovers word-topic matrix $\boldsymbol{B}$ and topic-topic matrix $\boldsymbol{A} = \mathbb{E}[\boldsymbol{W}\boldsymbol{W}^T]$ instead of $\boldsymbol{W}$ in the spirit of nonnegative matrix factorization. Though the true underlying word distribution for each document is unknown and could be far from the sample observation $\widetilde{\boldsymbol{H}}$, the empirical word-word matrix $\boldsymbol{C}$ converges to its expectation $\boldsymbol{B}\mathbb{E}[\boldsymbol{W}\boldsymbol{W}^T]\boldsymbol{B}^T = \boldsymbol{B}\boldsymbol{A}\boldsymbol{B}^T$ as the number of documents increases. Thus the learning task is to approximately recover $\boldsymbol{B}$ and $\boldsymbol{A}$ pretending that the empirical $\boldsymbol{C}$ is close to the true second-order moment matrix.

The critical assumption for this method is to suppose that every topic k has a specific anchor word s_k that occurs with non-negligible probability (> 0) only in that topic. The anchor word s_k need not always appear in every document about the topic k, but we can be confident that the document is at least to some degree about the topic k if it contains s_k. This assumption drastically improves inference by guaranteeing the presence of a diagonal sub-matrix inside the word-topic

matrix A. After constructing an estimate $\boldsymbol{C}$, the algorithm in [9] first finds a set $S = \{s_1, ..., s_K\}$ of K anchor words (K is user-specified), and recovers $\boldsymbol{B}$ and $\boldsymbol{A}$ subsequently based on S. Due to this structure, overall performance depends heavily on the quality of anchor words.

In the matrix algebra literature this greedy anchor finding method is called *QR with row-pivoting*. Previous work classifies a matrix into two sets of row (or column) vectors where the vectors in one set can effectively reconstruct the vectors in another set, called *subset-selection algorithms*. [35] suggest one important deterministic algorithm. A randomized algorithm provided by [17] is the state-of-the art using a pre-stage that selects the candidates in addition to [35]. We found no change in anchor selection using these algorithms, verifying the difficulty of the anchor finding process. This difficulty is mostly because anchors must be nonnegative convex bases, whereas the classified vectors from the subset selection algorithms yield unconstrained bases.

The t-SNE model has previously been used to display high-dimensional embeddings of words in 2D space by Turian.[1] Low-dimensional embeddings of topic spaces have also been used to support user interaction with models: [29] use a visual display of a topic embedding to create a navigator interface. Although t-SNE has been used to visualize the *results* of topic models, for example by [48] and [86], we are not aware of any use of the method as a fundamental component of topic inference.

[1] http://metaoptimize.com/projects/wordreprs/

1.3 Low-dimensional Embeddings

Real text corpora typically involve vocabularies in the tens of thousands of distinct words. As the input matrix $\boldsymbol{C}$ scales quadratically with N, the Anchor Words algorithm must depend on a low-dimensional projection of $\boldsymbol{C}$ in order to be practical. Previous work [9] uses random projections via either Gaussian random matrices [43] or sparse random matrices [1], reducing the representation of each word to around 1,000 dimensions. Since the dimensionality of the compressed word co-occurrence space is an order of magnitude larger than K, we must still approximate the convex hull by choosing extreme points as before.

In this work we explore two projection methods: PCA and t-SNE [79]. Principle Component Analysis (PCA) is a commonly-used dimensionality reduction scheme that linearly transforms the data to new coordinates where the largest variances are orthogonally captured for each dimension. By choosing only a few such principle axes, we can represent the data in a lower dimensional space. In contrast, t-SNE embedding performs a non-linear dimensionality reduction preserving the local structures. Given a set of points $\{\mathbf{x}_i\}$ in a high-dimensional space X, t-SNE allocates probability mass for each pair of points so that pairs of similar (closer) points become more likely to co-occur than dissimilar (distant) points.

$$p_{j|i} = \frac{\exp(-d(\mathbf{x}_i, \mathbf{x}_j)^2/2\sigma_i^2)}{\sum_{k \neq i} \exp(-d(\mathbf{x}_i, \mathbf{x}_k)^2/2\sigma_i^2)} \tag{1.1}$$

$$p_{ij} = \frac{p_{j|i} + p_{i|j}}{2N} \quad \text{(symmetrized)} \tag{1.2}$$

Then it generates a set of new points $\{\mathbf{y}_i\}$ in low-dimensional space Y so that probability distribution over points in Y behaves similarly to the distribution

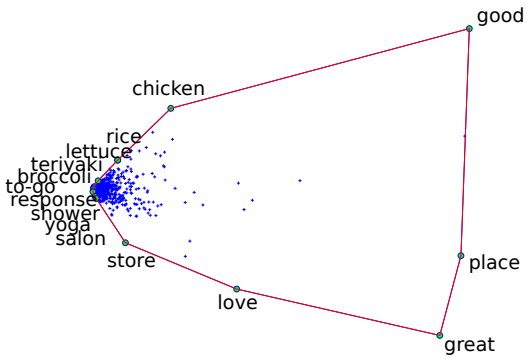

Figure 1.2: 2D PCA projections of a Yelp review corpus and its convex hulls.

over points in X by minimizing KL-divergence between two distributions:

$$q_{ij} = \frac{(1 + \|\mathbf{y}_i - \mathbf{y}_j\|^2)^{-1}}{\sum_{k \neq l}(1 + \|\mathbf{y}_k - \mathbf{y}_l\|^2)^{-1}} \tag{1.3}$$

$$min\ KL(P||Q) = \sum_{i \neq j} p_{ij} \log \frac{p_{ij}}{q_{ij}} \tag{1.4}$$

Instead of approximating a convex hull in such a high-dimensional space, we select the exact vertices of the convex hull formed in a low-dimensional projected space, which can be calculated efficiently. Figures 1.1 and 1.2 show the convex hulls for 2D projections of $\boldsymbol{C}$ using t-SNE and PCA for a corpus of Yelp reviews. Figure 1.3 illustrates the convex hulls for 3D t-SNE projection for the same corpus. Anchor words correspond to the vertices of these convex hulls. Note that we present the 2D projections as illustrative examples only; we find that three dimensional projections perform better in practice.

In addition to the computational advantages, this approach benefits anchor-based topic modeling in two aspects. First, as we now compute the exact convex hull, the number of topics depends on the dimensionality of the embedding, v. For example in the figures, 2D projection has 21 vertices, whereas 3D projection

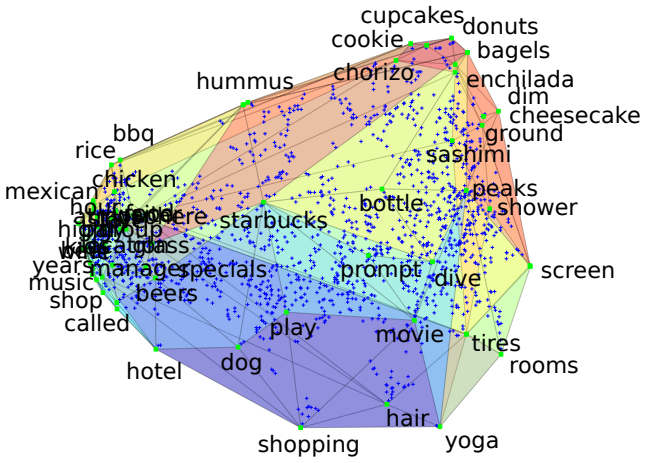

Figure 1.3: 3D *t*-SNE projection of a Yelp review corpus and its convex hull. Vertices on the convex hull correspond to anchor words.

supports 69 vertices. This implies users can easily tune the granularity of topic clusters by varying $v = 2, 3, 4, ...$ without increasing the number of topics by one each time. Second, we can effectively visualize the thematic relationships between topic anchors and the rest of words in the vocabulary, enhancing both interpretability and options for further vocabulary curation.

1.4 Experimental Results

We find that radically low-dimensional *t*-SNE projections are effective at finding anchor words that are much more salient than the greedy method, and topics that are more distinctive, while maintaining comparable held-out likelihood and semantic coherence. As noted in Section 1.1, the previous greedy anchor words algorithm tends to produce many nearly identical topics. For example, 37 out of 100 topics trained on a 2008 political blog corpus have *obama, mccain,*

bush, iraq or *palin* as their most probable word, including 17 just for *obama*. Only 66% of topics have a unique top word. In contrast, the *t*-SNE model on the same dataset has only one topic whose most probable word is *obama*, and 86% of topics have a unique top word (*mccain* is the most frequent top word, with five topics).

We use three real datasets: business reviews from the Yelp Academic Dataset,[2] political blogs from the 2008 US presidential election [30], and New York Times articles from 2007.[3] Details are shown in Table 1.1. Documents with fewer than 10 word tokens are discarded due to possible instability in constructing $\boldsymbol{C}$. We perform minimal vocabulary curation, eliminating a standard list of English stopwords[4] and terms that occur below frequency cutoffs: 100 times (Yelp, Blog) and 150 times (NYT). We further restrict possible anchor words to words that occur in more than three documents. As our datasets are not artificially synthesized, we reserve 5% from each set of documents for held-out likelihood computation.

Name	Documents	Vocab.	Avg. length
Yelp	20,000	1,606	40.6
Blog	13,000	4,447	161.3
NYT	41,000	10,713	277.8

Table 1.1: Statistics for datasets used in experiments

Unlike [9], which presents results on synthetic datasets to compare performance across different *recovery* methods given increasing numbers of documents, we are are interested in comparing *anchor finding* methods, and are mainly concerned with semantic quality. As a result, although we have

[2]https://www.yelp.com/academic_dataset
[3]http://catalog.ldc.upenn.edu/LDC2008T19
[4]We used the list of 524 stop words included in the Mallet library.

conducted experiments on synthetic document collections,[5] we focus on real datasets for this work. We also choose to compare only anchor finding algorithms, so we do not report comparisons to likelihood-based methods, which can be found in [9].

For both PCA and *t*-SNE, we use three-dimensional embeddings across all experiments. This projection results in matrices that are 0.03% as wide as the original $N \times N$ matrix for the NYT dataset. Without low-dimensional embedding, each word is represented by a N-dimensional vector where only several terms are non-zero due to the sparse co-occurrence patterns. Thus a vertex captured by the greedy anchor-finding method is likely to be one of many eccentric vertices in very high-dimensional space. In contrast, *t*-SNE creates an effective dense representation where a small number of pivotal vertices are more clearly visible, improving both performance and interpretability.

Note that since we can find an *exact* convex hull in these spaces,[6] there is an upper bound to the number of topics that can be found given a particular projection. If more topics are desired, one can simply increase the dimensionality of the projection. For the greedy algorithm we use sparse random projections with 1,000 dimensions with 5% negative entries and 5% positive entries. PCA and *t*-SNE choose (49, 32, 47) and (69, 77, 107) anchors, respectively for each of three Yelp, Blog, and NYTimes datasets.

[5]None of the algorithms are particularly effective at finding synthetically introduced anchor words possibly because there are other candidates around anchor vertices that approximate the convex hull to almost the same degree.

[6]In order to efficiently find an exact convex hull, we use the *Quickhull* algorithm.

1.4.1 Anchor-word Selection

We begin by comparing the behavior of low-dimensional embeddings to the behavior of the standard greedy algorithm. Table 1.2 shows ordered lists of the first 12 anchor words selected by three algorithms: *t*-SNE embedding, PCA embedding, and the original greedy algorithm. Anchor words selected by *t*-SNE (*police, business, court*) are more general than anchor words selected by the greedy algorithm (*cavalry, al-sadr, yiddish*). Additional examples of anchor words and their associated topics are shown in Table 1.3 and discussed in Section 1.4.2.

#	t-SNE	PCA	Greedy
1	police	beloved	cavalry
2	bonds	york	biodiesel
3	business	family	h/w
4	day	loving	kingsley
5	initial	late	mourners
6	million	president	pcl
7	article	people	carlisle
8	wife	article	al-sadr
9	site	funeral	kaye
10	mother	million	abc's
11	court	board	yiddish
12	percent	percent	great-grandmother

Table 1.2: The first 12 anchor words selected by three algorithms for the NYT corpus.

The Gram-Schimdt process used by Arora et al. greedily selects anchors in high-dimensional space. As each word is represented within V-dimensions, finding the word that has the next most distinctive co-occurrence pattern tends to prefer overly eccentric words with only short, intense bursts of co-occurring words. While the bases corresponding to these anchor words could be theoretically relevant for the original space in high-dimension, they are less likely to

be equally important in low-dimensional space. Thus projecting down to low-dimensional space can rearrange the points emphasizing not only uniqueness, but also longevity, achieving the ability to form measurably more specific topics.

Concretely, neither *cavalry, al-sadr, yiddish* nor *police, business, court* are full representations of New York Times articles, but the latter is a much better basis than the former due to its greater generality. We see the effect of this difference in the specificity of the resulting topics (for example in 17 *obama* topics). Most words in the vocabulary have little connection to the word *cavalry*, so the probability $p(z|x)$ does not change much across different x. When we convert these distributions into $P(x|z)$ using the Bayes' rule, the resulting topics are very close to the corpus distribution, a unigram distribution $p(x)$.

$$p(x|z = k_{cavalry}) \propto p(z = k_{cavalry}|x)p(x) \approx p(x) \tag{1.5}$$

This lack of specificity results in the observed similarity of topics.

1.4.2 Quantitative Results

In this section we compare PCA and t-SNE projections to the greedy algorithm along several quantitative metrics. To show the effect of different values of K, we report results for varying numbers of topics. As the anchor finding algorithms are deterministic, the anchor words in a K-dimensional model are identical to the first K anchor words in a $(K + 1)$-dimensional model. For the greedy algorithm we select anchor words in the order they are chosen. For the PCA and t-SNE methods, which find anchors jointly, we sort words in descending order by their distance from their centroid.

Recovery Error. Each non-anchor word is approximated by a convex combination of the K anchor words. The projected gradient algorithm [9] determines these convex coefficients so that the gap between the original word vector and the approximation becomes minimized. As choosing a good basis of K anchor words decreases this gap, Recovery Error (RE) is defined by the average ℓ_2-residuals across all words.

$$RE = \frac{1}{N} \sum_{i=1}^{N} \|\overline{C}_i - \sum_{k=1}^{K} p(z_1 = k | x_1 = i) \overline{C}_{S_k}\|_2 \tag{1.6}$$

Recovery error decreases with the number of topics, and improves substan-

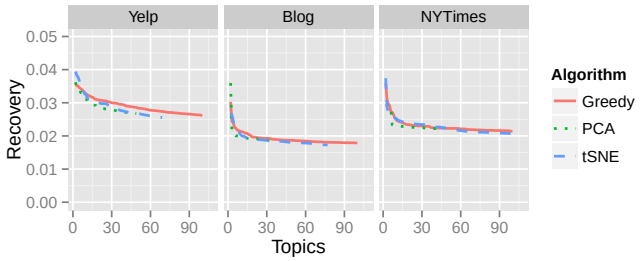

Figure 1.4: Recovery error is similar across algorithms.

tially after the first 10–15 anchor words for all methods. The t-SNE method has slightly better performance than the greedy algorithm, but they are similar. Results for recovery with the original, unprojected matrix (not shown) are much worse than the other algorithms, suggesting that the initial anchor words chosen are especially likely to be uninformative.

Normalized Entropy. As shown in Eq. 1.5, if the probability of topics given a word is close to uniform, the probability of that word in topics will be close to the corpus distribution. Normalized Entropy (NE) measures the entropy of this distribution relative to the entropy of a K-dimensional uniform distribution:

$$NE = \frac{1}{N} \sum_{i=1}^{N} \frac{H(z|x = i)}{\log K}. \tag{1.7}$$

The normalized entropy of topics given word distributions usually decreases

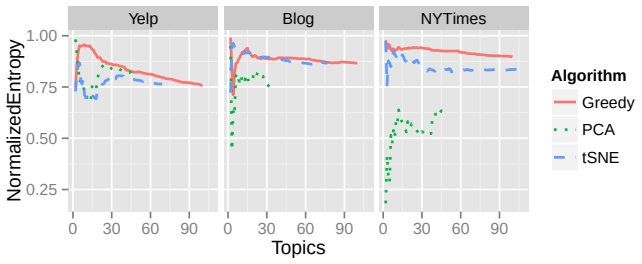

Figure 1.5: Words have higher topic entropy in the greedy model, especially in NYT, resulting in less specific topics.

as we add more topics, although both *t*-SNE and PCA show a dip in entropy for low numbers of topics. This result indicates that words become more closely associated with particular topics as we increase the number of topics. The low-dimensional embedding methods (*t*-SNE and PCA) have consistently lower entropy.

Topic Specificity and **Topic Dissimilarity.** We want topics to be both specific (that is, not overly general) and different from each other. When there are insufficient number of topics, $p(x|z)$ often resembles the corpus distribution $p(x)$, where high frequency terms become the top words contributing to most topics. Topic Specificity (TS) is defined by the average KL divergence from each topic's conditional distribution to the corpus distribution.[7]

$$TS = \frac{1}{K}\sum_{k=1}^{K} KL(p(x|z=k) \parallel p(x)) \tag{1.8}$$

One way to define the distance between multiple points is the minimum radius of a ball that covers every point. Whereas this is simply the distance form the centroid to the farthest point in the Euclidean space, it is an itself difficult

[7]We prefer *specificity* to [2]'s term *vacuousness* because the metric increases as we move away from the corpus distribution.

optimization problem to find such centroid of distributions under metrics such as KL-divergence and Jensen-Shannon divergence. To avoid this problem, we measure Topic Dissimilarity (TD) viewing each conditional distribution $p(x|z)$ as a simple N-dimensional vector in $\mathbb{R}^N$. Recall $\boldsymbol{B}_{ik} = p(x = i|z = k)$,

$$TD = \max_{1 \le k \le K} \|\frac{1}{K} \sum_{k'=1}^{K} \boldsymbol{B}_{*k'} - \boldsymbol{B}_{*k}\|_2. \tag{1.9}$$

Specificity and dissimilarity increase with the number of topics, suggesting that

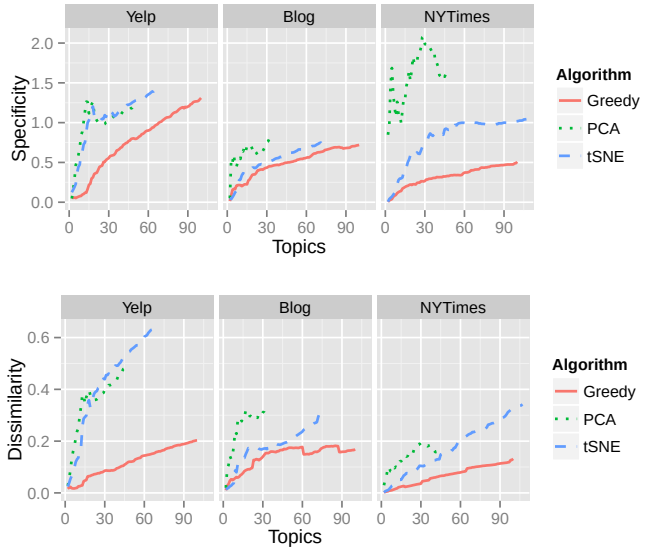

Figure 1.6: Greedy topics look more like the corpus distribution and more like each other.

with few anchor words, the topic distributions are close to the overall corpus distribution and very similar to one another. The *t*-SNE and PCA algorithms produce consistently better specificity and dissimilarity, indicating that they produce more useful topics early with small numbers of topics. The greedy algorithm produces topics that are closer to the corpus distribution and less distinct from each other (17 *obama* topics).

Topic Coherence is known to correlate with the semantic quality of topic

judged by human annotators [63]. Let $\mathcal{X}_k^{(T)}$ be T most probable words (i.e., top words) for the topic k.

$$TC = \sum_{x_1 \neq x_2 \in \mathcal{X}_k^{(T)}} \log \frac{D(x_1, x_2) + \epsilon}{D(x_1)} \tag{1.10}$$

Here $D(x_1, x_2)$ is the co-document frequency, which is the number of documents in M consisting of two words x_1 and x_2 simultaneously. $D(x)$ is the simple document frequency with the word x. The numerator contains smoothing count ϵ in order to avoid taking the logarithm of zero. Coherence scores for t-SNE and

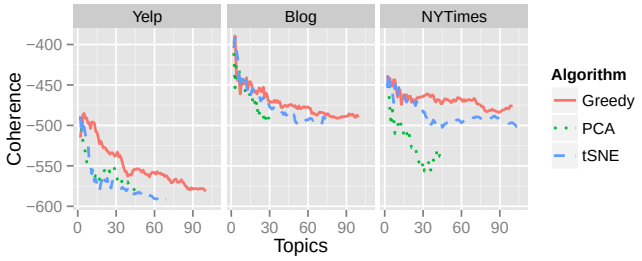

Figure 1.7: The greedy algorithm creates more coherent topics (higher is better), but at the cost of many overly general or repetitive topics.

PCA are worse than those for the greedy method, but this result must be understood in combination with the Specificity and Dissimilarity metrics. The most frequent terms in the overall corpus distribution $p(x)$ often appear together in documents. Thus a model creating many topics similar to the corpus distribution is likely to achieve high Coherence, but low Specificity by definition.

Saliency. [24] define saliency for topic words as a combination of distinctiveness and probability within a topic. Anchor words are distinctive by construction, so we can increase saliency by selecting more probable anchor words. We measure the probability of anchor words in two ways. First, we report the zero-based rank of anchor words within their topics. Examples of this metric,

which we call "hard" rank are shown in Table 1.3. The hard rank of the anchors in the PCA and *t*-SNE models are close to zero, while the anchor words for the greedy algorithm are much lower ranked, well below the range usually displayed to users. Second, while hard rank measures the perceived difference in rank of contributing words, position may not fully capture the relative importance of the anchor word. "Soft" rank quantifies the average log ratio between probabilities of the prominent word x_k^* and the anchor word s_k.

$$SR = \frac{1}{K} \sum_{k=1}^{K} \log \frac{p(x = x_k^* | z = k)}{p(x = s_k | z = k)} \tag{1.11}$$

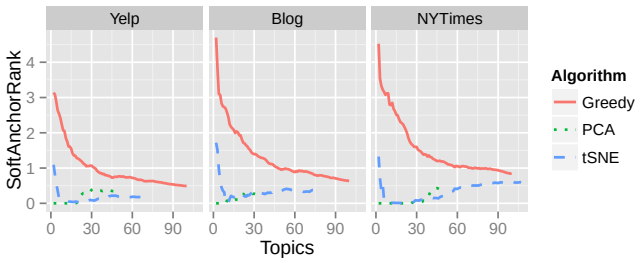

Figure 1.8: Anchor words have higher probability, and therefore greater salience, in *t*-SNE and PCA models (1 ≈ one third the probability of the top ranked word).

Lower values of soft rank (Fig. 1.8 indicate that the anchor word has greater relative probability to occur within a topic. As we increase the number of topics, anchor words become more prominent in topics learned by the greedy method, but *t*-SNE anchor words remain relatively more probable within their topics as measured by soft rank.

Held-out Probability. Given an estimate of the topic-word matrix A, we can compute the marginal probability of held-out documents under that model. We use the left-to-right estimator introduced by [80], which uses a sequential algorithm similar to a Gibbs sampler. This method requires a smoothing pa-

rameter for document-topic Dirichlet distributions, which we set to $\alpha_k = 0.1$. We note that the greedy algorithm run on the original, unprojected matrix has

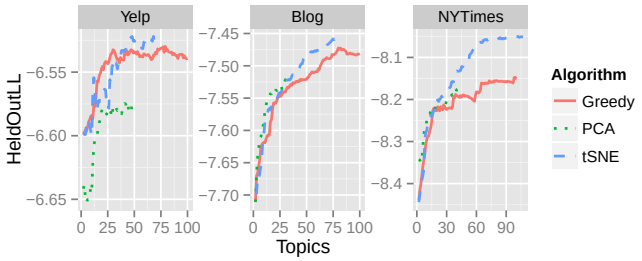

Figure 1.9: *t*-SNE topics have better held-out probability than greedy topics.

better held-out probability values than *t*-SNE for the Yelp corpus, but as this method does not scale to realistic vocabularies we compare here to the sparse random projection method used in [9]. The *t*-SNE method appears to do best, particularly in the NYT corpus, which has a larger vocabulary and longer training documents. The length of individual held-out documents has no correlation with held-out probability.

We emphasize that Held-out Probability is sensitive to smoothing parameters and should only be used in combination with a range of other topic-quality metrics. In initial experiments, we observed significantly worse held-out performance for the *t*-SNE algorithm. This phenomenon was because setting the probability of anchor words to zero for all but their own topics led to large negative values in held-out log probability for those words. As *t*-SNE tends to choose more frequent terms as anchor words, these "spikes" significantly affected overall probability estimates. To make the calculation more fair, we added 10^{-5} to any zero entries for anchor words in the word-topic matrix $\boldsymbol{B}$ across *all* models and renormalized.

Because *t*-SNE is a stochastic model, different initializations can result in different embeddings. To evaluate how steady anchor word selection is, we ran five random initializations for each dataset. For the Yelp dataset, the number of anchor words varies from 59 to 69, and 43 out of those are shared across at least four trials. For the Blog dataset, the number of anchor words varies from 80 to 95, with 56 shared across at least four trials. For the NYT dataset, this number varies between 83 and 107, with 51 shared across at least four models.

1.4.3 Qualitative Results

Table 1.3 shows topics trained by three methods (*t*-SNE, PCA, and greedy) for all three datasets. For each model, we select five topics *at random* from the *t*-SNE model, and then find the closest topic from each of the other models. If anchor words present in the top eight words, they are shown in boldface.

A fundamental difference between anchor-based inference and traditional likelihood-based inference is that we can give an *order* to topics according to their contribution to word co-occurrence convex hull. This order is intrinsic to the original algorithm, and we heuristically give orders to *t*-SNE and PCA based on their contributions. This order is listed as # in the previous table. For all but one topic, the closest topic from the greedy model has a higher order number than the associated *t*-SNE topic. As shown above, the standard algorithm tends to pick less useful anchor words at the initial stage; only the later, higher ordered topics are specific.

The most clear distinction between models is the rank of anchor words represented by Hard Rank for each topic. Only one topic corresponding to (*initial*)

has the anchor word which does not coincide with the top-ranked word. For the greedy algorithm, anchor words are often tens of words down the list in rank, indicating that they are unlikely to find a connection to the topic's semantic core. In cases where the anchor word is highly ranked (*unbelievers*, *parenthood*) the word is a good indicator of the topic, but still less decisive.

t-SNE and PCA are often consistent in their selection of anchor words, which provides useful validation that low-dimensional embeddings discern more relevant anchor words regardless of linear vs non-linear projections. Note that we are only varying the anchor selection part of the Anchor Words algorithm in these experiments, recovering topic-word distributions in the same manner given anchor words. As a result, any differences between topics with the same anchor word (for example *chicken*) are due to the difference in either the number of topics or the rest of anchor words. Since PCA suffers from a crowding problem in lower-dimensional projection (see Figure 1.2) and the problem could be severe in a dataset with a large vocabulary, *t*-SNE is more likely to find the proper number of anchors given a specified granularity.

1.5 Conclusion

One of the main advantages of the anchor words algorithm is that the running time is largely independent of corpus size. Adding more documents would not affect the size of the co-occurrence matrix, requiring more times to construct the co-occurrence matrix at the beginning. While the inference is scalable depending only on the size of the vocabulary, finding quality anchor words is crucial for the performance of the inference.

[9] presents a greedy anchor finding algorithm that improves over previous linear programming methods, but finding quality anchor words remains an open problem in spectral topic inference. We have shown that previous approaches have several limitations. Exhaustively finding anchor words by eliminating words that are reproducible by other words [8] is impractical. The anchor words selected by the greedy algorithm are overly eccentric, particularly at the early stages of the algorithm, causing topics to be poorly differentiated. We find that using low-dimensional embeddings of word co-occurrence statistics allows us to approximate a better convex hull. The resulting anchor words are highly *salient*, being both distinctive and probable. The models trained with these words have better quantitative and qualitative properties along various metrics. Most importantly, using radically low-dimensional projections allows us to provide users with clear visual explanations for the model's anchor word selections.

An interesting property of using low-dimensional embeddings is that the number of topics depends only on the projecting dimension. Since we can efficiently find an exact convex hull in low-dimensional space, users can achieve topics with their preferred level of granularities by changing the projection dimension. We do not insist this is the "correct" number of topics for a corpus, but this method, along with the range of metrics described in this paper, provides users with additional perspective when choosing a dimensionality that is appropriate for their needs.

We find that the *t*-SNE method, besides its well-known ability to produce high quality layouts, provides the best overall anchor selection performance. This method consistently selects higher-frequency terms as anchor words, re-

sulting in greater clarity and interpretability. Embeddings with PCA are also effective, but they result in less well-formed spaces, being less effective in held-out probability for sufficiently large corpora.

Anchor word finding methods based on low-dimensional projections offer several important advantages for topic model users. In addition to producing more salient anchor words that can be used effectively as topic labels, the relationship of anchor words to a visualizable word co-occurrence space offers significant potential. Users who can see why the algorithm chose a particular model will have greater confidence in the model and in any findings that result from topic-based analysis. Finally, visualizable spaces offer the potential to produce interactive environments for semi-supervised topic reconstruction.

Type	#	HR	Top Words **(Yelp)**
t-SNE	16	0	**mexican** good service great eat restaurant authentic delicious
PCA	15	0	**mexican** authentic eat chinese don't restaurant fast salsa
Greedy	34	35	good great food place service restaurant it's mexican
t-SNE	6	0	**beer** selection good pizza great wings tap nice
PCA	39	6	wine beer selection nice list glass **wines** bar
Greedy	99	11	beer selection great happy place wine good bar
t-SNE	3	0	**prices** great good service selection price nice quality
PCA	12	0	**atmosphere** prices drinks friendly selection nice beer ambiance
Greedy	34	35	good great food place service restaurant it's mexican
t-SNE	10	0	**chicken** salad good lunch sauce ordered fried soup
PCA	10	0	**chicken** salad lunch fried pita time back sauce
Greedy	69	12	chicken rice sauce fried ordered i'm spicy soup
Type	#	HR	Top Words **(Blog)**
t-SNE	10	0	**hillary** clinton campaign democratic bill party win race
PCA	4	0	**hillary** clinton campaign democratic party bill democrats vote
Greedy	45	19	obama hillary campaign clinton obama's barack it's democratic
t-SNE	3	0	**iraq** war troops iraqi mccain surge security american
PCA	9	1	iraq **iraqi** war troops military forces security american
Greedy	91	8	iraq mccain war bush troops withdrawal obama iraqi
t-SNE	9	0	**allah** muhammad qur verses unbelievers ibn muslims world
PCA	18	14	allah muhammad qur verses unbelievers story time update
Greedy	4	5	allah muhammad people qur verses **unbelievers** ibn sura
t-SNE	19	0	**catholic** abortion catholics life hagee time biden human
PCA	2	0	**people** it's time don't good make years palin
Greedy	40	1	abortion **parenthood** planned people time state life government
Type	#	HR	Top Words **(NYT)**
t-SNE	0	0	**police** man yesterday officers shot officer year-old charged
PCA	6	0	**people** it's police way those three back don't
Greedy	50	198	police man yesterday officers officer people street city
t-SNE	19	0	**senator** republican senate democratic democrat state bill
PCA	33	2	state republican **republicans** senate senator house bill party
Greedy	85	33	senator republican president state campaign presidential people
t-SNE	2	0	**business** chief companies executive group yesterday billion
PCA	21	0	**billion** companies business deal group chief states united
Greedy	55	10	radio business companies percent day music article satellite
t-SNE	14	0	**market** sales stock companies prices billion investors price
PCA	11	0	**percent** market rate week state those increase high
Greedy	77	44	companies percent billion million group business chrysler people

Table 1.3: Example *t*-SNE topics and their most similar topics across algorithms. The Greedy algorithm can find similar topics, but the anchor words are much less salient.

CHAPTER 2

RECTIFICATION FOR ROBUST LOW-RANK SPECTRAL TOPIC INFERENCE

Robust Spectral inference provides fast algorithms and provable optimality for latent topic analysis. But for real data these algorithms require additional ad-hoc heuristics, and even then often produce unusable results. We explain this poor performance by casting the problem of topic inference in the framework of Joint Stochastic Matrix Factorization (JSMF) and showing that previous methods violate the theoretical conditions necessary for a good solution to exist. We then propose a novel rectification method that learns high quality topics and their interactions even on small, noisy data. This method achieves results comparable to probabilistic techniques in several domains while maintaining scalability and provable optimality.

2.1 Introduction

Summarizing large data sets using pairwise co-occurrence frequencies is a powerful tool for data mining. Objects can often be better described by their relationships than their inherent characteristics. Communities can be discovered from friendships [65], song genres can be identified from co-occurrence in playlists [23], and neural word embeddings are factorizations of pairwise co-occurrence information [70, 53]. Recent *Anchor Word* algorithms [8, 9] perform spectral inference on co-occurrence statistics for inferring topic models [39, 15]. Co-occurrence statistics can be calculated using a single parallel pass through a training corpus. While these algorithms are fast, deterministic, and provably

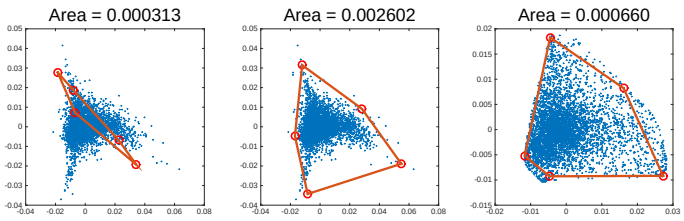

Figure 2.1: 2D visualizations show the low-quality convex hull found by Anchor Words [9] (left) and a better convex hull (middle) found by discovering anchor words on a rectified space (right).

guaranteed, they are sensitive to observation noise and small samples, often producing effectively useless results on real documents that present no problems for probabilistic algorithms.

We cast this general problem of learning overlapping latent clusters as Joint-Stochastic Matrix Factorization (JSMF), a subset of non-negative matrix factorization that contains topic modeling as a special case. We explore the conditions necessary for inference from co-occurrence statistics and show that the Anchor Words algorithms necessarily violate such conditions. Then we propose a rectified algorithm that matches the performance of probabilistic inference—even on small and noisy datasets—without losing efficiency and provable guarantees. Validating on both real and synthetic data, we demonstrate that our rectification not only produces better clusters, but also, unlike previous work, learns meaningful cluster interactions.

Let the matrix $\boldsymbol{C}$ represent the co-occurrence of pairs drawn from N objects: $\boldsymbol{C}_{ij}$ is the joint probability $p(X_1 = i, X_2 = j)$ for a pair of objects i and j. Our goal is to discover K latent clusters by approximately decomposing $\boldsymbol{C} \approx \boldsymbol{B}\boldsymbol{A}\boldsymbol{B}^T$. $\boldsymbol{B}$ is the object-cluster matrix, in which each column corresponds to a cluster and $\boldsymbol{B}_{ik} = p(X = i|Z = k)$ is the probability of drawing an object i conditioned on the

Domain	Object	Cluster	Basis
Document	Word	Topic	Anchor Word
Image	Pixel	Segment	Pure Pixel
Network	User	Community	Representative
Legislature	Member	Party/Group	Partisan
Playlist	Song	Genre	Signature Song

Figure 2.2: JSMF applications, with anchor-word equivalents.

object belonging to the cluster k; and $\boldsymbol{A}$ is the cluster-cluster matrix, in which $\boldsymbol{A}_{kl} = p(Z_1 = k, Z_2 = l)$ represents the joint probability of pairs of clusters. We call the matrices C and A *joint-stochastic* (i.e., $\boldsymbol{C} \in \mathcal{JS}_N, \boldsymbol{A} \in \mathcal{JS}_K$) due to their correspondence to joint distributions; $\boldsymbol{B}$ is *column-stochastic*. Example applications are shown in Table 2.1.

Anchor Word algorithms [8, 9] solve JSMF problems using a separability assumption: each topic contains at least one "anchor" word that has non-negligible probability exclusively in that topic. The algorithm uses the co-occurrence patterns of the anchor words as a summary basis for the co-occurrence patterns of all other words. The initial algorithm [8] is theoretically sound but unable to produce column-stochastic word-topic matrix $\boldsymbol{B}$ due to unstable matrix inversions. A subsequent algorithm [9] fixes negative entries in $\boldsymbol{B}$, but still produces large negative entries in the estimated topic-topic matrix $\boldsymbol{A}$. As shown in Figure 2.4, the proposed algorithm infers valid topic-topic interactions.

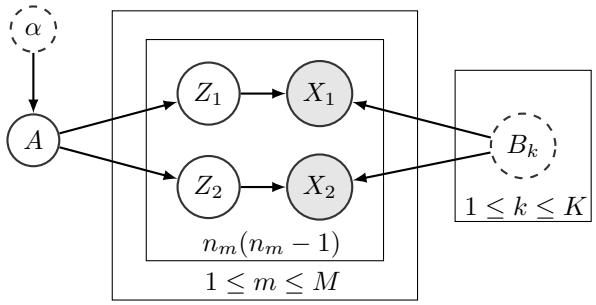

Figure 2.3: The JSMF event space differs from LDA's. JSMF deals only with pairwise co-occurrence events and does not generate observations/documents.

2.2 Requirements for Factorization

In this section we review the probabilistic and statistical structures of JSMF and then define geometric structures of co-occurrence matrices required for successful factorization. $\boldsymbol{C} \in \mathbb{R}^{N \times N}$ is a joint-stochastic matrix constructed from M training examples, each of which contain some subset of N objects. We wish to find $K \ll N$ latent clusters by factorizing $\boldsymbol{C}$ into a column-stochastic matrix $\boldsymbol{B} \in \mathbb{R}^{N \times K}$ and a joint-stochastic matrix $\boldsymbol{A} \in \mathbb{R}^{K \times K}$, satisfying $\boldsymbol{C} \approx \boldsymbol{B}\boldsymbol{A}\boldsymbol{B}^T$.

Probabilistic structure. Figure 2.3 shows the event space of our model. The distribution $\boldsymbol{A}$ over pairs of clusters is generated first from a stochastic process with a hyperparameter $\boldsymbol{\alpha}$. If the m-th training example contains a total of n_m objects, our model views the example as consisting of all possible $n_m(n_m - 1)$ pairs of objects.[1] For each of these pairs, cluster assignments are sampled from the selected distribution ($(z_1, z_2) \sim \boldsymbol{A}$). Then an actual object pair is drawn with respect to the corresponding cluster assignments ($x_1 \sim \boldsymbol{B}_{z_1}$, $x_2 \sim \boldsymbol{B}_{z_2}$). Note that this pro-

[1]Due to the bag-of-words assumption, every object can pair with any other object in that example, except itself. One implication of our work is better understanding the self-co-occurrences, the diagonal entries in the co-occurrence matrix.

cess does not explain how each training example is generated from a model, but shows how our model understands the objects in the training examples.

Following [8, 9], our model views $\boldsymbol{B}$ as a set of parameters rather than random variables.[2] The primary learning task is to estimate $\boldsymbol{B}$; we then estimate $\boldsymbol{A}$ to recover the hyperparameter $\boldsymbol{\alpha}$. Due to the conditional independence $X_1 \perp X_2 \mid (Z_1 \text{ or } Z_2)$, the factorization $\boldsymbol{C} \approx \boldsymbol{B}\boldsymbol{A}\boldsymbol{B}^T$ is equivalent to

$$p(X_1, X_2|\boldsymbol{A}; \boldsymbol{B}) = \sum_{z_1} \sum_{z_2} p(X_1|Z_1; \boldsymbol{B}) p(Z_1, Z_2|\boldsymbol{A}) p(X_2|Z_2; \boldsymbol{B}).$$

Under the *separability assumption*, each cluster k has a basis object s_k such that $p(X = s_k|Z = k) > 0$ and $p(X = s_k|Z \neq k) = 0$. In matrix terms, we assume the submatrix of $\boldsymbol{B}$ comprised of the rows with indices $S = \{s_1, \ldots, s_K\}$ is diagonal. As these rows form a non-negative basis for the row space of $\boldsymbol{B}$, the assumption implies $\text{rank}^+(\boldsymbol{B}) = K = \text{rank}(\boldsymbol{B})$.[3] Providing identifiability to the factorization, this assumption becomes crucial for inference of both $\boldsymbol{B}$ and $\boldsymbol{A}$. Note that JSMF factorization is unique up to column permutation, meaning that no specific ordering exists among the discovered clusters, equivalent to probabilistic topic models (see the Appendix).

Statistical structure. Let $f(\boldsymbol{\alpha})$ be a (known) distribution of distributions from which a cluster distribution is sampled for each training example. Saying $\boldsymbol{W}_m \sim f(\boldsymbol{\alpha})$, we have M *i.i.d* samples $\{\boldsymbol{W}_1, \ldots, \boldsymbol{W}_M\}$ which are not directly observable. Defining the posterior cluster-cluster matrix $\boldsymbol{A}^*_M = \frac{1}{M} \sum_{m=1}^{M} \boldsymbol{W}_m \boldsymbol{W}_m^T$ and

[2] In LDA, each column of B is generated from a known distribution $\boldsymbol{B}_k \sim Dir(\boldsymbol{\beta})$.

[3] $\text{rank}^+(\boldsymbol{B})$ means the non-negative rank of the matrix B, whereas $\text{rank}(\boldsymbol{B})$ means the usual rank.

the expectation $\boldsymbol{A}^* = \mathbb{E}[\boldsymbol{W}_m \boldsymbol{W}_m^T]$, Lemma 2.2 in [8] showed that[4]

$$\boldsymbol{A}_M^* \longrightarrow \boldsymbol{A}^* \quad \text{as} \quad M \longrightarrow \infty. \tag{2.1}$$

Denote the *posterior co-occurrence* for the m-th training example by $\boldsymbol{C}_m^*$ and all examples by $\boldsymbol{C}^*$. Then $\boldsymbol{C}_m^* = B W_m W_m^T B^T$, and $\boldsymbol{C}^* = \frac{1}{M} \sum_{m=1}^M C_m^*$. Thus

$$\boldsymbol{C}^* = \boldsymbol{B} \left(\frac{1}{M} \sum_{m=1}^{M} \boldsymbol{W}_m \boldsymbol{W}_m^T \right) \boldsymbol{B}^T = \boldsymbol{B} \boldsymbol{A}_M^* \boldsymbol{B}^T. \tag{2.2}$$

Denote the *noisy observation* for the m-th training example by $\boldsymbol{C}_m$, and all examples by $\boldsymbol{C}$. Let $\boldsymbol{W} = [\boldsymbol{W}_1 | ... | \boldsymbol{W}_M]$ be a matrix of topics. We will construct $\boldsymbol{C}_m$ so that $\mathbb{E}[\boldsymbol{C}|\boldsymbol{W}]$ is an *unbiased estimator* of $\boldsymbol{C}^*$. Thus as $M \to \infty$

$$\boldsymbol{C} \longrightarrow \mathbb{E}[\boldsymbol{C}] = \boldsymbol{C}^* = \boldsymbol{B} \boldsymbol{A}_M^* \boldsymbol{B}^T \longrightarrow \boldsymbol{B} \boldsymbol{A}^* \boldsymbol{B}^T. \tag{2.3}$$

Geometric structure. Though the separability assumption allows us to identify $\boldsymbol{B}$ even from the noisy observation $\boldsymbol{C}$, we need to throughly investigate the structure of cluster interactions. This is because it will eventually be related to how much useful information the co-occurrence between corresponding anchor bases contains, enabling us to best use our training data. Say $\mathcal{DNN}_n$ is the set of $n \times n$ *doubly non-negative* matrices: entrywise non-negative and positive semidefinite (PSD).

Claim $\boldsymbol{A}_M^*, \boldsymbol{A}^* \in \mathcal{DNN}_K$ and $\boldsymbol{C}^* \in \mathcal{DNN}_N$

Proof Take any vector $y \in \mathbb{R}^K$. As $\boldsymbol{A}_M^*$ is defined as a sum of outer-products,

$$y^T \boldsymbol{A}_M^* y = \frac{1}{M} \sum_{m=1}^{M} y^T \boldsymbol{W}_m \boldsymbol{W}_m^T y = \frac{1}{M} \sum (\boldsymbol{W}_m^T y)^T (\boldsymbol{W}_m^T y) = \sum (\textit{non-negative}) \geq 0.$$

[4] This convergence is not trivial while $\frac{1}{M} \sum_{m=1}^M \boldsymbol{W}_m \to \mathbb{E}[\boldsymbol{W}_m]$ as $M \to \infty$ by the Central Limit Theorem.

Thus $\boldsymbol{A}_M^* \in \mathcal{PSD}_K$. In addition, $(\boldsymbol{A}_M^*)_{kl} = p(Z_1 = k, Z_2 = l) \geq 0$ for all k, l. Proving $\boldsymbol{A}^* \in \mathcal{DNN}_K$ is analogous by the linearity of expectation. Relying on double non-negativity of $\boldsymbol{A}_M^*$, Equation (2.3) implies not only the low-rank structure of $\boldsymbol{C}^*$, but also double non-negativity of $\boldsymbol{C}^*$ by a similar proof (see the Appendix).

The Anchor Word algorithms in [8, 9] consider neither double non-negativity of cluster interactions nor its implication on co-occurrence statistics. Indeed, the empirical co-occurrence matrices collected from limited data are generally indefinite and full-rank, whereas the posterior co-occurrences must be positive semidefinite and low-rank. Our new approach will efficiently enforce double non-negativity and low-rankness of the co-occurrence matrix $\boldsymbol{C}$ based on the geometric property of its posterior behavior. We will later clarify how this process substantially improves the quality of the clusters and their interactions by eliminating noises and restoring missing information.

2.3 Rectified Anchor Words Algorithm

In this section, we describe how to estimate the co-occurrence matrix $\boldsymbol{C}$ from the training data, and how to rectify $\boldsymbol{C}$ so that it is low-rank and doubly non-negative. We then decompose the rectified $\boldsymbol{C}'$ in a way that preserves the doubly non-negative structure in the cluster interaction matrix.

Generating co-occurrence $\boldsymbol{C}$. Let $\boldsymbol{H}_m$ be the vector of object counts for the m-th training example, and let $p_m = \boldsymbol{B}\boldsymbol{W}_m$ where $\boldsymbol{W}_m$ is the document's latent topic distribution. Then $\boldsymbol{H}_m$ is assumed to be a sample from a multinomial distribu-

tion $\boldsymbol{H}_m \sim \text{Multi}(n_m, p_m)$ where $n_m = \sum_{i=1}^{N} \boldsymbol{H}_m^{(i)}$, and recall $\mathbb{E}[\boldsymbol{H}_m] = n_m p_m = n_m \boldsymbol{B}\boldsymbol{W}_m$ and $\text{Cov}(\boldsymbol{H}_m) = n_m \left(\text{diag}(p_m) - p_m p_m^T\right)$. As in [9], we generate the co-occurrence for the m-th example by

$$\boldsymbol{C}_m = \frac{\boldsymbol{H}_m \boldsymbol{H}_m^T - \text{diag}(\boldsymbol{H}_m)}{n_m(n_m - 1)}. \tag{2.4}$$

The diagonal penalty in Eq. 2.4 cancels out the diagonal matrix term in the variance-covariance matrix, making the estimator unbiased. Putting $d_m = n_m(n_m - 1)$, that is $\mathbb{E}[\boldsymbol{C}_m|\boldsymbol{W}_m] = \frac{1}{d_m}\mathbb{E}[\boldsymbol{H}_m \boldsymbol{H}_m^T] - \frac{1}{d_m}\text{diag}(\mathbb{E}[\boldsymbol{H}_m]) = \frac{1}{d_m}(\mathbb{E}[\boldsymbol{H}_m]\mathbb{E}[\boldsymbol{H}_m]^T + \text{Cov}(\boldsymbol{H}_m) - \text{diag}(\mathbb{E}[\boldsymbol{H}_m])) = B(\boldsymbol{W}_m \boldsymbol{W}_m^T)\boldsymbol{B}^T \equiv \boldsymbol{C}_m^*$. Thus $\mathbb{E}[\boldsymbol{C}|\boldsymbol{W}] = \boldsymbol{C}^*$ by the linearity of expectation.

Rectifying co-occurrence $\boldsymbol{C}$. While $\boldsymbol{C}$ is an unbiased estimator for $\boldsymbol{C}^*$ in our model, in reality the two matrices often differ due to a mismatch between our model assumptions and the data[5] or due to error in estimation from limited data. The computed $\boldsymbol{C}$ is generally full-rank with many negative eigenvalues, causing a large approximation error. As the posterior co-occurrence $\boldsymbol{C}^*$ must be low-rank, doubly non-negative, and joint-stochastic, we propose two rectification methods: Diagonal Completion (DC) and Alternating Projection (AP). DC modifies only diagonal entries so that $\boldsymbol{C}$ becomes low-rank, non-negative, and joint-stochastic; while AP enforces modifies every entry and enforces the same properties as well as positive semi-definiteness. As our empirical results strongly favor alternating projection, we defer the details of diagonal completion to the Appendix.

Based on the desired property of the posterior co-occurrence $\boldsymbol{C}^*$, we seek to project our estimator $\boldsymbol{C}$ onto the set of joint-stochastic, doubly non-negative,

[5]There is no reason to expect real data to be generated from topics, much less exactly K latent topics.

low rank matrices. Alternating projection methods like Dykstra's algorithm [18] allow us to project onto an intersection of finitely many convex sets using projections onto each individual set in turn. In our setting, we consider the intersection of three sets of symmetric $N \times N$ matrices: the elementwise non-negative matrices $\mathcal{NN}_N$, the normalized matrices $\mathcal{NOR}_N$ whose entry sum is equal to 1, and the positive semi-definite matrices with rank K, $\mathcal{PSD}_{NK}$. We project onto these three sets as follows:

$$\Pi_{\mathcal{PSD}_{NK}}(\boldsymbol{C}) = U\Lambda_K^+ U^T, \;\; \Pi_{\mathcal{NOR}_N}(\boldsymbol{C}) = \boldsymbol{C} + \frac{1 - \sum_{i,j} \boldsymbol{C}_{ij}}{N^2}\mathbf{1}\mathbf{1}^T, \;\; \Pi_{\mathcal{NN}_N}(\boldsymbol{C}) = \max\{\boldsymbol{C},\ 0\}.$$

where $\boldsymbol{C} = U\Lambda U^T$ is an eigendecomposition and Λ_K^+ is the matrix Λ modified so that all negative eigenvalues and any but the K largest positive eigenvalues are set to zero. Truncated eigendecompositions can be computed efficiently, and the other projections are likewise efficient. While $\mathcal{NN}_N$ and $\mathcal{NOR}_N$ are convex, $\mathcal{PSD}_{NK}$ is not. However, [54] show that alternating projection with a non-convex set still works under certain conditions, guaranteeing a local convergence. Thus iterating three projections in turn until the convergence rectifies $\boldsymbol{C}$ to be in the desired space. We will show how to satisfy such conditions and the convergence behavior in Section 2.5.

Selecting basis S. The first step of the factorization is to select the subset S of objects that satisfy the separability assumption. We want the K best rows of the row-normalized co-occurrence matrix $\boldsymbol{C}$ so that all other rows lie nearly in the convex hull of the selected rows. [9] use the Gram-Schmidt process to select anchors, which computes *pivoted QR decomposition*, but did not utilize the sparsity of $\boldsymbol{C}$. To scale beyond small vocabularies, they use random projections that approximately preserve ℓ_2 distances between rows of $\boldsymbol{C}$. For all experiments we

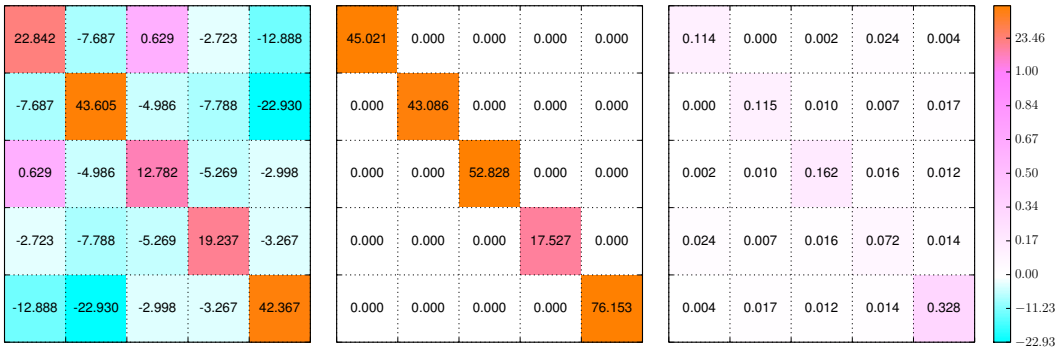

Figure 2.4: The algorithm of [9] (first panel) produces negative cluster co-occurrence probabilities. A probabilistic reconstruction alone (this paper & [8], second panel) removes negative entries but has no off-diagonals and does not sum to one. Trying after rectification (this paper, third panel) produces a valid joint stochastic matrix.

use a new pivoted QR algorithm (see the Appendix) that exploits sparsity instead of using random projections, and thus preserves deterministic inference.[6]

Recovering object-cluster $\boldsymbol{B}$. After finding the set of basis objects S, we can infer each entry of $\boldsymbol{B}$ by Bayes' rule as in [9]. Let $\{p(Z_1 = k|X_1 = i)\}_{k=1}^{K}$ be the coefficients that reconstruct the i-th row of $\boldsymbol{C}$ in terms of the basis rows corresponding to S. Since $\boldsymbol{B}_{ik} = p(X_1 = i|Z_1 = k)$, we can use the corpus frequencies $p(X_1 = i) = \sum_j \boldsymbol{C}_{ij}$ to estimate $\boldsymbol{B}_{ik} \propto p(Z_1 = k|X_1 = i)p(X_1 = i)$. Thus the main task for this step is to solve simplex-constrained QPs to infer a set of such coefficients for each object. We use an exponentiated gradient algorithm to solve the problem similar to [9]. Note that this step can be efficiently done in parallel for each object.

[6]To effectively use random projections, it is necessary to either find proper dimensions based on multiple trials or perform low-dimensional random projection multiple times [85] and merge the resulting anchors.

Recovering cluster-cluster $\boldsymbol{A}$. [9] recovered $\boldsymbol{A}$ by minimizing $\|\boldsymbol{C} - \boldsymbol{B}\boldsymbol{A}\boldsymbol{B}^T\|_F$; but the inferred $\boldsymbol{A}$ generally has many negative entries, failing to model the probabilistic interaction between topics. While we can further project $\boldsymbol{A}$ onto the joint-stochastic matrices, this produces a large approximation error.

We consider an alternate recovery method that again leverages the separability assumption. Let $\boldsymbol{C}_{SS}$ be the submatrix whose rows and columns correspond to the selected objects S, and let D be the diagonal submatrix $\boldsymbol{B}_{S*}$ of rows of $\boldsymbol{B}$ corresponding to S. Then

$$\boldsymbol{C}_{SS} = \boldsymbol{D}\boldsymbol{A}\boldsymbol{D}^T = \boldsymbol{D}\boldsymbol{A}\boldsymbol{D} \Longrightarrow \boldsymbol{A} = \boldsymbol{D}^{-1}\boldsymbol{C}_{SS}\boldsymbol{D}^{-1}. \tag{2.5}$$

This approach efficiently recovers a cluster-cluster matrix $\boldsymbol{A}$ mostly based on the co-occrrurence information between corresponding anchor basis, and produces no negative entries due to the stability of diagonal matrix inversion. Note that the principle submatrices of a PSD matrix are also PSD; hence, if $\boldsymbol{C} \in \mathcal{PSD}_N$ then $\boldsymbol{C}_{SS}, \boldsymbol{A} \in \mathcal{PSD}_K$. Thus, not only is the recovered $\boldsymbol{A}$ an unbiased estimator for $\boldsymbol{A}^*_M$, but also it is now doubly non-negative as $\boldsymbol{A}^*_M \in \mathcal{DNN}_K$ after the rectification.[7]

2.4 Experimental Results

Our Rectified Anchor Words algorithm with alternating projection fixes many problems in the baseline Anchor Words algorithm [9] while matching the performance of Gibbs sampling [34] and maintaining spectral inference's determinism and independence from corpus size. We evaluate direct measurement of matrix quality as well as indicators of topic utility. We use two text datasets:

[7] We later realized that essentially same approach was previously tried in [8], but it was not able to generate a valid topic-topic matrix as shown in the middle panel of Figure 2.4.

Dataset	M	N	Avg. Len
NIPS	1,348	5k	380.5
NYTimes	269,325	15k	204.9
Movies	63,041	10k	142.8
Songs	14,653	10k	119.2

Table 2.1: Statistics of four datasets.

NIPS full papers and New York Times news articles.[8] We eliminate a minimal list of 347 English stop words and prune rare words based on tf-idf scores and remove documents with fewer than five tokens after vocabulary curation. We also prepare two non-textual item-selection datasets: users' movie reviews from the Movielens 10M Dataset,[9] and music playlists from the complete Yes.com dataset.[10] We perform similar vocabulary curation and document tailoring, with the exception of frequent stop-object elimination. Playlists often contain the same songs multiple times, but users are unlikely to review the same movies more than once, so we augment the movie dataset so that each review contains $2 \times$ (*stars*) number of movies based on the half-scaled rating information that varies from 0.5 stars to 5 stars. Statistics of our datasets are shown in Table 2.1.

We run DC 30 times for each experiment, randomly permuting the order of objects and using the median results to minimize the effect of different orderings. We also run 150 iterations of AP alternating $\mathcal{PSD}_{NK}$, $\mathcal{NOR}_N$, and $\mathcal{NN}_N$ in turn. For probabilistic Gibbs sampling, we use the Mallet with the standard option doing 1,000 iterations. All metrics are evaluated against the original $\boldsymbol{C}$, *not against the rectified* $\boldsymbol{C}'$, whereas we use $\boldsymbol{B}$ and A inferred from the rectified $\boldsymbol{C}'$.

[8] `https://archive.ics.uci.edu/ml/datasets/Bag+of+Words`
[9] `http://grouplens.org/datasets/movielens`
[10] `http://www.cs.cornell.edu/~shuochen/lme`

Arora et al. 2013 (Baseline)
neuron layer hidden recognition signal cell noise
neuron layer hidden cell signal representation noise
neuron layer cell hidden signal noise dynamic
neuron layer cell hidden control signal noise
neuron layer hidden cell signal recognition noise
This paper (AP)
neuron circuit cell synaptic signal layer activity
control action dynamic optimal policy controller reinforcement
recognition layer hidden word speech image net
cell field visual direction image motion object orientation
gaussian noise hidden approximation matrix bound examples
Probabilistic LDA (Gibbs)
neuron cell visual signal response field activity
control action policy optimal reinforcement dynamic robot
recognition image object feature word speech features
hidden net layer dynamic neuron recurrent noise
gaussian approximation matrix bound component variables

Table 2.2: Each line is a topic from NIPS ($K = 5$). Previous work simply repeats the most frequent words in the corpus five times.

Qualitative results. Although [9] report comparable results to probabilistic algorithms for LDA, the algorithm fails under many circumstances. The algorithm prefers rare and unusual anchor words that form a poor basis, so topic clusters consist of the same high-frequency terms repeatedly, as shown in the upper third of Table 3. In contrast, our algorithm with AP rectification successfully learns themes similar to the probabilistic algorithm. One can also verify that cluster interactions given in the third panel of Figure 2.4 explain how the five topics correlate with each other.

Similar to [52], we visualize the five anchor words in the co-occurrence space after 2D PCA of $\boldsymbol{C}$. Each panel in Figure 2.1 shows a 2D embedding of the NIPS vocabulary as blue dots and five selected anchor words in red. The first plot shows standard anchor words and the original co-occurrence space. The second plot shows anchor words selected from the rectified space overlaid on

the original co-occurrence space. The third plot shows the same anchor words as the second plot overlaid on the AP-rectified space. The rectified anchor words provide better coverage on both spaces, explaining why we are able to achieve reasonable topics even with $K = 5$.

Rectification also produces better clusters in the non-textual movie dataset. Each cluster is notably more genre-coherent and year-coherent than the clusters from the original algorithm. When $K = 15$, for example, we verify a cluster of *Walt Disney 2D Animations* mostly from the 1990s and a cluster of Fantasy movies represented by *Lord of the Rings* films, similar to clusters found by probabilistic Gibbs sampling. The Baseline algorithm [9] repeats *Pulp Fiction* and *Silence of the Lambs* 15 times.

Quantitative results. We measure the intrinsic quality of inference and summarization with respect to the JSMF objectives as well as the extrinsic quality of resulting topics. Lines correspond to four methods: ○ Baseline for the algorithm in the previous work [9] without any rectification, △ DC for Diagonal Completion, □ AP for Alternating Projection, and ◇ Gibbs for Gibbs sampling.

Anchor objects should form a good basis for the remaining objects. We measure **Recovery** error ($\frac{1}{N}\sum_i^N \|\boldsymbol{C}_i - \sum_k^K p(Z_1 = k|X_1 = i)\boldsymbol{C}_{S_k}\|_2$) with respect to the original $\boldsymbol{C}$ matrix, *not* the rectified matrix. AP reduces error in almost all cases and is more effective than DC. Although we expect error to decrease as we increase the number of clusters K, reducing recovery error for a fixed K by choosing better anchors is extremely difficult: no other subset selection algorithm [19] decreased error by more than 0.001. A good matrix factorization should have small element-wise **Approximation** error ($\|\boldsymbol{C} - \boldsymbol{B}\boldsymbol{A}\boldsymbol{B}^T\|_F$). DC and AP preserve

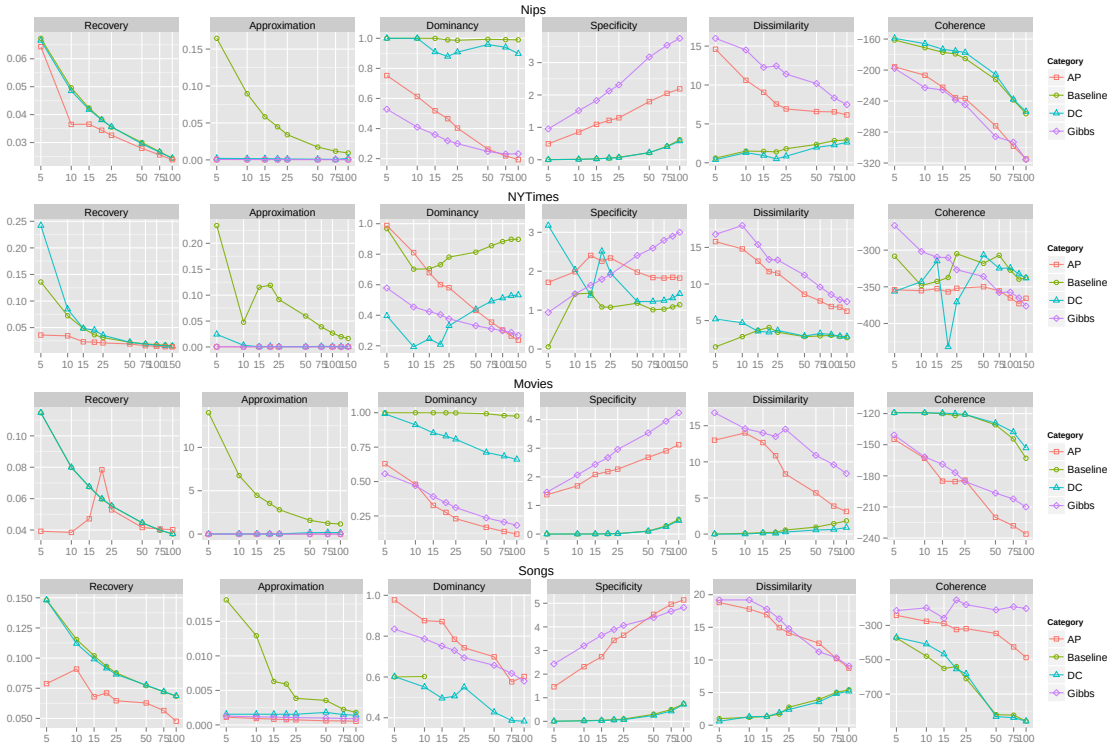

Figure 2.5: Experimental results on real dataset. The x-axis indicates $logK$ where K varies by 5 up to 25 topics and by 25 up to 100 or 150 topics. Whereas the Baseline algorithm largely fails with small K and does not infer quality B and A even with large K, Alternating Projection (AP) not only finds better basis vectors (Recovery), but also shows stable and comparable behaviors to probabilistic inference (Gibbs) in every metric.

more of the information in the original matrix $\boldsymbol{C}$ than the Baseline method, especially when K is small.[11] We expect non-trivial interactions between clusters, even when we do not explicitly model them as in [14]. Greater diagonal **Dominancy** ($\frac{1}{K}\sum_k^K p(Z_2 = k|Z_1 = k)$) indicates lower correlation between clusters.[12] AP and Gibbs results are similar. We do not report held-out probability because we find that relative results are determined by user-defined smoothing parameters

[11] In the NYTimes corpus, 10^{-2} is a large error: each element is around 10^{-9} due to the number of normalized entries.

[12] Dominancy in Songs corpus lacks any Baseline results at $K > 10$ because dominancy is undefined if an algorithm picks a song that occurs at most once in each playlist as a basis object. In this case, the original construction of $\boldsymbol{C}_{SS}$, and hence of A, has a zero diagonal element, making dominancy NaN.

[52, 67].

Specificity ($\frac{1}{K}\sum_k^K KL\left(p(X|Z=k)||p(X)\right)$) measures how much each cluster is distinct from the corpus distribution. When anchors produce a poor basis, the conditional distribution of clusters given objects becomes uniform, making $p(X|Z)$ similar to $p(X)$. Inter-topic **Dissimilarity** counts the average number of objects in each cluster that do not occur in any other cluster's top 20 objects. Our experiments validate that AP and Gibbs yield comparably specific and distinct topics, while Baseline and DC simply repeat the corpus distribution as in Table 3. **Coherence** ($\frac{1}{K}\sum_k^K \sum_{x_1 \neq x_2}^{\epsilon Top_k} \log \frac{D_2(x_1,x_2)+\epsilon}{D_1(x_2)}$) penalizes topics that assign high probability (rank > 20) to words that do not occur together frequently. AP produces results close to Gibbs sampling, and far from the Baseline and DC. While this metric correlates with human evaluation of clusters [63] "worse" coherence can actually be better because the metric does not penalize repetition [52].

In **semi-synthetic experiments** [9] AP matches Gibbs sampling and outperforms the Baseline, but the discrepancies in topic quality metrics are smaller than in the real experiments (see Appendix). We speculate that semi-synthetic data is more "well-behaved" than real data, explaining why issues were not recognized previously.

2.5 Analysis of Algorithm

Why does AP work? Before rectification, diagonals of the empirical $\boldsymbol{C}$ matrix may be far from correct. Bursty objects yield diagonal entries that are too large; extremely rare objects that occur at most once per document yield zero diagonals. Rare objects are problematic in general: the corresponding rows in the $\boldsymbol{C}$

matrix are sparse and noisy, and these rows are likely to be selected by the pivoted QR. Because rare objects are likely to be anchors, the matrix $\boldsymbol{C}_{SS}$ is likely to be highly diagonally dominant, and provides an uninformative picture of topic correlations. These problems are exacerbated when K is small relative to the effective rank of $\boldsymbol{C}$, so that an early choice of a poor anchor precludes a better choice later on; and when the number of documents M is small, in which case the empirical $\boldsymbol{C}$ is relatively sparse and is strongly affected by noise. To mitigate this issue, [67] run exhaustive grid search to find document frequency cutoffs to get informative anchors. As model performance is inconsistent for different cutoffs and search requires cross-validation for each case, it is nearly impossible to find good heuristics for each dataset and number of topics.

Fortunately, a low-rank PSD matrix cannot have too many diagonally-dominant rows, since this violates the low rank property. Nor can it have diagonal entries that are small relative to off-diagonals, since this violates positive semi-definiteness. Because the anchor word assumption implies that nonnegative rank and ordinary rank are the same, the AP algorithm ideally does not remove the information we wish to learn; rather, 1) the low-rank projection in AP suppresses the influence of small numbers of noisy rows associated with rare words which may not be well correlated with the others, and 2) the PSD projection in AP recovers missing information in diagonals. (As illustrated in the Dominancy panel of the Songs corpus in Figure 2.5, AP shows valid dominancies even after $K > 10$ in contrast to the Baseline algorithm.)

Why does AP converge? AP enjoys local linear convergence [54] if 1) the initial $\boldsymbol{C}$ is near the convergence point $\boldsymbol{C}'$, 2) $\mathcal{PSD}_{NK}$ is *super-regular* at $\boldsymbol{C}'$, and 3) *strong regularity* holds at $\boldsymbol{C}'$. For the first condition, recall that we rectified $\boldsymbol{C}'$ by

pushing $\boldsymbol{C}$ toward $\boldsymbol{C}^*$, which is the ideal convergence point inside the intersection. Since $\boldsymbol{C} \to C^*$ as shown in (5), $\boldsymbol{C}$ is close to $\boldsymbol{C}'$ as desired.The prox-regular sets[13] are subsets of super-regular sets, so prox-regularity of $\mathcal{PSD}_{NK}$ at $\boldsymbol{C}'$ is sufficient for the second condition. For permutation invariant $\mathcal{M} \subset \mathbb{R}^N$, the spectral set of symmetric matrices is defined as $\lambda^{-1}(\mathcal{M}) = \{X \in \mathcal{S}_N : (\lambda_1(X), \ldots, \lambda_N(X)) \in \mathcal{M}\}$, and $\lambda^{-1}(\mathcal{M})$ is prox-regular if and only if $\mathcal{M}$ is prox-regular [25, Th. 2.4]. Let $\mathcal{M}$ be $\{x \in \mathbb{R}^+_n : |supp(x)| = K\}$. Since each element in $\mathcal{M}$ has exactly K positive components and all others are zero, $\lambda^{-1}(\mathcal{M}) = \mathcal{PSD}_{NK}$. By the definition of $\mathcal{M}$ and $K < N$, $P_{\mathcal{M}}$ is locally unique almost everywhere, satisfying the second condition almost surely. (As the intersection of the convex set $\mathcal{PSD}_N$ and the smooth manifold of rank K matrices, $\mathcal{PSD}_{NK}$ is a smooth manifold almost everywhere.)

Checking the third condition a priori is challenging, but we expect noise in the empirical $\boldsymbol{C}$ to prevent an irregular solution, following the argument of Numerical Example 9 in [54]. We expect AP to converge locally linearly and we can verify local convergence of AP in practice. Empirically, the ratio of average distances between two iterations are always ≤ 0.9794 on the NYTimes dataset (see the Appendix), and other datasets were similar. Note again that our rectified $\boldsymbol{C}'$ is a result of pushing the empirical $\boldsymbol{C}$ toward the ideal $\boldsymbol{C}^*$. Because approximation factors of [9] are all computed based on how far $\boldsymbol{C}$ and its co-occurrence shape could be distant from $\boldsymbol{C}^*$'s, all provable guarantees of [9] hold better with our rectified $\boldsymbol{C}'$.

[13] A set $\mathcal{M}$ is prox-regular if $P_{\mathcal{M}}$ is locally unique.

2.6 Related and Future Work

JSMF is a specific structure-preserving Non-negative Matrix Factorization (NMF) performing spectral inference. [77, 47] exploit a similar separable structure for NMF problmes. To tackle hyperspectral unmixing problems, [66, 33] assume *pure pixels*, a separability-equivalent in computer vision. In more general NMF without such structures, RESCAL [68] studies tensorial extension of similar factorization and SymNMF [45] infers $\boldsymbol{BB}^T$ rather than $\boldsymbol{BAB}^T$. For topic modeling, [3] performs spectral inference on third moment tensor assuming topics are uncorrelated.

As the core of our algorithm is to rectify the input co-occurrence matrix, it can be combined with several recent developments. [67] proposes two regularization methods for recovering better $\boldsymbol{B}$. [52] nonlinearly projects co-occurrence to low-dimensional space via t-SNE and achieves better anchors by finding the exact anchors in that space. [85] performs multiple random projections to low-dimensional spaces and recovers approximate anchors efficiently by divide-and-conquer strategy. In addition, our work also opens several promising research directions. How exactly do anchors found in the rectified $\boldsymbol{C}'$ form better bases than ones found in the original space $\boldsymbol{C}$? Since now the topic-topic matrix $\boldsymbol{A}$ is again doubly non-negative and joint-stochastic, can we learn super-topics in a multi-layered hierarchical model by recursively applying JSMF to topic-topic co-occurrence $\boldsymbol{A}$?

CHAPTER 3

APPLICATIONS AND COMPARISON WITH OTHER RELATED MODELS

Spectral topic modeling transforms learning a low-dimensional latent geometry into a provable decomposition of co-occurrence statistics. Despite their theoretical guarantees and vast scalability, spectral topic models are not widely used due to the absence of reliability in real data. Matrix models and tensor models often complain the less realistic assumption of each other without thoroughly investigating their topic quality against the probabilistic counterparts. Parameter sensitivity of the matrix models and learning cost of the tensor models are another main barriers that hinder the fair comparison between them. This paper is the first work that provides unifying explanations of the two popular approaches, measuring their real performance on various metrics. Proposing robust and complete algorithms for the anchor-based topic inference, we then demonstrate the versatile power of the matrix models in learning from correlated to hierarchical topics within a simple framework.

3.1 Introduction

Increasing access to massive data streams can be strategic asset to both theoreticians and practitioners, but only if they are capable of extracting meaningful patterns. Topic models learn low-dimensional hidden structures in arbitrary type of data that involves groups of discrete observations [39, 15], thereby being flexibly applicable to a wide range of modalities without human annotations. Users can find common themes that underlie text articles [34], expressive features or segments that characterize image streams [82], hidden prefer-

ences/genres on movie/music consumption [51], and latent communities from network snapshots [57]. For clarity this paper sticks to using the standard terms — words, documents, and topics — but the concepts generalize to various applications beyond these examples.

Standard probabilistic algorithms for topic modeling lack of scalability. To learn quality topics, the likelihood-based inference such as Variational Bayes (VB) or Markov Chain Monte Carlo (MCMC) needs to iterate through the training data multiple times until parameter convergence, being hardly scaling to millions and billions of documents.[1] To learn correlations or hierarchies of topics, more complex models are necessary with the expansive inference [14, 22, 56, 62], setting the Latent Dirichlet Allocation [15] still as the default tool for practitioners. Spectral algorithms are newer alternatives to likelihood-based training. Since topics are *frequently co-occurring terms* in essence, these algorithms explicitly construct word co-occurrence moments as statistically unbiased estimators for the underlying generative process via a trivially parallelizable single-pass iteration. Then users can infer latent topics via moment-matching without revisiting the individual training documents.

Using the method of moments provides provable guarantees but becomes susceptible to statistical noise. To learn the latent topics, matrix decomposition algorithms [8, 9, 51, 12, 42] factorize the second-order co-occurrence between pairs of words, matching its posterior moments. Tensor decomposition algorithms [3, 5, 4, 6] factorize the third-order co-occurrence among triples of words, matching its population moments. Whereas these algorithms do not suffer from spurious local minima or slow mixing problems of VB/MCMC, the learning

[1] Leveraging the stochasticity like [38, 61] is not the major focus of this paper because the same approach is also applicable to spectral algorithms.

quality quickly degrades if the input data does not agree well with the underlying models. These spectral inference do not handle the **model-data mismatch** as well as the likelihood-based inference [46, 59], being less useful in real data.

The practicality of topic learning also matters model assumptions and learning complexity. The Anchor Words algorithms [8, 9, 51], the most popular matrix-based approach, assume **separable topics**[2], whereas the CP-decomposition [3, 5], the most popular tensor-based approach, assume **orthogonal topics**.[3] Most topic models with large vocabulary are shown separable [27], but the vocabulary is often tailored down for manageable inference. Topics are rarely orthogonal unless we learn tiny topic models on the distinctive sets of documents without sharing much of their vocabulary. While each approach has pointed out the assumption of the other approach as a major weakness, no thorough comparison has been made especially on real data. For the Anchor Words algorithms, this is because the exponentiated gradient topic recovery requires fuzzy tuning of learning rate and heterogeneous document frequency cut-offs. For the tensor algorithms, even the simplest CP-decomposition takes too long time in learning beyond the small topic models.

This paper provides unifying explanations for spectral topic inference. Measuring the topic quality of matrix-based and tensor-based approaches against the probabilistic inference on various metrics, we show that the Rectified Anchor Words (RAW) algorithm [51] substantially outperforms the CP-decomposition, better handling the model-data mismatch within a few orders of magnitude smaller times. Revisiting the framework of the **Joint Stochastic Matrix Factorization (JSMF)** [51], we propose the Robust RAW (RRAW) algo-

[2]Each topic has one specific *anchor word* that occurs only in the context of that topic.
[3]Topics which are probability vectors must be perpendicular, being uncorrelated.

rithm that is complete and free from the intricate control of the model parameters. Our new algorithm is based on Alternating Direction Method of Multipliers (ADMM) with Douglas-Rachford splitting (DR), and further improves the topic quality and sparsity from the previous work [51] without adding a complex regularizer like [67]. We verify that the RRAW algorithm discover superior topics than its probabilistic counterpart: the Correlated Topic Models (CTM) [14], better explaining genre-topic associations at smaller costs. By maximally reusing the learned topic correlations, we further propose a novel approach for hierarchical topic modeling that can learn supertopics within the same simple framework.

3.2 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 prepared topics, all topic 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.[4] 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 ϱ. 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 ϱ; 2) Write n_m words by repeatedly selecting a topic z from the composition $\boldsymbol{w}_m$ and a word x from the

[4] K is considerably smaller than N in the general settings. The setting with $K > N$ is called *overcomplete*, which requires additional assumptions for identifiability.

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 $\mathfrak{f} = \mathrm{Dir}(\boldsymbol{\alpha})$ for $\boldsymbol{\alpha} \in \mathbb{R}_+^K$ [15]. In the correlated topic model (CTM) $\mathfrak{f} = \mathcal{LN}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ for $\boldsymbol{\mu} \in \mathbb{R}^{K-1}, \boldsymbol{\Sigma} \in \mathbb{R}^{(K-1)\times(K-1)}$ [14, 22]. In the Pachinko allocation model $\mathfrak{f}$ is not a parametric family, but a DAG-induced distribution, which is not always uniquely identifiable [56, 62]. 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 $\mathfrak{f}$ [11], and can be similarly incorporated in spectral inference by putting additional regularizers when recovering each $\boldsymbol{b}_k$ [67].

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 like $\mathfrak{f} = \mathrm{Dir}(\boldsymbol{\alpha})$, one can recover the hyperparameter $\boldsymbol{\alpha}$ [8].[5]

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 $\frac{1}{2}\|\widetilde{\boldsymbol{H}} - \boldsymbol{BW}\|_F^2$ with the column-stochastic constraints $\boldsymbol{B} \in^{N\times K}, \boldsymbol{W} \in^{K\times M}$. While this factorization could be

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

identifiable under some additional sparsity constraints [41], solving it by the NMF methods like [50] produces incoherent topics even if the approximation error is small enough [73]. This is essentially because $\boldsymbol{H}$ itself is too noisy and sparse statistics where only a tiny subset of vocabulary appears for each document.

3.2.1 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 $p_{x_1 x_2}(i, j)$ 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 topic modeling objective into a second-order non-negative matrix factorization: $\boldsymbol{C} \approx \boldsymbol{B}\boldsymbol{A}\boldsymbol{B}^T$, which is equivalent to $p(x_1, x_2|A; B) = \sum_{z_1} \sum_{z_2} p(x_1|z_1; B) p(z_1, z_2|A) p(x_2|z_2; B)$. 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

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

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

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

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

$$\boldsymbol{C}_m = \frac{\boldsymbol{h}_m\boldsymbol{h}_m^T - diag(\boldsymbol{h}_m)}{n_m(n_m-1)}. \tag{3.1}$$

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 the 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 [8]. 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 constructing the empirical moment $\boldsymbol{C}$ from the input data as an unbiased estimator of the underlying generative process, JSMF enables us to recover the correct $\boldsymbol{B}$ and $\boldsymbol{A}$ up to some precision by matching $\boldsymbol{C}$ to its posterior moments $\boldsymbol{B}\boldsymbol{A}^*\boldsymbol{B}^T$. 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 $\mathfrak{f}(\boldsymbol{\alpha})$ rahter than performing an inference [8]. The **separability assumption** implies *non-negative rank*($\boldsymbol{B}$) = *rank*($\boldsymbol{B}$) = K, guaranteeing the existence of an identifiable factorization.

3.2.2 Tensor Decomposition

The separability assumption in JSMF is necessary because having only up to the second moments is not sufficient by itself to identify latent topics [4]. While one could release this assumption by adopting *sufficiently scattered* condition, it maps the factorization into another NP-hard optimization problem [42]. Alternatively, one can leverage third-order moments to provide sufficient statistics for identifiable topic inference [3, 5]. In contrast to JSMF, tensor-based algorithms first specify $\mathfrak{f}$ as a tractable parametric prior like the Dirichlet distribution. For example, if $\mathfrak{f}$ = 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

$$\mathbb{E}^{(2nd)}_{\boldsymbol{w}\sim\mathfrak{f}(\boldsymbol{\alpha})}[w_k w_l] = \begin{cases} \frac{\alpha_k(\alpha_k+1)}{\alpha_0(\alpha_0+1)} & (k=l) \\ \frac{\alpha_k\alpha_l}{\alpha_0(\alpha_0+1)} & (k\neq l) \end{cases}. \tag{3.2}$$

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

by $\boldsymbol{\alpha}$.

$$\mathbb{E}[\boldsymbol{x}_1] = \mathbb{E}_{\boldsymbol{w}_m}[\mathbb{E}[\boldsymbol{x}_1|\boldsymbol{w}_m]] = \boldsymbol{B}\mathbb{E}_{\boldsymbol{w}_m}[\boldsymbol{w}_m] = \boldsymbol{B}\mathbb{E}^{(1st)}_{\boldsymbol{w}\sim\text{Dir}(\boldsymbol{\alpha})}[\boldsymbol{w}],$$

$$\begin{aligned}\mathbb{E}[\boldsymbol{x}_1\boldsymbol{x}_2^T] &= \mathbb{E}_{\boldsymbol{w}_m}[\mathbb{E}[\boldsymbol{x}_1|\boldsymbol{w}_m] \cdot \mathbb{E}[\boldsymbol{x}_2|\boldsymbol{w}_m]^T] \\ &= \boldsymbol{B}\mathbb{E}_{\boldsymbol{w}_m}[\boldsymbol{w}_m\boldsymbol{w}_m^T]\boldsymbol{B}^T = \boldsymbol{B}\mathbb{E}^{(2nd)}_{\boldsymbol{w}\sim\text{Dir}(\boldsymbol{\alpha})}[\boldsymbol{w}\boldsymbol{w}^T]\boldsymbol{B}^T.\end{aligned}$$

Similarly we can represent up to the third moments:

$$\mathbb{E}[\boldsymbol{x}_1 \otimes \boldsymbol{x}_2] = \mathbb{E}^{(2nd)}_{\boldsymbol{w}\sim\dagger(\boldsymbol{\alpha})}[\boldsymbol{w} \otimes \boldsymbol{w}](\boldsymbol{B}, \boldsymbol{B}),$$

$$\mathbb{E}[\boldsymbol{x}_1 \otimes \boldsymbol{x}_2 \otimes \boldsymbol{x}_3] = \mathbb{E}^{(3rd)}_{\boldsymbol{w}\sim\dagger(\boldsymbol{\alpha})}[\boldsymbol{w}^{\otimes 3}](\boldsymbol{B}, \boldsymbol{B}, \boldsymbol{B}).$$

By assuming $\boldsymbol{w} \sim \text{Dir}(\boldsymbol{\alpha})$, we can fortunately attain a 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 [3]. Thus once users input α_0, we can construct the empirical moments given the training data, and then tensor decomposition allows us to recover $\boldsymbol{B}$ and $\boldsymbol{\alpha}$ up to some precision by matching the empirical moments to these population moments. Note that JSMF does not ask users to specify α_0, flexibly and transparently modeling arbitrary pairwise correlations between topics by the co-occurrence between the pairs of the anchor words.

There are several caveats. First, finding such closed-form moment combinations is not easy. Normally all higher-order moments are necessary for learning with the general prior $\dagger$ [6]. Second, $\mathbb{E}[\boldsymbol{w}^{\otimes 3}]$ should be a diagonal tensor in order to apply popular CP-decomposition for learning topics $\boldsymbol{B}$. It means that we need to assume the **uncorrelated topics** instead of the separable topics, though one might later capture weak negative correlations via learned $\boldsymbol{\alpha}$ given $\dagger$ = Dir. Using Tucker decomposition [78] is another option for learning correlated topics,

but it instead requires additional sparsity constraints on $\boldsymbol{B}$, asking notably more parameters to be estimated [4]. Overall, correlated topic modeling via tensor decomposition is not as transparent as using JSMF.

3.3 The Robust 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 [51] is the first working version that can learn quality topics and their correlations under model-data mismatch. However it requires intricate and heterogeneous tuning of model parameters. We propose the Robust RAW (RRAW) algorithm based on ADMM-DR. This is the first complete version comparable to full probabilistic inference. Thanks 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 the topic correlations $\boldsymbol{A}$ and hyperparameter $\boldsymbol{\alpha}$ if available.

Step 0: Create $\boldsymbol{C}$. For spectral inference, we need to first construct the empirical word co-occurrence statistics as an unbiased estimator for the underlying random process: $\boldsymbol{C} = (1/M)\sum_{m=1}^{M} \boldsymbol{C}_m$ with $\boldsymbol{C}_m$ specified in Equation (3.1). Due to the efficiency of the anchor-based inference, the moment construction often

Algorithm 1 Alternate Projection (AP)

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

$\boldsymbol{C}_{\mathcal{NN}} \leftarrow \boldsymbol{C}$
repeat
  $(\boldsymbol{U}, \boldsymbol{\Lambda}_K) =$ 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 an operation that maps the input vector into the diagonal matrix or extracts the diagonal vector from the input matrix.)

becomes the most expansive step for large corpora, but it is trivially parallelizable for each document because the averaging at the end is the only between-documents computation.

Step 1: Rectify $\boldsymbol{C}$. The typical failure mode of low-rank spectral learning is mimatch between the model and the data. Thus rectifying the co-occurrence estimator is the key to successful inference [51]. Though $\boldsymbol{C}$ is shown statistically more stable than $\widetilde{\boldsymbol{H}}$ [8], its empirical construction does unlikely exhibit the proper structures of the posterior moments $\boldsymbol{B}\boldsymbol{A}^*\boldsymbol{B}^T$: a low-rank ($\mathcal{LR}$), 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 space until convergence [51].

By running the truncated eigenvalue decomposition, it only finds K largest eigenvalues $\boldsymbol{\Lambda}_K$ with the corresponding eigenvectors $\boldsymbol{U}$ at tiny cost. Then it projects $\boldsymbol{C}$ to $\mathcal{PSD}_N$ and $\mathcal{LR}_K$ spaces by the reconstruction $\boldsymbol{U}\boldsymbol{\Lambda}_K^+\boldsymbol{U}^T$. The next

[6] Due to the diagonal penalty in (3.1) for the unbiased construction and the variance of the generative process, $\boldsymbol{C}$ is almost always full-rank and indefinite in limited real data.

Algorithm 2 Sparse Implicit Column-pivoted QR

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

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

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

is an orthogonal projection to $\mathcal{NOR}_N$ by subtracting mean overage entry-wisely from the desired total, 1.0. The negative entries from this procedure are 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.[7] Note that tensor-based methods similarly have a **whitening step** for handling the model-data mismatch. By running a full SVD, they transform the third-order moments into an orthogonal tensor for CP-decomposition.

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

[7] After convergence, we normalize the co-occurrence via dividing by the entry sum just for consistent comparison.

by a convex combination: $\sum_k p(z=k|x=i)=1$ of the rows $\{\overline{\boldsymbol{C}}_{s_k}\}$ corresponding to the anchor words $\{s_k\}$. Therefore the inference quality depends primarily on the quality of the anchor words S, providing a clear metric for diagnosis. Since the rectification is proven crucial for finding better anchors [51], it again articulates the importance of the rectification step.

While using the pivoted QR [9] substantially expedites the running time against solving a number of LPs [8], it cannot maintain the sparsity of $\overline{\boldsymbol{C}}$ because it explicitly projects every non-anchor row to the orthogonal complement for each iteration. Random projections are suggested for sizable vocabulary, but such projections can no longer maintain the insisted structures of the rectified $\boldsymbol{C}$ and likely degrade the topic quality [52]. Our 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 $\overline{\boldsymbol{C}}$.

Step 3: Recover $\boldsymbol{B}$. Provided with the set of the anchor words S and the convex coefficients $\breve{\boldsymbol{B}}_{ki}=\{p(z=k|x=i)\}$, one can easily recover $\boldsymbol{B}$ by applying the Bayes rule:

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

where $c_i := p(x=i)$ is the unigram probability for the word i, which can be acquired by $\sum_j \boldsymbol{C}_{ij}$. Hence the core of this step is to find the coefficient matrix $\boldsymbol{B}'$ by solving multiple Simplex-constrained Non-negative Least Squares (SNLS) hat satisfies $\overline{\boldsymbol{C}}_{ij}=\sum_k \breve{\boldsymbol{B}}_{ki}\overline{\boldsymbol{C}}_{s_k,j}$ for each i. While the exponentiated gradient (ExpGrad) algorithm used in the previous work [9, 51] quickly converges, tuning the learning rate is mysterious, less ensuring the confidence of the results. Instead, we propose another algorithm that uses Alternating Direction Method of Multipliers (ADMM).

Algorithm 3 ADMM by Douglas-Rachford (DR)

def RECOVER-B($\overline{\boldsymbol{C}}, \boldsymbol{c}, S, \lambda, \gamma$)

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

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 $\boldsymbol{F} = (\gamma \boldsymbol{U}^T\boldsymbol{U} + \boldsymbol{I}_K)^{-1}$ just once and the second invariant $\boldsymbol{f} = \gamma \boldsymbol{U}^T\boldsymbol{v}$ only $N - K$ times for different $\boldsymbol{v}$'s.[8]

[8]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 $\mathrm{g}(\boldsymbol{\beta})$ [67].

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

(Set indexing extracts a principle submatrix whose rows and columns correspond to the arguments. The † operation means entrywise scalar inverse.)

Step 4: Recover A and α. 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, $p(x_1 = s_k, x_2 = s_l) = \sum_{l'}\left(\sum_{k'} p(x_1 = s_k|z_1 = k')p(z_1 = k', z_2 = l')\right)p(x_2 = s_l|z_2 = l') = p(x_1 = s_k|z_1 = k)\sum_{l'} p(z_1 = k, z_2 = l')p(x_2 = s_l|z_2 = l') = p(x_1 = s_k|z_1 = k)p(z_1 = k, z_2 = l)p(x_2 = s_l|z_2 = l)$. 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. Thus the co-occurrence of the 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. JSMF effectively balances these entries via the rectification

[51], thereby realizing **correlated topic modeling**.

Suppose that the recovered $\boldsymbol{A}$ is close to the population second moments (3.2) of Dir($\boldsymbol{\alpha}$). Then its row sum vector $\boldsymbol{a}$ becomes $\boldsymbol{\alpha}/\alpha_0$, the first moments of 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 (3.2) 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 [64].[9]

3.4 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 [16, 56, 62]. As the correlations $\boldsymbol{A}$ learned by the RRAW algorithm have the same positive semidefinite and joint-stochastic structures as

[9]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.

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 RRAW algorithm on the recovered $\boldsymbol{A}$, learning supertopics of the current topics.

Suppose that the initial run of the RRAW 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.

3.5 Experimental Results

We evaluate our algorithms on two standard textual datasets: NIPS full papers (NIPS) and New York Times news articles (NYTimes). We also prepare two

small textual datasets: political blogs (Blog) [29] and business reviews (Yelp) [52] especially for tensor decomposition. In addition, we adopt two non-textual preference datasets: Movielens 10m reviews (Movies) and Yes.com complete playlists (Songs).[10] In contrast to textual datasets, we can retrieve genre information for Movies and Songs.[11] We process training documents identically to [51] for fair comparison. Basic statistics of each dataset are available in Figure 3.1 and 3.2.

3.5.1 Quantitative Analysis

After constructing $\boldsymbol{C}$ (Step 0), Baseline method [9] jumps to the anchor-finding (Step 2) to demonstrate the power of the rectification. However we do not use any random projection or pseudo-inverse recovery of $\boldsymbol{A}$ given in [9] 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).[12] Since our new anchor-finding (Step 2) only improves time/space complexity, topic learning (Step 3) contrasts our work from the previous work [51].

For the exponentiated gradient (ExpGrad), we set the learning rate and the document frequency cut-offs as the best known values from [51]. 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-

[10]Movies: `https://grouplens.org/datasets/movielens/10m/` and Songs: `http://csinpi.github.io/lme/data_page.html`

[11]We maximally use the existing genre information and user tags given in the datasets. If no information is provided, we scrape from IMDB and Discogs.com, respectively.

[12]We verify that running only 5 iterations of AP is sufficiently practical, and 15 iterations makes closer to the current results.

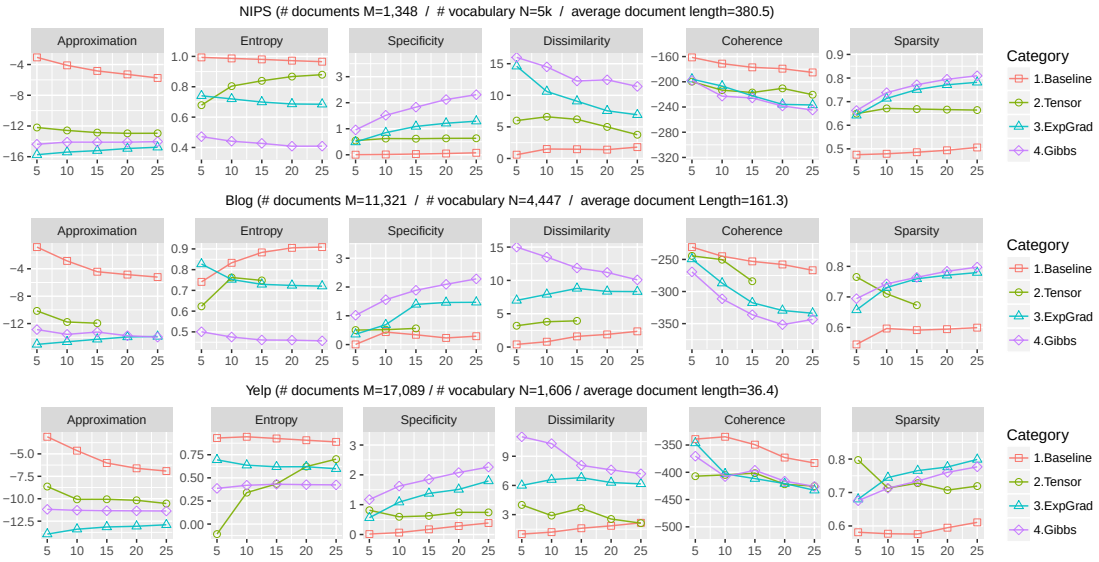

Figure 3.1: Matrix vs Tensor. Tensor algorithm performs better than Baseline Anchor Word algorithm [9], but much poorer than the Rectified Anchor Word algorithm: ExpGrad [51] and Gibbs. Surprisingly, tensor algorithm does not show consistent behavior for increasing numbers of topics in X-axis. Close to Gibbs is generally better in Y-axis.

based inference, we identically use Gibbs Sampling (MCMC) with 1,000 iterations after the initial 200 burn-in samples. We run the CP-decomposition for Tensor algorithm.[13]

Following the metrics on [51], 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 better compare ADMM-DR against ExpGrad. We also add two new metrics: Entropy ($\frac{1}{N}\sum_i \frac{H(z|x=i)}{\log_2 K}$) [52] and Sparsity ($\frac{1}{K}\sum_k \frac{\sqrt{N}-(\|\boldsymbol{b}_k\|_1/\|\boldsymbol{b}_k\|_2)}{\sqrt{N}-1}$) [40]. As given in Figure 3.1, only Entropy in Tensor algorithm grows as K increases, unusually saying that topic distribution given a words becomes closer to the uniform distribution. For sparsity closer to 1.0 is better. Sparsity of Tensor algorithm fluctuates as well, questioning the consistency of its spectral aspect. Specificity ($\frac{1}{K}\sum_k KL(p(x|z=k)\|p(x))$) measures the average KL-distance of each topic from

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

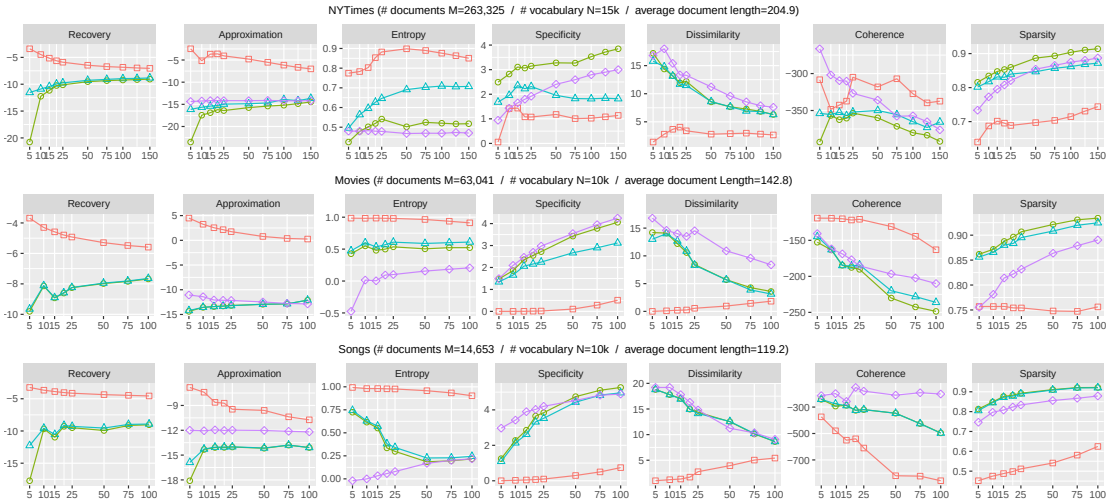

Figure 3.2: ADMM-DR vs ExpGrad. Our ADMM-DR algorithm outperforms the previous state-of-the-art rectified algorithm ExpGrad, being more comparable to probabilistic Gibbs sampling. Baseline algorithm without rectification works consistently poor due to model-data mismatch. Panel 1, 2, 3: lower is better / 4, 5, 7: higher is better / 6: closer to Gibbs is better.

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. But Coherence could be deceptive if a model learns many duplicated topics containing the frequent words [42]. In every metric, being closer to Gibbs is generally better, implying that the matrix model (ExpGrad) outperforms the tensor model.

Figure 3.2 shows that the Baseline method works notably worse than other methods and is far behind the trend of Gibbs sampling, reconfirming [51]. ADMM-DR generally agree well with ExpGrad on many metrics, but ADMM-DR produces more specific and sparse topics with the lower entropies without requiring mysterious tuning of the learning rate and the document frequency cut-offs. ADMM-DR also improves inference quality by decreasing Recovery

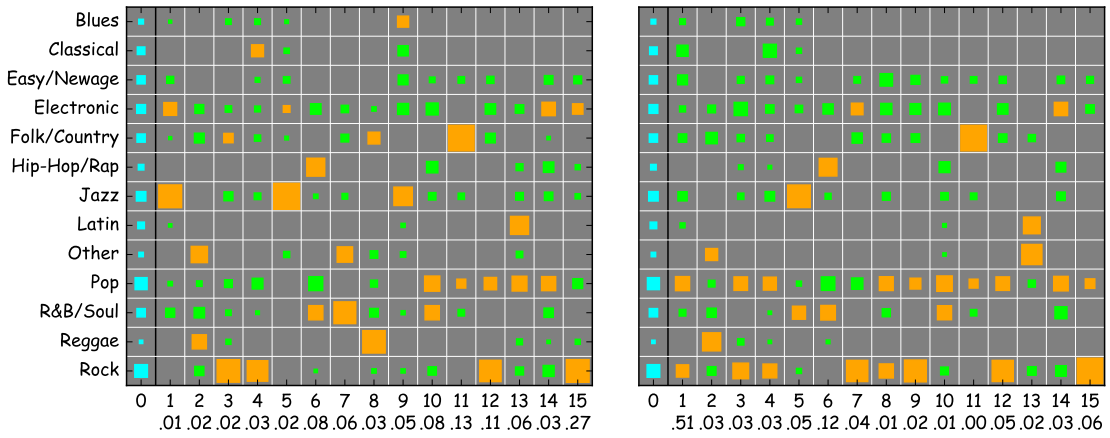

Figure 3.3: ADMM-DR (left) vs CTM (right). The column 0 shows the genre distribution of the entire corpus. Each column 1-15 stands for k-th topic where two most prominent genres are of orange colors. The size of each box is proportional to the relative intensity. The number below each topic indicates the marginal probability $p_z(k)$. ADMM-DR topics capture more about genres.

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.

3.5.2 Qualitative Analysis

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 [14], which uses logistic-normal priors with pairwise covariance between topics. When running Varational CTM [14] with the default parameters, the

resulting topics do not show distinguishable genre associations. Most topics involve with Pop and Rock, emulating the overall genre distribution of the corpus as illustrated in Figure 3.3. In contrast, our ADMM-DR 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 ADMM-DR's associate more with Folk/Country or Pop, respectively.

In addition, CTM puts spuriously high marginal topic probability on T1, which is the closest topic to the corpus genre distribution. While it can highly contribute to maximizing the data likelihood, an unseen playlist would be most likely classified as a mixture of Pop and Rock if it contains just a couple of Pop or Rock songs. This also happens in Movies, explaining why we prefer using various metrics than merely showing the held-out likelihood. Genre association in Movies is less clear than Songs because each playlist more likely has genre-specific themes, whereas people often watch and review newly released movies rather than consuming only similar genres. Thus Movies consist of year-specific topics as well as Fantasy or Sci-Fi.

3.5.3 Hierarchy and Further Analysis

For hierarchy analysis, we compare across three different settings with our ADMM-DR in the NIPS dataset: 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 3.1 shows the most prominent 7

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

Table 3.1: Top 7 words for each of five topics by three models.

words out of top 20 words for each topic similar to [51]. 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.

When we run the variational CTM-5 again with the default parameters [14], the resulting topics do not have distinguishable genre associations as illustrated in Figure 3.4. This failure may be the result of spurious correlations as pointed out in [69]. 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.

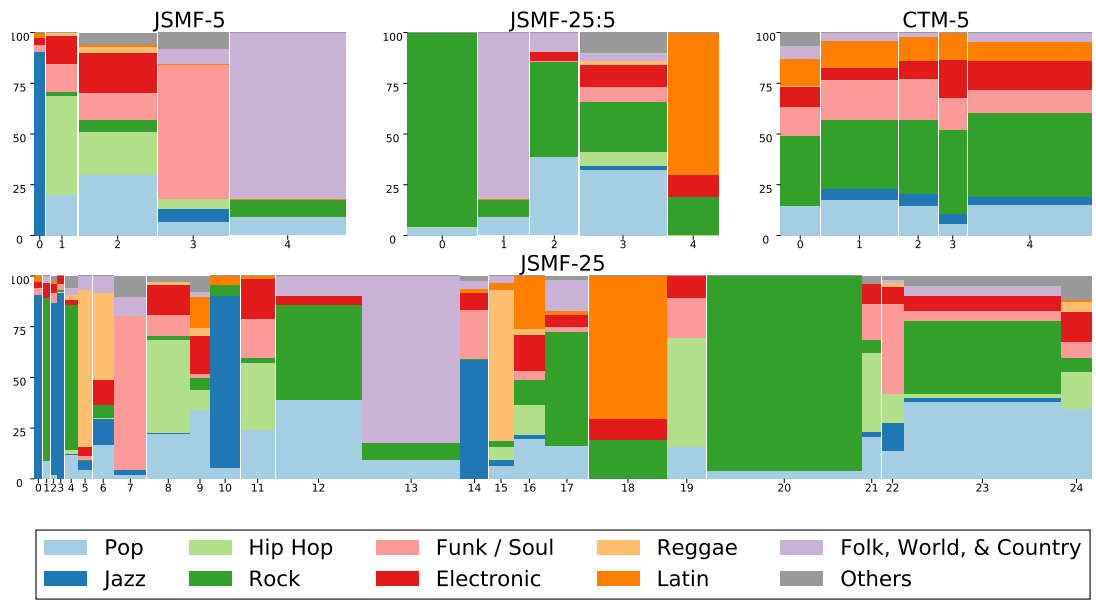

Figure 3.4: **Second row.** 25 subtopics on Songs dataset. Given 20 top songs of each topic, the stacked bar chart indicates the percentages of the most popular 9 genres. The width of each topic is proportional to the marginal likelihood of the topic $p(z = k) = \sum_l A_{kl}$. **First row.** The leftmost and the rightmost panels show 5 topics from independent running of the JSMF with the ADMM-DR and the CTM, respectively. The middle panel represents 5 supertopics by recursive running of the same JSMF on top of 25 subtopics given in the second row.

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.

3.6 Discussion

Tensor algorithm takes 5 hours for learning 5 topics on NIPS and 48 days for learning 25 topics on Yelp, having not yet finished learning 20-25 topics on Blog during 4 months. In the same computing environment, the Robust RAW algorithm takes only a few seconds in these toy datasets and at most a couple hours for processing the largest NYTimes. The CTM takes 15 mins for learning 15 topics on Songs, but 6 hours for 50 topics. In contrast, our ADMM-DR takes less

than 10 mins for finding 50 topics on Songs.

By removing the dependence on the training documents, spectral topic modeling provides great scalability in finding compact high-level structures in sparse and discrete data such as text and user-preference. Our Roubst RAW algorithm enjoys its transparent and consistent behaviors, working well on various types textual and non-textual real datasets without asking intricate tuning of model parameters. The experimental results show that the matrix-based inference notably outperforms the tensor-based inference in both topic quality and learning complexity. Our ADMM-DR algorithm further improves the previous ExpGrad [51], being more comparable to probabilistic Gibbs sampling without increasing the model complexity. The anchor-based inference has high potential for better modeling arbitrary pairwise topic correlations at the lower cost than CTM and additional flexibility to model the hierarchical topics within the one unified framework of the Joint Stochastic Matrix Factorization.

CHAPTER 4

PRIOR-AWARE DOCUMENT-SPECIFIC TOPIC INFERENCE

Spectral topic modeling algorithms operate on matrices/tensors of word co-occurrence statistics to learn topic-specific word distributions. This approach removes the dependence on the original documents and produces substantial gains in efficiency and provable inference, but at a cost: the model can no longer produce information about individual documents. We introduce a novel Prior-aware Dual Decomposition (PADD) method that estimates document-topic distributions using topic correlations. Experiments on several synthetic and real collections demonstrate that PADD outperforms a variety of baseline methods because of its better handling of correlated topics.

4.1 Introduction

Unsupervised topic modeling is a foundation of contemporary machine learning. It transforms a collection of documents into two matrices that represent **topics**, which are distributions over words, and document **compositions**, which are distributions over topics [39, 15]. Interpretable topics allow users to quickly assess the main themes, common genres, and underlying communities in various types of data [51, 57], while document-topic compositions enable users to retrieve documents that are representative of query topics or to measure connections between topics and metadata like time variables. [14, 75, 36, 76, 32, 31].

Spectral topic models have emerged as an alternative to likelihood-based inference such as Variational Bayes [15] or Gibbs Sampling [34], which provides

provable optimality and transparent, deterministic inference. Anchor Word algorithms [8, 9, 12, 51, 42] factorize the second-order co-occurrence matrix between pairs of words to recover the matrix of topic-word distributions. Higher-order algorithms [3, 5, 4, 83] factorize a third-order tensor of word triples. Because the input to these algorithms is purely in terms of word-word relationships, we can limit our interaction with the training documents to a single trivially-parallelizable pre-processing step to construct the co-occurrence statistics. We can then learn topic models of various sizes without revisiting the training documents.

But the efficiency advantage of factoring out the documents is also a weakness: we lose the ability to say anything about the documents themselves. In practice, users of spectral topic models must go back and apply traditional inference on the original training documents as if these were new, held-out documents. That is, given topic-word distributions and a sequence of words for each document, they need to estimate the posterior probability of topics, with the assumption of a sparse Dirichlet prior or a more complex logistic-normal prior [14] on the topic composition. Estimating topics with a sparse Dirichlet prior can be NP-Hard even for trivial models [72]. Gibbs Sampling for topic inference is asymptotically unbiased, but has no provable guarantees and may require large numbers of samples for high-dimensional models [84]. Variational Bayes often becomes trapped in local minima, learning inconsistent models for various numbers of topics.

To learn the document-specific topic distributions for spectral topic models, [10] recently propose the Thresholded Linear Inverse (TLI) method. Since the original document's word-count vector could be modeled as the product of

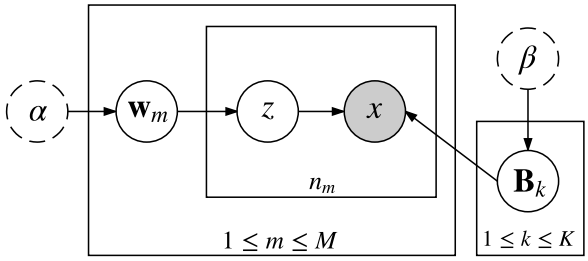

Figure 4.1: LDA asserts a topic composition $\boldsymbol{w}_m$ for each document m. Dir($\boldsymbol{\alpha}$) provides prior information for the entire corpus.

the words-by-topics matrix and the document's topic composition vector, TLI multiplies the word count vector by the inverse of the words-by-topics matrix and removes entries below a threshold, reconstructing the sparse topic composition vector. Unfortunately this non-square matrix inversion is expensive for large vocabularies and numerically unstable, often producing NaN entries and thereby learning compositions inferior to likelihood-based inference. Even though TLI has provable guarantees, its thresholding scheme quickly loses both precision and recall as the number of topics increases. More fundamentally, this method only uses the matrix of topic-word probabilities, and makes no use of prior information about topic correlations. In practice, topics are often strongly correlated: biology occurs with chemistry more often than with economics.

In this work we propose two new methods. The first, **Simple Probabilistic Inverse (SPI)**, renormalizes the topic-word distributions into conditional distributions given words. This simple baseline can outperform TLI when topics have no meaningful correlation structure. The second, **Prior-aware Dual Decomposition (PADD)** is capable of learning quality document-specific topic compositions by leveraging the learned joint distribution over pairs of topics as a prior. PADD regularizes topic correlations of each document to be not too far from the overall topic correlations, thereby guessing reasonable compositions

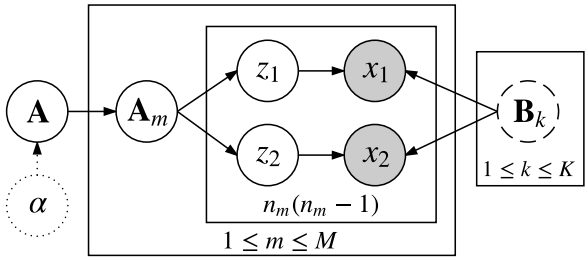

Figure 4.2: JSMF asserts a *joint* distribution $\boldsymbol{A}_m$ over topic pairs for each document m. $\boldsymbol{A}$ serves as a prior for the entire corpus.

even for short documents. Because PADD requires high-quality estimates of topic correlation, it has only become feasible for spectral topic models with the recent introduction of the rectified anchor words algorithm within the framework of **Joint Stochastic Matrix Factorization (JSMF)** [51]. The anchor-word assumption is applicable in most large topic models [26], but PADD can also be extended to third-order tensor models that are capable of modeling topic correlations [7].

We demonstrate the effectiveness of SPI and PADD on several real-world document collections, as well as semi-synthetic corpora generated from those real collections using both correlated and uncorrelated models. PADD is both efficient and accurate, matching the performance of a long-running Gibbs Sampler in a fraction of the time.

4.2 Foundations and Related Work

In this section, we formalize matrix-based spectral topic modeling, especially JSMF. We call particular attention to the presence of a topic-topic matrix that represents joint distribution between pairs of topics in overall corpus. This ma-

trix will serve as a prior for document-specific joint probabilities between pairs of topics in later sections.

Suppose that a dataset has M documents consisting of tokens drawn from a vocabulary of N words. Topic models assume that K topics are used to generate this dataset, where each topic k is a distribution $p(x|z = k)$ over N words. Denoting all topics by the column-stochastic matrix $\boldsymbol{B} \in \mathbb{R}^{N\times K}$ where the k-th column vector $\boldsymbol{b}_k \in \Delta^{N-1}$ corresponds to the topic k, each document m is written by first choosing a composition of topics $\boldsymbol{w}_m \in \Delta^{K-1}$ from a certain prior ϱ. Then from the first position to its length n_m, a topic z is selected with respect to the composition $\boldsymbol{w}_m$, and a word x is chosen with respect to the topic $\boldsymbol{b}_z$. Different models adopt different ϱ. For example, Latent Dirichlet Allocation (LDA), [15] ϱ = Dir($\boldsymbol{\alpha}$) as depicted in Figure 4.1. For the Correlated Topic Model (CTM), [14], ϱ = $\mathcal{LN}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$.

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. If we denote all topic compositions 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 two main tasks for topic modeling are to learn **topics** (i.e., the word-topic matrix $\boldsymbol{B}$) and their **compositions** (i.e., the topic-document matrix $\boldsymbol{W}$). Inferring the latent variables $\boldsymbol{B}$ and $\boldsymbol{W}$ are coupled through the observed terms, making exact inference intractable. Likelihood-based algorithms such as Variational EM and MCMC update both parts until convergence by iterating through documents multiple times. If denoting the word-probability matrix by $\widetilde{\boldsymbol{H}}$, which is the column-normalized $\boldsymbol{H}$, these two learning tasks can be viewed as Non-negative Matrix Factorization (NMF): $\widetilde{\boldsymbol{H}} \approx \boldsymbol{BW}$, where $\boldsymbol{B}$ and $\boldsymbol{W}$ are also coupled.

Joint Stochastic Matrix Factorization The word-probability matrix $\widetilde{\boldsymbol{H}}$ is highly noisy due to its extreme sparsity, and it does not scale well with the size of dataset. Instead, let $\boldsymbol{C} \in \mathbb{R}^{N \times N}$ be the word co-occurrence matrix where $\boldsymbol{C}_{ij}$ indicates the joint probability to observe a pair of words (i, j). Then we can represent the topic modeling as a second-order non-negative matrix factorization: $\boldsymbol{C} \approx \boldsymbol{B}\boldsymbol{A}\boldsymbol{B}^T$ where we decompose the joint-stochastic $\boldsymbol{C}$ into the column-stochastic $\boldsymbol{B}$ (i.e., the word-topic matrix) and the joint-stochastic $\boldsymbol{A}$ (i.e., the topic-topic matrix). If the ground-truth topic compositions $\boldsymbol{W}^*$ that generate the data is known, we can define the posterior topic-topic matrix by $\boldsymbol{A}^* := \frac{1}{M}\boldsymbol{W}^*\boldsymbol{W}^{*T} \in \mathbb{R}^{K \times K}$ where $\boldsymbol{A}^*_{kl}$ indicates the joint posterior probability for a pair of latent topics (k, l). In this second-order factorization, $\boldsymbol{C}$ is constructed as an unbiased estimator from which we can identify $\boldsymbol{B}$ and $\boldsymbol{A}$ close to the truthful topics and their correlations.[1].

It is helpful to compare the matrix-based view of JSMF to the generative view of standard topic models. The generative view focuses on how to produce streams of word tokens for each document, and the resulting correlations between words could be implied but not explicitly modeled. In the matrix-based view, in contrast, we begin with word co-occurrence matrix which explicitly models the correlations between words and produce **pairs of words** rather than individual words. Given the prior topic correlations $\boldsymbol{A}$ between pairs of topics, each document has its own **topic correlations** $\boldsymbol{A}_m$ from $\boldsymbol{A}$ as a joint distribution $p_m(z_1, z_2)$.[2] Then for each of the possible $n_m(n_m - 1)$ pairs of positions, a topic pair (z_1, z_2) is selected first from $\boldsymbol{A}_m$, then a pair of words (x_1, x_2) is chosen with respect

[1]It is proven that the learned $\boldsymbol{A}$ is close to the $\boldsymbol{A}^*$ and the prior $\mathbb{E}_{\boldsymbol{w} \sim \dagger}[\boldsymbol{w}\boldsymbol{w}^T]$ (i.e., the population moment) for sufficiently large M. It allows us to perform topic modeling [8].

[2]Strictly speaking, $\boldsymbol{A}_m$ and $\boldsymbol{A}$ (also $\boldsymbol{A}^*$) are all joint distributions, neither covariances or correlations. However, as the covariance/correlations $\propto p(z_1, z_2) - p(z_1)p(z_2)$, which are directly inducible from $\boldsymbol{A}'s$, we keep using the naming convention from previous work, calling them *topic correlations*.

to the topics $(\boldsymbol{b}_{z_1}, \boldsymbol{b}_{z_2})$ as illustrated in Figure 4.2. Two important implications are:

- The matrix of topic correlations $\boldsymbol{A}$ represents the prior $\mathfrak{f}$ without specifying any particular parametric family.
- $\boldsymbol{A}_m$ is a rank-1 matrix $\boldsymbol{w}_m \boldsymbol{w}_m^T$ with $\boldsymbol{w}_m \sim \mathfrak{f}$, providing the fully generative story for documents.

Note that the columns of $\boldsymbol{B}$ in spectral topic models are sets of parameters rather than random variables which are sampled from another distribution $\mathfrak{g}$ (e.g., $\mathfrak{g} = \text{Dir}(\boldsymbol{\beta})$). Other work relaxes this assumption [67], but we find that it is not an issue for the present work. As putting a prior $\mathfrak{f}$ over $\{\boldsymbol{w}_m\}$ is the crux of modern topic modeling [11], our flexible matrix prior $\boldsymbol{A}$ allows us to identify the topics $\boldsymbol{B}$ from $\boldsymbol{C}$ without hurting the quality of topics. However, learning $\boldsymbol{B}$ and $\boldsymbol{A}$ via the Anchor Word algorithms might seem loosely decoupled because the algorithms first recover $\boldsymbol{B}$ and then $\boldsymbol{A}$ from $\boldsymbol{B}$ and $\boldsymbol{C}$. Previous work has found that rectifying $\boldsymbol{C}$ is essential for quality spectral inference in JSMF [51]. The empirical $\boldsymbol{C}$ must match the geometric structure of its posterior $\boldsymbol{B}\boldsymbol{A}^*\boldsymbol{B}^T$, otherwise the model will fit noise. Because this rectification step alternatingly projects $\boldsymbol{C}$ based on the geometric structures of $\boldsymbol{B}$ and $\boldsymbol{A}$ until convergence, the rest of inference would no longer require mutual updates.

Related work Second-order word co-occurrence is not by itself sufficient to identify topics [4], so much work on second-order topic models adopts the *separability assumption* such that each topic has an *anchor word* which occurs only in the context of that topic.[3] However, the first Anchor Word algorithm [8] is not able to produce meaningful topics due to numerical instability. A second

[3]Indeed, according to [26], most large topic models are separable.

version [9] works if K is sufficiently large, but the quality of topics is not fully satisfactory in real data even with the large enough K, and this version is not able to learn meaningful topic correlations. Adding a rectification step [51] as in JSMF results in high-quality topics and topic correlations, comparable to those produced by more expensive probabilistic inference methods.

There have been several extensions to the anchor words assumption that also provide identifiability. These include the Catchwords algorithm [12] and the *sufficient scatteredness* condition [42], but neither offers a solution for document composition inference.

Another approach to guarantee identifiability is to leverage third-order moments. The popular CP-decomposition [37] transforms the third-order tensor into a orthogonally decomposable form[4] and learns the topics under the assumption that the topics are uncorrelated [3]. Another method is to perform Tucker decomposition [78], which does not assume uncorrelated topics. This approach requires additional sparsity constraints for identifiability and includes more parameters to learn [4]. While correlations between topics are not an immediate by-product of tensor-based models, the PADD method presented here is still applicable for learning topic compositions of these models if the modeler chooses proper priors that can capture rich correlations [7].[5]

[4]This step is called the *whitening*, which is *conceptually* similar to the rectification in JSMF.

[5]One can also use the simple Dirichlet prior, although in theory it only captures negative correlations between topics.

4.3 Document-specific Topic Inference

In Bayesian settings, learning topic compositions $\boldsymbol{W}$ of individual documents is an inference problem, which is coupled with learning topics $\boldsymbol{B}$. As each update depends also on the parametric prior $\dagger$ and its hyper-parameters $\boldsymbol{\alpha}$, $\boldsymbol{\alpha}$ must be optimized as well to fully maximize the likelihood of the data [81]. But in the spectral setting, we can recover from higher order moments both the latent topics $\boldsymbol{B}$ and their correlations $\boldsymbol{A}$. The learned $\boldsymbol{A}$ implicitly contains the information of the proper prior $\dagger(\boldsymbol{\alpha})$ with respect to the data.[6] Since $\boldsymbol{B}$ and $\boldsymbol{A}$ are both provided and fixed, it is natural to formulate learning each column of $\boldsymbol{W}$ as an estimation problem rather than an inference problem.

Beside likelihood-based inference methods, Thresholded Linear Inference (TLI) is the only algorithm we are aware of in the recent literature that has been designed for second-order spectral inference [10]. In this section, we begin by describing TLI and SPI algorithms that only use the learned topics $\boldsymbol{B}$, then we propose our main algorithm, PADD, that uses the learned correlations $\boldsymbol{A}$ as well. By formulating the estimation as a dual decomposition [44, 71], PADD can effectively learn the compositions $\boldsymbol{W}$ given $\boldsymbol{B}$ and $\boldsymbol{A}$.

4.3.1 Simple Probabilistic Inverse (SPI)

Recall that selecting n_m words in the document m is the series of multinomial choices (i.e., $\boldsymbol{h}_m \sim \text{Mult}(n_m, \boldsymbol{B}\boldsymbol{w}_m)$). Denote $\boldsymbol{h}_m/n_m$ by $\widetilde{\boldsymbol{h}}_m$, then the conditional expectation satisfies $\mathbb{E}_{\boldsymbol{w}_m}[\widetilde{\boldsymbol{h}}_m] = \boldsymbol{B}\boldsymbol{w}_m$. If there is a left-inverse $\boldsymbol{B}^\dagger$ of $\boldsymbol{B}$ that satis-

[6] If $\dagger = \text{Dir}(\boldsymbol{\alpha})$, we can estimate $\boldsymbol{\alpha}$ via matching $\mathbb{E}_{\boldsymbol{w}\sim\dagger}[\boldsymbol{w}\boldsymbol{w}^T]$ and $\boldsymbol{A}$. A line search is sufficient to learn a quality $\boldsymbol{\alpha}$.

fies $\boldsymbol{B}^\dagger \boldsymbol{B} \approx \boldsymbol{I}_K$, then $\mathbb{E}_{\boldsymbol{w}_m}[\boldsymbol{B}^\dagger \widetilde{\boldsymbol{h}}_m] = \boldsymbol{B}^\dagger \boldsymbol{B} \boldsymbol{w}_m \approx \boldsymbol{w}_m$. However, not every left inverse is equivalent. The less $\boldsymbol{B}^\dagger \boldsymbol{B}$ is close to I_K, the more bias the estimation causes. On the other hand, large entries of $\boldsymbol{B}^\dagger$ increases variance of the estimation. To recover a high-quality document-topic decomposition, one seeks a left-inverse that balances the bias and the variance. One attractive choice for $\boldsymbol{B}^\dagger$ is the optimizer that minimizes its largest entry $|\boldsymbol{B}^\dagger|_\infty$ under the small bias constraint: $|\boldsymbol{B}^\dagger \boldsymbol{B} - \boldsymbol{I}_K|_\infty \leq \delta$.

Let $\boldsymbol{w}_m^* \in \Delta^{K-1}$ be the true topic distribution used to generate the document m. Denoting the value $|\boldsymbol{B}^\dagger|_\infty$ at the optimum by $\lambda_\delta(\boldsymbol{B})$, one can bound the maximum violation $\|\boldsymbol{B}^\dagger \widetilde{\boldsymbol{h}}_m - \boldsymbol{w}_m^*\|_\infty$ by $\delta + 2\lambda_\delta(\boldsymbol{B}) \sqrt{(\log K)/n_m}$ for an arbitrary prior $\mathfrak{f}$ from which $\boldsymbol{w}_m^* \sim \mathfrak{f}$ [10]. Thus the TLI algorithm first computes the best left-inverse $\boldsymbol{B}^\dagger$ of $\boldsymbol{B}$ given the fixed δ and linearly predicts $\boldsymbol{W} = \boldsymbol{B}^\dagger \widetilde{\boldsymbol{H}}$ via one single estimator $\boldsymbol{B}^\dagger$. Then for every column $\boldsymbol{w}_m$ of $\boldsymbol{W}$, it thresholds out each of the unlikely topics whose mass is smaller than $\tau = 2\lambda_\delta(\boldsymbol{B}) \sqrt{(\log K)/n_m} + \delta$. While TLI is supported by provable guarantees, it quickly loses accuracy if the given document m exhibits correlated topics, its length n_m is not sufficiently large, or $\boldsymbol{w}_m^*$ is not sparse enough. In addition, since the algorithm does not provide any guidance on the optimal bias/variance trade-off, users might end up computing many inverses with different δ's.[7]

We instead propose the Simple Probabilistic Inverse (SPI) method, which is a one-shot benchmark algorithm that predicts $\boldsymbol{W}$ as $\breve{\boldsymbol{B}} \widetilde{\boldsymbol{H}}$ without any additional learning costs. Recall that Anchor Word algorithms first recover $\breve{\boldsymbol{B}}$ whose $\breve{\boldsymbol{B}}_{ki} = p(z = k | x = i)$, and then convert it into $\boldsymbol{B}$ whose $\boldsymbol{B}_{ik} = p(x = i | z = k)$ via Bayes rule [9]. For the probabilistic perspective, $\breve{\boldsymbol{B}}$ is a more natural linear estimator without

[7] Recall that computing this inverse is expansive and unstable.

having any negative entries like $\boldsymbol{B}^{\dagger}$.[8] By construction, in contrast, the predicted topic composition via SPI is more likely to contain all possible topics that each word in the given document can be sampled from, no matter how negligible they are. But it can still be useful for certain applications that require extremely fast estimations with high recall. We later see in which conditions the SPI works reasonably well through the various experiments.

4.3.2 Prior-aware Dual Decomposition (PADD)

To better infer topic compositions, PADD uses the learned correlations $\boldsymbol{A}$ as well as the learned topics $\boldsymbol{B}$. While people have been more interested in finding better inference methods, many algorithms including the family of Variational Bayes and Gibbs Sampling turn out to be different only in the amount of smoothing applied to the document-specific parameters for each update [11]. On the other hand, a good prior $\dagger$ and the proper hyper-parameter $\boldsymbol{\alpha}$ are critical, allowing us to perform successful topic modeling with less information about documents, but these values are rarely considered [81].

Second-order spectral models do not specify $\dagger$ as a parametric family $\dagger(\boldsymbol{\alpha})$, but the posterior topic-topic matrix $\boldsymbol{A}^{*}$ closely captures topic prevalence and correlations. Since the learned $\boldsymbol{A}$ is close to $\boldsymbol{A}^{*}$ given a sufficient number of documents, one can estimate better topic compositions by matching the overall topic correlations (by $\boldsymbol{A}$) as well as the individual word observations (by $\boldsymbol{B}$).[9] For

[8] Due to the low-bias constraint, $\boldsymbol{B}^{\dagger}$ is destined to have many negative entries, thus yielding negative probability masses on the predicted topic compositions $\boldsymbol{B}^{\dagger}\widetilde{\boldsymbol{H}}$ even if its pure column sums are all close to 1. While such negative masses are fixed via the thresholding step, zeroing out both tiny positive masses and non-negligible negative masses is equally questionable.

[9] Because the learned $\boldsymbol{B}$ and the posterior moment $\boldsymbol{A}^{*}$ are close to the population moment $\mathbb{E}_{\boldsymbol{w}\sim\dagger}[\boldsymbol{w}\boldsymbol{w}^{T}]$ if M is sufficiently large, PADD might not be able to find quality compositions if both

a collection of M documents, PADD tries to find the best compositions $\boldsymbol{W} = \{\boldsymbol{w}_m\}$ that satisfy the following optimization:

$$\begin{aligned} \min \quad & \sum_{m=1}^{M} \|\boldsymbol{B}\boldsymbol{w}_m - \widetilde{\boldsymbol{h}}_m\|_2^2 \\ \text{subject to} \quad & \boldsymbol{w}_m \in \Delta^{K-1} \ \text{ and } \ \frac{1}{M}\sum_{m=1}^{M} \boldsymbol{w}_m \boldsymbol{w}_m^T = \boldsymbol{A}. \end{aligned} \tag{4.1}$$

Solutions from (4.1) try to match the observed word-probability $\widetilde{\boldsymbol{h}}_m$ as individuals (i.e., loss minimization), while simultaneously matching the learned topic correlations $\boldsymbol{A}$ as a whole (i.e., regularization). Therefore, whereas the performance of TLI depends only on the quality of the estimated word-topic matrix $\boldsymbol{B}$, PADD also leverages the learned correlations $\boldsymbol{A}$ to perform an analogous estimation to the prior-based probabilistic inference. With further tuning of the balance between the loss and the regularization with respect to the particular task, PADD can be more flexible for various types of data, whose topics might not empirically well fit to any known parametric prior.

4.3.3 Parallel formulation with ADMM

It is not easy to solve (4.1) due to the non-linear coupling constraint $(1/M)\sum \boldsymbol{w}_m \boldsymbol{w}_m^T = \boldsymbol{A}$. We can construct a Lagrangian by adding a symmetric matrix of dual variables $\boldsymbol{\Lambda} \in \mathbb{R}^{K\times K}$. Then $\mathcal{L}(\boldsymbol{w}_1, ..., \boldsymbol{w}_M, \boldsymbol{\Lambda})$ is equal to

$$\begin{aligned} & \sum_{m=1}^{M} \|\boldsymbol{B}\boldsymbol{w}_m - \widetilde{\boldsymbol{h}}_m\|_2^2 \quad + \langle \boldsymbol{\Lambda}, \left(\frac{1}{M}\sum_{m=1}^{M} \boldsymbol{w}_m \boldsymbol{w}_m^T\right) - \boldsymbol{A}\rangle_F \\ & = \sum_{m=1}^{M} \left\{ \|\boldsymbol{B}\boldsymbol{w}_m - \widetilde{\boldsymbol{h}}_m\|_2^2 + \frac{1}{M}\langle \boldsymbol{\Lambda}, \boldsymbol{w}_m \boldsymbol{w}_m^T - \boldsymbol{A}\rangle_F \right\} \end{aligned} \tag{4.2}$$

M and n_m are small. However, this problem also happens in probabilistic topic models, and is due to lack of information.

Algorithm 5 Estimate the best compositions $\boldsymbol{W}$.
(Master problem governing the overall estimation)

def PADD($\boldsymbol{H}, \boldsymbol{B}, \boldsymbol{A}, \lambda, \gamma$)
1: $\widetilde{\boldsymbol{H}} \leftarrow$ column-normalize($\boldsymbol{H}$)
2: $\boldsymbol{\Lambda}^{(0)} \leftarrow \boldsymbol{0}^{K\times K}$, $\boldsymbol{W}^{(0)} \leftarrow \check{\boldsymbol{B}}\widetilde{\boldsymbol{H}}$, $\boldsymbol{F} \leftarrow \gamma \boldsymbol{B}^T \widetilde{\boldsymbol{H}}$
3: **repeat**
4: $\quad \boldsymbol{G}^{(t)} \leftarrow (\gamma(\boldsymbol{B}^T\boldsymbol{B} + \frac{1}{M}\boldsymbol{\Lambda}^{(t-1)}) + \boldsymbol{I}_K)^{-1}$
5: $\quad$ **for each** $m \in \{1, ..., M\}$ (in parallel) **do**
6: $\quad\quad \boldsymbol{f}_m \leftarrow \boldsymbol{F}_m$
7: $\quad\quad \boldsymbol{w}_m^{(0)} \leftarrow \boldsymbol{W}_m^{(0)}$ (initial guess)
8: $\quad\quad \bar{\boldsymbol{w}}_m \leftarrow$ ADMM-DR($\boldsymbol{G}^{(t)}, \boldsymbol{f}_m, \boldsymbol{w}_m^{(0)}, \lambda$)
9: $\quad$ **end for**
10: $\quad \boldsymbol{\Lambda}^{(t)} \leftarrow \boldsymbol{\Lambda}^{(t-1)} - \tau_t(\boldsymbol{A} - \frac{1}{M}\sum_{m=1}^{M}(\bar{\boldsymbol{w}}_m\bar{\boldsymbol{w}}_m^T)))$
11: **until** the convergence
12: **return** $\boldsymbol{W} = \{\bar{\boldsymbol{w}}_1|...|\bar{\boldsymbol{w}}_M\}$

The equation (4.2) implies that given a fixed dual matrix $\boldsymbol{\Lambda}$, minimizing the Lagrangian can be decomposed into M subproblems, allowing us to use the dual decomposition [44, 71]. Each subproblem tries to find the best topic composition $\boldsymbol{w}_m \in \Delta^{K-1}$ that minimizes $\|\boldsymbol{B}\boldsymbol{w}_m - \widetilde{\boldsymbol{h}}_m\|_2^2 + (1/M)\langle \boldsymbol{\Lambda}, \boldsymbol{w}_m\boldsymbol{w}_m^T - \boldsymbol{A}\rangle_F$.[10] Once every subproblem is solved and has provided the current optimal solution $\bar{\boldsymbol{w}}_m$, the master problem simply updates the dual matrix based on its subgradient: $-\frac{1}{M}(\sum_{m=1}^{M}(\bar{\boldsymbol{w}}_m\bar{\boldsymbol{w}}_m^T - \boldsymbol{A})) \in \partial(\boldsymbol{\Lambda})$, and then distributes it back to each subproblem. For robust estimation, we adopt the Alternating Direction Method of Multiplier (ADMM) [13] with Douglas-Rachford (DR) splitting [55]. Then the overall procedures become as illustrated in Algorithm 5 and 6.

Note first that the outer loop in Algorithm 5 computes a matrix inverse, but the computation is cheap and stable. This is because it only algebraically inverts a $K \times K$ matrix rather than solving a constraint optimization to invert an $N \times K$ matrix as TLI does. Note also that the overall algorithm repeats the mas-

[10] The operation $\langle \cdot, \cdot \rangle_F$ indicates the Frobenius product, which is the matrix version of the inner product.

Algorithm 6 Estimate the best individual $\boldsymbol{w}_m$.
(Subproblem running for each document m in parallel)

def ADMM-DR($\boldsymbol{G}, \boldsymbol{f}, \boldsymbol{w}^{(0)}, \lambda$)

$\boldsymbol{q}^{(0)} \leftarrow \boldsymbol{w}^{(0)}$
repeat
  $\boldsymbol{p}^{(t)} \leftarrow \boldsymbol{G}(2\boldsymbol{w}^{(t-1)} - \boldsymbol{q}^{(t-1)} + \boldsymbol{f})$
  $\boldsymbol{q}^{(t)} \leftarrow \boldsymbol{q}^{(t-1)} + \lambda(\boldsymbol{p}^{(t)} - \boldsymbol{w}^{(t-1)})$
  $\boldsymbol{w}^{(t)} \leftarrow \Pi_{\Delta^{K-1}}(\boldsymbol{q}^{(t)})$
until the convergence of $\boldsymbol{w}^{(t)}$
$\bar{\boldsymbol{w}} \leftarrow \boldsymbol{w}^{(t)}$
return $\bar{\boldsymbol{w}}$

($\Pi_{\Delta^{K-1}}(\cdot)$ is the orthogonal projection to the simplex Δ^{K-1}. See the reference for the detailed implementation [28].)

ter problem only a small number of times, whereas each subproblem repeats the convergence loop more times. The exponentiated gradient algorithm [9] is also applicable for quick inference, but tuning the learning rate would be less intuitive, although users of PADD should be careful in tuning the learning rate τ_t due to its non-linear characteristics. Note last that we need not further project the subgradient to the set of symmetric matrices because only the symmetric matrices $\boldsymbol{A}$ and $\bar{\boldsymbol{w}}_m \bar{\boldsymbol{w}}_m^T$ are added and subtracted from $\boldsymbol{\Lambda}$ for every iteration of the master problem.

Why does it work? Probabilistic topic models try to infer both topics $\boldsymbol{B}$ and document compositions $\boldsymbol{W}$ that approximately maximize the marginal likelihood of the observed documents: $\prod_{m=1}^{M} \int_{\boldsymbol{w}_m} p(\boldsymbol{w}_m|\boldsymbol{\alpha}) \prod_{i=1}^{N} (\boldsymbol{B}\boldsymbol{w}_m)_i^{\boldsymbol{h}_{mi}} d\boldsymbol{w}_m$, considering all possible topic distributions $\boldsymbol{w}_m \in \Delta^{K-1}$ under the prior $\dagger(\boldsymbol{\alpha})$. However, if the goal in spectral settings is to find the best individual composition $\boldsymbol{w}_m$ provided with the learned $\boldsymbol{B}$ and $\boldsymbol{A}$, the following MAP estimation

$$\arg\max_{\boldsymbol{w}_m \in \Delta^{K-1}} p(\boldsymbol{w}_m; \boldsymbol{A}) \prod_{i=1}^{N} (\boldsymbol{B}\boldsymbol{w}_m)_i^{\boldsymbol{h}_{mi}} \quad (\boldsymbol{h}_{mi} = \boldsymbol{H}_{im}) \tag{4.3}$$

is a reasonable pointwise choice in the likelihood perspective. Recall that the MLE parameters that maximize the likelihood of the multinomial choices $\boldsymbol{h}_m$ is to assign the word-probability parameters $\boldsymbol{B}\boldsymbol{w}_m$ equal to the empirical frequencies $\widetilde{\boldsymbol{h}}_m$. The loss function of the PADD objective tries to find the best $\boldsymbol{w}_m$ that makes $\boldsymbol{B}\boldsymbol{w}_m \approx \widetilde{\boldsymbol{h}}_m$, maximizing the second term $\prod_{i=1}^{N}(\boldsymbol{B}\boldsymbol{w}_m)_i^{\boldsymbol{h}_{mi}}$ in Equation (4.3).

While we are not aware of methods for sampling a rank-1 correlation matrix $\boldsymbol{A}_m = \boldsymbol{w}_m\boldsymbol{w}_m^T$ from $\boldsymbol{A}$ directly, PADD maximizes the first term $p(\boldsymbol{w}_m; \boldsymbol{A})$ in (4.3) by preventing $\boldsymbol{w}_m$'s from deviating too far from the learned topic correlations $\boldsymbol{A}$, which is a good approximation of the prior: the population moment $\mathbb{E}_{\boldsymbol{w}\sim\dagger}[\boldsymbol{w}\boldsymbol{w}^T]$. Indeed, when learning the document-specific topic distributions for spectral topic models, it is shown that a proper point estimation is likely a good solution also in the perspective of Bayesian inference because the posterior is concentrated on the ϵ-ball of the point estimator with high probability [10].

4.4 Experimental Results

We present experiments on real documents and two varieties of semi-synthetic documents. Evaluating reconstructed topic compositions is not easy for real data because no ground truth compositions exist for quantitative comparison. Unlike topic-word distributions, document-topic distributions do not support qualitative evaluations because of the number of documents, and because topics in each document may not be as obviously coherent or incoherent as words in each topic [21]. Synthesizing documents from scratch is an option as we can manipulate the ground truth $\boldsymbol{B}$ and $\boldsymbol{W}^*$, but the resulting documents would not be realistic. Thus we generate documents from two distinct processes that

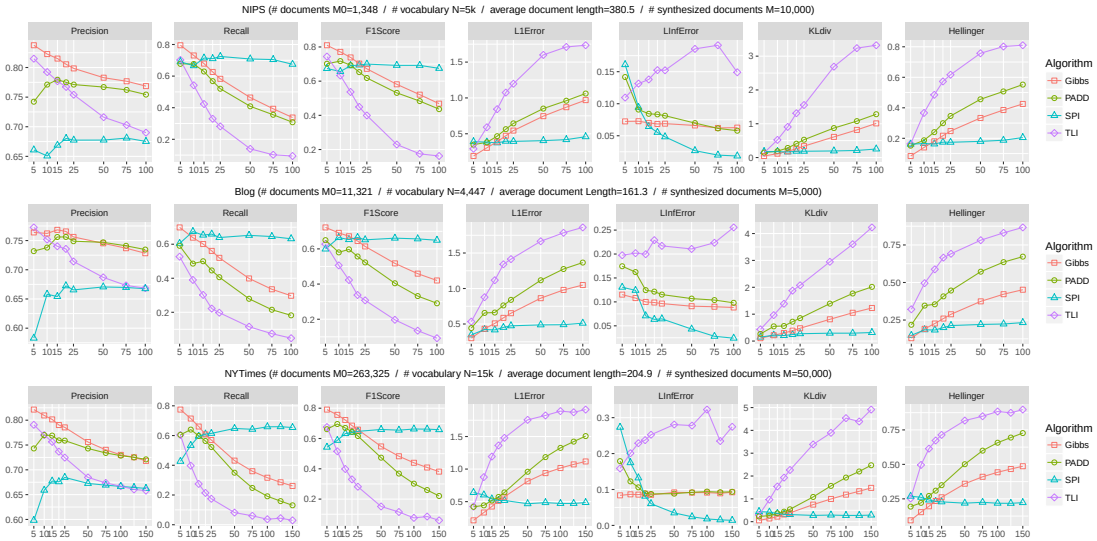

Figure 4.3: Artificial experiment on Semi-Synthetic (*SS*) corpus with highly sparse topics and little correlation. X-axis: # topics K. Y-axis: higher numbers are better for the left three columns, lower numbers are better for the right four. SPI performs the best with $K \geq 25$.

sample synthetic documents based on models trained on real data, one that samples uncorrelated topics and another that samples topics with correlation.

The uncorrelated setting (semi-synthetic, *SS*) involves sampling from an LDA model with a Dirichlet prior. We first extract K topics $\boldsymbol{B}_0$ and their correlations $\boldsymbol{A}_0$ from each real corpus of training data $\boldsymbol{H}_0$ using JSMF [51]. We next sample M columns of $\boldsymbol{W}^*$ from a rarely correlated Dir($\boldsymbol{\alpha}$) with $\boldsymbol{\alpha} = (5/K)\vec{1}$, and then synthesize a corpus $\boldsymbol{H}_{SS}$ by sampling each document m with respect to the topics $\boldsymbol{B}_0$ and the compositions $\boldsymbol{W}^*$, matching the average document length of the original corpus.

The correlated setting (semi-real, *SR*) involves sampling from a CTM model [14] with a logistic-normal prior. We first learn K topics $\boldsymbol{B}_0$ with the topic means $\boldsymbol{\mu}_0$ and covariance $\boldsymbol{\Sigma}_0$ for each training corpus $\boldsymbol{H}_0$ using CTM-Gibbs [22]. We then sample M columns of $\boldsymbol{W}^*$ from $\mathcal{LN}(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)$, synthesizing a corpus $\boldsymbol{H}_{SR}$ analogously. While it is less realistic, the uncorrelated corpus *SS* provides a fair

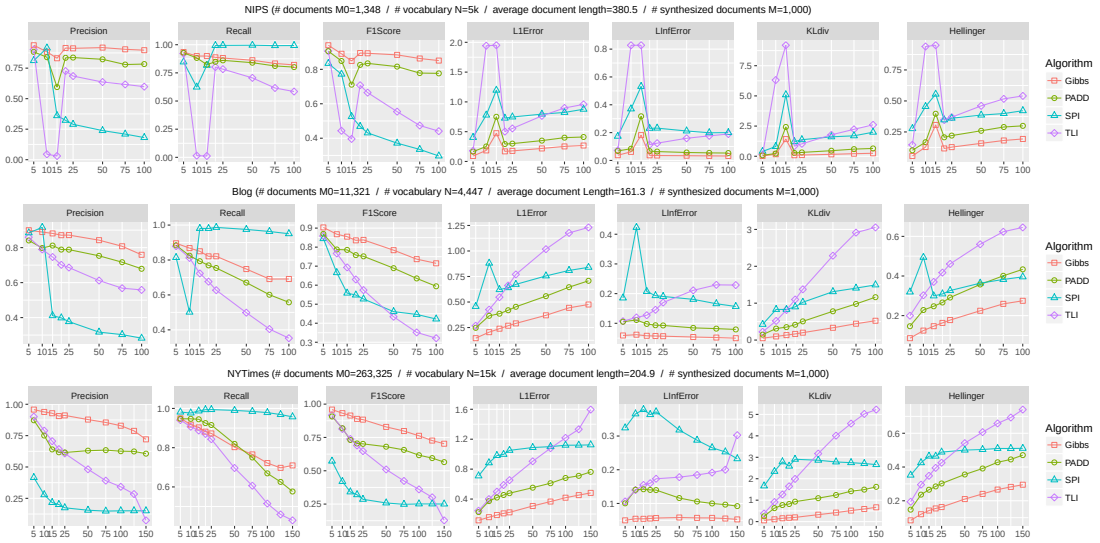

Figure 4.4: Realistic experiment on Semi-Real (*SR*) corpus with non-trivial topic correlations. X-axis: # topics K. Y-axis: higher numbers are better for the left three columns, lower numbers are better for the right four. PADD is consistent and comparable to Gibbs Sampling.

comparison to the experiments of TLI in [10], whereas the correlated *SR* exploits the learned hyper-parameters $(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)$ so that it can maximally simulate the real world characteristics with non-trivial correlations between topics. Note that we use CTM-Gibbs for constructing synthetic data here only because it is a well-known model that supports topic correlation; we do not find that it is a competitive inference method.[11]

For Fully-Real (*FR*) experiments, we prepare the unseen documents $\boldsymbol{H}_{US}$, which is 10% of the original data which has never been used in training (i.e., $\boldsymbol{H}_0 \cap \boldsymbol{H}_{US} = \emptyset$), and then test on $\boldsymbol{H}_{US}$ as well as the original training set $\boldsymbol{H}_0$. We use the result of a long-running Gibbs sampler as a proxy to the ground-truth $\boldsymbol{W}^*$ equivalent to [10]. [12]

[11] Similar to [69], we notice that CTM often puts spuriously high probability mass to one particular topic, though it is capable of learning quality correlations. Thus we use Dir-Gibbs instead of CTM-Gibbs in learning topic compositions.

[12] After 200 burn-in iterations we run 1,000 further iterations to gather samples from the posterior using Mallet, providing the original topics $\boldsymbol{B}_0$ and fitted hyper-parameters based on $\boldsymbol{A}_0$. Only the topic compositions $\boldsymbol{W}$ are updated over iterations.

As strong baselines, we run TLI and SPI on the *SS*, *SR*, and *FR* corpora with the real parameters: $\boldsymbol{B}_0$ and $\check{\boldsymbol{B}}_0$, while PADD also uses $\boldsymbol{A}_0$. For Gibbs Sampling, we use the ground-truth hyper-parameters $(5/K)\vec{1}$ for the *SS* corpus, whereas we determine the best Dirichlet hyper-parameters $\boldsymbol{\alpha}$ for the *SR* and *FR* corpora by matching the topic-topic moments between $(1/M)\boldsymbol{W}^*\boldsymbol{W}^*$ and $\mathbb{E}_{\boldsymbol{w}\sim\mathrm{Dir}(\boldsymbol{\alpha})}[\boldsymbol{w}\boldsymbol{w}^T]$.

Following [51], we use the standard sources: NIPS papers and NYTimes articles. We also add political blogs [30] to experiment variations in document lengths. For evaluating information retrieval performance, we first find the prominent topics whose cumulative mass is close to 0.8 for each document, and compute the precision, recall, and F1 score as [84]. For measuring distributional similarity, we use KL-divergence and Hellinger distance. In contrast to assymetric KL, Hellinger is a symmetric and normalized distance used for evaluating the CTM. For comparing the reconstruction errors with TLI, we also report ℓ_1-error and ℓ_∞-error [10]. For fully real experiments, we also report the distance to prior $\|\boldsymbol{A}_0 - (1/M)\boldsymbol{W}\boldsymbol{W}^T\|_F$ and the mass on non-supports, the total probability that each algorithm puts on non-prominent topics of the ground-truth composition $\boldsymbol{W}^*$.

In the uncorrelated Semi-Synthetic (*SS*) experiments given in Figure 4.3, SPI performs the best as the number of topics K increases. As expected, SPI is good at recall all possible topics but loses precision at the same time. Note that even Gibbs Sampling shows relatively high ℓ_1-error especially for the models with large K. This is because $\mathrm{Dir}((5/K)\vec{1})$ generates highly sparse compositions, so any variability in other topics causes catastrophic errors even with sufficiently mixed Gibbs Sampling. The same problem also happens in [10]. TLI performs well only for tiny topic models. Despite the unrealistic nature of *SS*, PADD

Data	Algo	Precision	Recall	F1-score	ℓ_1-error	ℓ_∞-error	Hellinger	Prior-dist	Non-supp
NIPS	Rand	.092/.107	.426/.429	.260/.268	1.60/1.55	0.52/0.44	0.74/0.72	.0123/.0126	.908/.894
(1.3k/	TLI	.396/.421	.954/.890	.513/.536	0.88/0.88	0.29/0.24	0.44/0.45	.0085/.0080	.617/.604
143)	SPI	.169/.193	**.994/.994**	.271/.307	1.20/1.14	0.41/0.34	0.55/0.53	.0118/.0118	.771/.755
	PADD	**.691/.734**	.876/.740	**.759/.744**	**0.47/0.55**	**0.16/0.18**	**0.27/0.31**	**.0049/.0016**	**.382/.324**
Blog	Rand	.095/.100	.424/.428	.261/.263	1.55/1.53	0.49/0.46	0.70/0.69	.0101/.0100	.905/.900
(11k/	TLI	.421/.422	.856/.814	.530/.527	0.95/0.97	0.28/0.27	0.47/0.49	.0066/.0067	.619/.623
1.2k)	SPI	.170/.176	**.982/.979**	.279/.289	1.18/1.16	0.39/0.37	0.51/0.50	.0106/.0107	.782/.776
	PADD	**.642/.671**	.800/.758	**.713/.715**	**0.61/0.62**	**0.21/0.21**	**0.33/0.34**	**.0057/.0044**	**.428/.403**
NY	Rand	.061/.062	.427/.427	.207/.208	1.73/1.73	0.62/0.61	0.80/0.79	.0188/.0188	.939/.938
Times	TLI	.321/.320	.925/.922	.428/.427	1.16/1.16	0.40/0.40	0.56/0.56	.0078/.0197	.690/.690
(263k/	SPI	.183/.183	**.964/.964**	.281/.281	1.29/1.29	0.47/0.47	0.58/0.58	.0143/.0143	.775/.775
30k)	**PADD**	**.474/.543**	.899/.904	**.575/.668**	**0.76/0.71**	**0.27/0.28**	**0.39/0.37**	**.0005/.0113**	**.520/.493**

Table 4.1: Real experiment on Fully-Real (*FR*) corpora. For each entry, a pair of values indicates the corresponding metrics on training/unseen documents. Averaged across all models with different K's. Rand estimates randomly. For two new metrics: Prior-dist and Non-supp, smaller numbers are better. PADD performs the best considering topic compositions learned by Gibbs Sampling as the ground-truth.

outperforms TLI by a large margin in most cases, showing similar behaviors to probabilistic Gibbs Sampling across all datasets.

The situation is quite different in the correlated Semi-Real (*SR*) experiments shown in Figure 4.4. SPI's high recall is no longer helpful because of its drastic loss of precision, whereas PADD is comparable to Gibbs Sampling across all models and datasets even with only 1,000 documents. We also vary the number of synthesized documents up to 100k, verifying that the results are mostly stable. This is because PADD captures correlations through its prior-aware formulation. TLI performs poorly because it is linear, and does not consider topic correlations.

When testing on the Fully-Real (*FR*) corpora, PADD shows the best performance on both training documents and the unseen documents. Considering that Gibbs Sampling with the ground-truth parameters does not have perfect accuracy in other settings, the metrics evaluated against Gibbs Sampling in Table 4.1 are noteworthy. Prior-dist, the Frobenius distance to the prior $\boldsymbol{A}_0$, implies PADD-learned $\boldsymbol{w}_m$ likely improves $p(\boldsymbol{w}_m; \boldsymbol{A}_0)$ than other algorithms. While TLI uses provably guaranteed thresholding,[13] Non-supp values show that it still puts more than half of probability mass on the non-prominent topics in average.

Although we are optimizing for accuracy rather than speed, PADD converges efficiently relative to comparable methods. We iterate 15 times for the master procedure PADD and 150 times for the slave procedure ADMM-DR with $(\lambda, \gamma) = (1.9, 3.0)$.[14] When using an equivalent level of parallel processing,

[13]We use the same less conservative threshold $\tau/4.5$ and unbias setting with $\delta = 0$ as conducted in [10]. We also try to loosen the unbias constraint when the inversion fails, but it does not help.

[14]Inference quality is almost equivalent when running only 100 times for each slave procedure, and is not sensitive to parameter values: $1.0 \leq \gamma \leq 5.0$. The parameter $\lambda = 1.9$ is known best in optimization literature.

computing $\boldsymbol{B}_0^\dagger$ via TLI takes 2,297 seconds,[15] whereas PADD takes 849 seconds for the entire inference on the semi-synthetic NIPS dataset with $K = 100$ and $M = 10,000$. SPI is the by far the fastest, requiring one matrix multiplication; our Gibbs configuration takes 3,794 seconds on the same machine. While we choose ADMM-DR mainly for the tightest optimization, one can easily incorporate faster gradient-based algorithms inside our formulation of prior-aware dual decomposition.

4.5 Discussion

Fast and accurate topic inference for new documents is a vital component of a topic-based workflow, especially for spectral algorithms that do not by themselves produce document-topic distributions even for training documents. For mixed-membership data with little topic correlation, we find that our Simple Probabilistic Inverse (SPI) performs well. Although this is rarely true for textual documents, topic models can be also applicable to a wide variety of other modalities. SPI is extremely fast but fails when a word has large probability in two or more topics, as it is not able to disambiguate based on context, thereby naively distributing its prediction weights. Future research may offer ways to threshold or post-process SPI's estimation analogous to TLI, or in its use as an initialization.

We find that Prior-aware Dual Decomposition (PADD) performs comparable to probabilistic Gibbs Sampling especially for realistic data. PADD provides

[15] We also observe that AP-rectification in JSMF significantly boosts the condition of $\boldsymbol{B}_0$ on various datasets, removing TLI's failures in computing the left-inverse $\boldsymbol{B}_0^\dagger$. However, even if the inverse is computed, TLI sometimes yields NaN values due to numerical instability of matrix inversion. We omit those results in evaluation to prevent TLI's quality degradation.

rigorous theoretical motivation and an efficient parallel implementation. The experimental results show that PADD also predicts the topic compositions of unseen real data well, notably outperforming the existing TLI method. With robust and efficient topic inference that is aware of topical correlations latent in the data, we can now fill out the necessary tools to make spectral topic models a full competitor to likelihood-based methods. Although the benefits of PADD for topic inference are mostly relevant in second-order spectral methods, they are flexibly applicable in any setting that involves inferring mixture proportions.

APPENDIX A

APPENDIX FOR CHAPTER 2

A.1 Introduction

This is a supplementary document for the paper: Robust Spectral Inference for Joint Stochastic matrix Factorization. It is organized accordingly to the main paper so that the readers can find the missing proofs, deferred details, and further explanations in the corresponding sections. We also include more algorithms, experiments, and analysis that are discarded from the main paper due to the page limit.

A.2 Requirements for Factorization

Proof for uniqueness of JSMF. When factorizing co-occurrence matrix $\boldsymbol{C}$ into $\boldsymbol{BAB}^T$ with constraints $\boldsymbol{B}$: $N \times K$ column-stochastic and $\boldsymbol{A}$: $K \times K$ joint-stochastic, the resulting $(\boldsymbol{B}, \boldsymbol{A})$ may not be an unique decomposition of $\boldsymbol{C}$ if $K \geq 2$. Assume there exists a $K \times K$ column-stochastic square matrix $\boldsymbol{Y}$ such that $\boldsymbol{Y}$ and $\boldsymbol{Y}^{-1}$ are both non-negative. Then,

$$\boldsymbol{C} \approx \boldsymbol{BAB}^T = \boldsymbol{B}(\boldsymbol{YY}^{-1})\boldsymbol{A}(\boldsymbol{YY}^{-1})\boldsymbol{B}^T = (\boldsymbol{BY})(\boldsymbol{Y}^{-1}\boldsymbol{AY}^{-T})(\boldsymbol{BY})^T. \tag{A.1}$$

As $\boldsymbol{BY}$ is $N \times K$ column-stochastic and $\boldsymbol{Y}^{-1}\boldsymbol{AY}^{-T}$ is $K \times K$ joint-stochastic, $(\boldsymbol{BY}, \boldsymbol{Y}^{-1}\boldsymbol{AY}^{-T})$ can be another equally meaningful solution for JSMF of $\boldsymbol{C}$. In fact, it is known that if an inverse of non-negative matrix $\boldsymbol{Y}$ is again non-negative, $\boldsymbol{Y}$

must be a *generalized permutation matrix* which satisfies $\boldsymbol{Y} = \boldsymbol{DP}$ for some diagonal matrix $\boldsymbol{D}$ and permutation matrix $\boldsymbol{P}$. Since both $\boldsymbol{BY}$ and $\boldsymbol{B}$ are column-stochastic, $\boldsymbol{Y}$ must be column-stochastic as well. Thus the only diagonal matrix $\boldsymbol{D}$ that makes $\boldsymbol{DP}$ column-stochastic with respect to permutation matrix $\boldsymbol{P}$ is the identity matrix. Therefore we can conclude that the only possible $\boldsymbol{Y}$ is a permutation matrix. It means that our factorization is **unique up to the column permutation**. This is equivalent to the fact that there is no order between resulting topics in probabilistic topic models.

Proof for double non-negativity of posterior co-occurrence. Take any vector $y \in \mathbb{R}^N$ and say $y' = \boldsymbol{B}^T y$. Then

$$y^T \mathbb{E}[\boldsymbol{C}^*] y = y^T \boldsymbol{B} \boldsymbol{A}^*_M \boldsymbol{B}^T y = (y')^T \boldsymbol{A}^*_M y' \geq 0 \quad (\because \boldsymbol{A}^*_M \in \mathcal{PSD}_K). \tag{A.2}$$

Thus $\boldsymbol{C}^* \in \mathcal{PSD}_N$. Also, $\boldsymbol{C}^*_{ij} = p(X_1 = i, X_2 = j) \geq 0$ for all i, j. Therefore $\boldsymbol{C}^* \in \mathcal{DNN}_N$.

A.3 Rectified Anchor Word Algorithm

Rectifying co-occurrence C by Diagonal Completion (DC). As we explained in Section 6 of the main paper, the diagonal entries of the co-occurrence matrix are the most difficult elements to interpret and the least likely to conform to the model. For instance, frequent words in a document are likely to be bursty, leading to large diagonal elements; but popular songs appear at most a few times in any given playlist, leading to relatively small and noisy diagonal elements. Instead of ignoring such high variance, we fix the diagonal so that $\boldsymbol{C}$ has low rank.

Algorithm 7 Diagonal Completion (DC)

In: $\boldsymbol{F}$: $(N/2) \times (N/2)$ block of $\boldsymbol{C}$ in diagonal side
$\boldsymbol{G}$: $(N/2) \times (N/2)$ block of $\boldsymbol{C}$ in off-diagonal side
Out: $d \in \mathbb{R}^{N/2}$: a vector of new diagonal entries
def DIAGONAL-COMPLETE($\boldsymbol{F}, \boldsymbol{G}$)
  $(\boldsymbol{U}, \Sigma, \boldsymbol{V}) \leftarrow$ *truncated-svd*$(\boldsymbol{G}, K)$
  $\boldsymbol{L} \leftarrow \boldsymbol{U}^T \times (\boldsymbol{F} - \boldsymbol{F}_{diag})$
  for $j = 1$ to $N/2$ **do**
    $u_j^T \leftarrow \boldsymbol{U}_{j*}$
    $d_j \leftarrow \frac{1}{1-\|u_j\|_2^2}(u_j^T \times \boldsymbol{L}_{*j})$
  end for
  return d

Algorithm 7 estimates the diagonal elements of $\boldsymbol{C}$ from the off-diagonal elements, assuming the off-diagonal elements come from a low-rank matrix. The key observation is that the top or bottom halves of $\boldsymbol{C}$ are themselves low rank, and we can find the range space for each matrix from those columns that are completely known. Once we know a space in which all the columns of the top half of $\boldsymbol{C}$ should belong, we can determine the unknown diagonal elements through a least-squares fit using the known elements.

More concretely, the algorithm proceeds by partitioning $\boldsymbol{C}$ into four quadrants of near-equal size. Let $\boldsymbol{F} = \boldsymbol{C}_{11}$ and $G = \boldsymbol{C}_{12}$, and for each $j \in [1, N/2]$, let J be all indices from 1 to $N/2$ except j. We want each column $\boldsymbol{F}_{*j}$ to dwell in the range space of G. We find a basis $\boldsymbol{U}$ for this range space from the first K left singular vectors of G. To find $\boldsymbol{F}_{jj}$, we seek a K-dimensional vector y such that $(\boldsymbol{U}_{J*})y = \boldsymbol{F}_{Jj}$. Because we are unlikely to exactly satisfy this equation, we seek the least-square solution

$$(\boldsymbol{U}_{J*})^T(\boldsymbol{U}_{J*})y = (\boldsymbol{U}_{J*})^T\boldsymbol{F}_{Jj}. \tag{A.3}$$

Denote the $K \times K$ identity matrix by $\boldsymbol{I}_K$ and j-th row vector of $\boldsymbol{U}$ by u_j^T. Since $\boldsymbol{U}$ has orthonormal columns, $(\boldsymbol{U}_{J*})^T(\boldsymbol{U}_{J*}) = \boldsymbol{I}_K - u_j u_j^T$. By the Sherman-Morrison

formula,

$$\left(\boldsymbol{I}_K - u_j u_j^T\right)^{-1} = \boldsymbol{I}_K + u_j u_j^T/(1 - u_j^T u_j). \tag{A.4}$$

Let $\boldsymbol{L}_{*j} = (\boldsymbol{U}_{J*})^T \boldsymbol{F}_{Jj}$. Under the low rank assumption, the diagonal should be $d_j = (\boldsymbol{U}y)_j = (u_j^T)y$, and therefore

$$d_j = (u_j^T)\left(\boldsymbol{L}_{*j} + \frac{u_j u_j^T}{1 - u_j^T u_j}\boldsymbol{L}_{*j}\right) = \frac{u_j^T \boldsymbol{L}_{*j}}{1 - u_j^T u_j}. \tag{A.5}$$

As we precompute $\boldsymbol{L}$ and run the truncated SVD on a half-size block with $K \ll N$, DC is efficient. Simply execute Algorithm 7 twice with the inputs $(\boldsymbol{C}_{11}, \boldsymbol{C}_{12})$ and $(\boldsymbol{C}_{22}, \boldsymbol{C}_{21})$, and replace the existing diagonal with the output vector e. We present an error analysis in Section 6.

Selecting basis S. After rectification, the next step is to select the subset S of objects that satisfy the separability assumption. Our goal is to choose the K best rows of the row-normalized co-occurrence matrix $\boldsymbol{C}$ so that all other rows lie nearly in the convex hull of the selected rows. [9] use the Gram-Schmidt process to select these anchor rows, but they do not use the sparsity of $\boldsymbol{C}$. In order to scale beyond relatively small vocabularies, they resort to random projections that approximately preserve ℓ_2 distances.

Denote the row-normalized $\boldsymbol{C}$ matrix by $\boldsymbol{C}$. Then by the conditional independence,

$$\boldsymbol{C}_{ij} = p(X_2 = j|X_1 = i) = \sum_{k'} p(X_2 = j|Z_1 = k')p(Z_1 = k'|X_1 = i). \tag{A.6}$$

Let $S = \{s_1, ..., s_K\}$ be the set of K basis objects. Then $\boldsymbol{C}_{s_k,j} = p(X_2 = j|Z_1 = k)$

Algorithm 8 Finding Bases S

In: $\boldsymbol{P}$: $N \times N$ matrix (e.g., $P \leftarrow \boldsymbol{C}^T$)
Out: S: the set of K indices
$r \in \mathbb{R}^K$: a vector of distances to each subspace
def FIND-S($\boldsymbol{P}$)
Initialize $S \leftarrow \varnothing$, $\boldsymbol{Q} \leftarrow 0^{N \times K}$, $r \leftarrow 0^K$
$norm \leftarrow$ squared norms of column vectors of $\boldsymbol{P}$
for $k = 1$ to K **do**
$n \leftarrow \text{argmax}_{1 \le i \le N}\, norm(i)$
$S \leftarrow S \cup \{n\}$, $\boldsymbol{Q}_{*k} \leftarrow \boldsymbol{P}_{*n}$, $r_k \leftarrow \sqrt{norm(n)}$
$\boldsymbol{Q}_{*k} \leftarrow (\boldsymbol{Q}_{*k} - \sum_{l=1}^{k-1} \langle \boldsymbol{Q}_{*l}, \boldsymbol{P}_{*n} \rangle \boldsymbol{Q}_{*l}) / r_k$
$norm \leftarrow norm - (\boldsymbol{Q}_{*k}^T \boldsymbol{P}) \circ (\boldsymbol{Q}_{*k}^T \boldsymbol{P})$
end for
return (S, r)

($\circ$ operation is the Hadamard Product, a simple element-wise multiplication between two vectors)

because the separability assumption implies

$$p(Z_1 = k' | X_1 = s_k) = \begin{cases} 1 & (\text{if } k' = k) \\ 0 & (\text{if } k' \neq k) \end{cases}. \tag{A.7}$$

Thus $\boldsymbol{C}_{ij} = \sum_k p(Z_1 = k | X_1 = i) \boldsymbol{C}_{s_k, j}$, which means every row vector of $\boldsymbol{C}$ can be represented by a convex combination of the row vectors corresponding to the basis objects.

The (unprojected) Gram-Schmidt process in [9] computes a *pivoted QR decomposition* [35]. Several other algorithms compute the same decomposition and exploit sparsity [74]. In particular, one can find the set S with $O(NK)$ auxiliary space and $O(\text{nnz}(C)K)$ time without modifying $\boldsymbol{C}$; this has the advantage that $\boldsymbol{C}$ is unchanged in memory and ready for use in the recovery step. Algorithm 8 requires only $O(NK)$ space to store $\boldsymbol{Q}$ doing every update implicitly rather than changing the original input matrix. Not modifying $\boldsymbol{C}$ in place has the additional advantage of leaving $\boldsymbol{C}$ unchanged in memory and ready for use in the recovery

step. Note that we only return the set of indices S corresponding to the basis objects and the diagonal entries r of $\boldsymbol{R}$ as their absolute values.

In Algorithm 8, *norm* is a N-dimensional row vector that provides a criterion to greedily choose the next best column for column-pivoting. It is updated once at the end of each iteration because for each $1 \leq j \leq N$,

$$\|\boldsymbol{P}_{*j} - \langle \boldsymbol{Q}_{*k}, \boldsymbol{P}_{*j} \rangle\|_2^2 = \langle \boldsymbol{P}_{*j}, \boldsymbol{P}_{*j} \rangle - \langle \boldsymbol{Q}_{*k}, \boldsymbol{P}_{*j} \rangle^2. \tag{A.8}$$

(*norm* was initialized as the first inner-product term). However, this greedy strategy is only one way of approaching the general problem of *subset selection*. Recent work on this subject includes [17, 19]. [58] present a CUR decomposition whose matrix factors consist of columns and rows of the input matrix. These alternate subset selection strategies were not designed for non-negative approximation; unlike ordinary pivoted QR, they will not necessarily recover the desired basis in the absence of noise. In the presence of noise and model error, however, these alternate selection strategies may merit further attention.

Recovering cluster-example $\boldsymbol{W}$. Recall that the standard topic modeling consists of two inferences: inferring topic distributions in terms of words and inferring document distributions in terms of topics. So far, we have recovered the cluster-cluster interaction $\boldsymbol{A}$, which is a noisy expectation of $\boldsymbol{W}\boldsymbol{W}^T$ instead of directly seeking $\boldsymbol{W}$. In our JSMF model, $\boldsymbol{W}$ is unknown and its columns (i.e., example-cluster distributions) are stochastically generated from a known distribution $f(\boldsymbol{\alpha})$ governed by the hyperparameter $\boldsymbol{\alpha}$, rather than sets of parameters to estimate. [9] points out that $\boldsymbol{W}$ is never be able to be recovered in this sense, especially under the limited samples.

Once we recover quality object-cluster matrix B after rectification based on

the doubly non-negative geometry of $\boldsymbol{A}^*_M, \boldsymbol{C}^*$, however, we can further try to recover cluster-example matrix $\boldsymbol{W}$ assuming our recovered $\boldsymbol{A}$ is quite close to $\boldsymbol{A}^*_M$. Since $\mathbb{E}[\boldsymbol{H}_m] = n_m \boldsymbol{B}\boldsymbol{W}_m$ for each example m, we can compute $\boldsymbol{W}_m$ by solving the following simplex-constrained Non-Negative Least Square (NNLS) problem:

$$\min_{\boldsymbol{W}_m \in \Delta^K} \|\boldsymbol{B}\boldsymbol{W}_m - \boldsymbol{H}_m/n_m\|$$

This optimization can be solved via exponentiated gradient algorithm similar to what we use for recovering object-cluster matrix $\boldsymbol{A}$. Analogously, it can be easily parallelizable by per-document fashion because we are solving independent optimization for each document given inferred $\boldsymbol{B}$.

A.4 Experimental Results

Qualitative results. The 15 clusters from the Movies dataset is attached at the end. One can verify that while Gibbs learns slightly better clusters, AP's results are comparable, whereas Baseline algorithm learns nothing.

Quantitative results. The following shows full results from real experiments.

Legality ($\sum_{k=1}^K \sum_{l=1}^K \boldsymbol{A}_{kl} = \sum_{k=1}^K \sum_{l=1}^K p(Z_1 = k, Z_2 = l)$) assesses how close the recovered cluster-cluster matrix $\boldsymbol{A}$ is to a legal joint distribution whose entries sum to 1. The results show that the recovered $\boldsymbol{A}$ becomes close to a legal joint distribution under the DC and AP rectification, whereas the entry sum for Baseline is far higher than 1. Note that we intentionally avoid projecting the recovered $\boldsymbol{A}$ down to $\mathcal{DNN}_K$ in order to verify the quality of our new recovery algorithm in terms of legality. **Validity** ($\underset{sym}{KL}\left(\sum_{k=1}^K p(Z_1, Z_2 = k) \| \sum_{i=1}^N p(Z_1|W_1 = i)p(W_1 = i)\right)$)

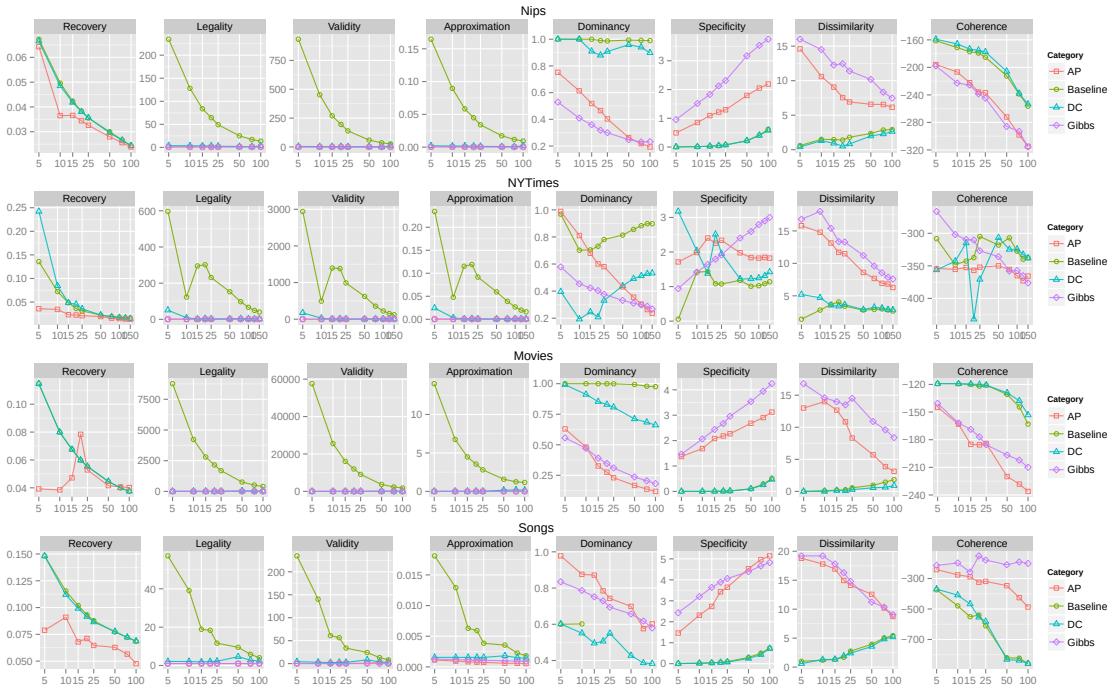

Figure A.1: Full experimental results on real dataset. The x-axis indicates $logK$ where K varies by 5 up to 25 topics and by 25 up to 100 or 150 topics.

gauges the discrepancy between two different constructions of the marginal $p(Z)$: column-sum of $\boldsymbol{A}$ vs applying Bayes' rule. As shown in the results, DC and AP eliminate the discrepancies between two different constructions. Note that the behavior is similar to Legality because marginal construction from an illegal $\boldsymbol{A}$ could be a source of discrepancies.

Synthetic experiments. With the same vocabulary curation, we generate (10,000, 25,000, 50,000) semi-synthetic corpora from the models trained with 50/150/100/100 synthetic anchors for NIPS, NYTimes, Movies and Songs, respectively. We sampled documents with 300 tokens for each dataset from a Dirichlet with symmetric hyperparameters 0.03. The following shows full results on semi-synthetic data with $M = 50,000$ corresponding to real experi-

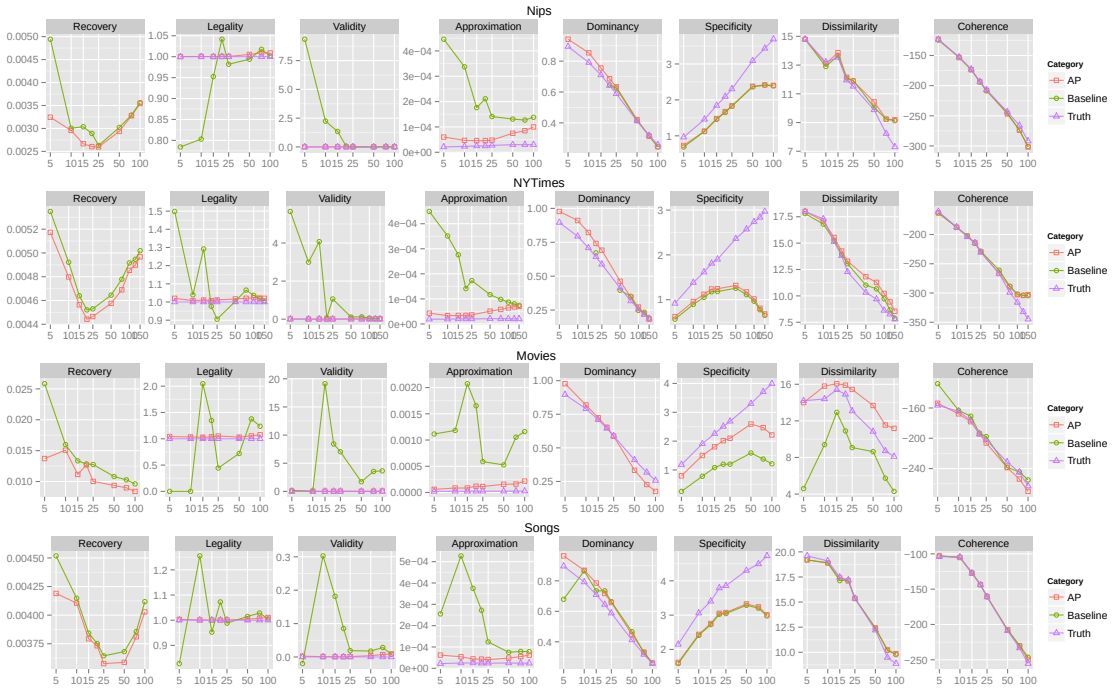

Figure A.2: Full experimental results on synthetic dataset. The x-axis indicates $logK$ where K varies by 5 up to 25 topics and by 25 up to 100 or 150 topics.

ments. You can verify AP and Gibbs are comparable, but the gaps against the Baseline algorithm are lower than what we have seen in real experiments. This is because semi-synthetic data is generated from the model, whereas the real data in practice never precisely follows the model.

We also measure several parametric gaps between the learned matrices and the truth matrices that we used for generating semi-synthetic documents varying two different sizes of documents. We can verify that AP not only learns better topics $\boldsymbol{B}$ and their interactions $\boldsymbol{A}$ than the Baseline algorithm, but also increases the anchor recovery rates. In addition, as we have more documents, the gaps between AP and the Baseline algorithm decrease. We are also showing how well our cluster-example recovery proposal works, and how much the result is consistent to the learned cluster interaction by the panels in the second

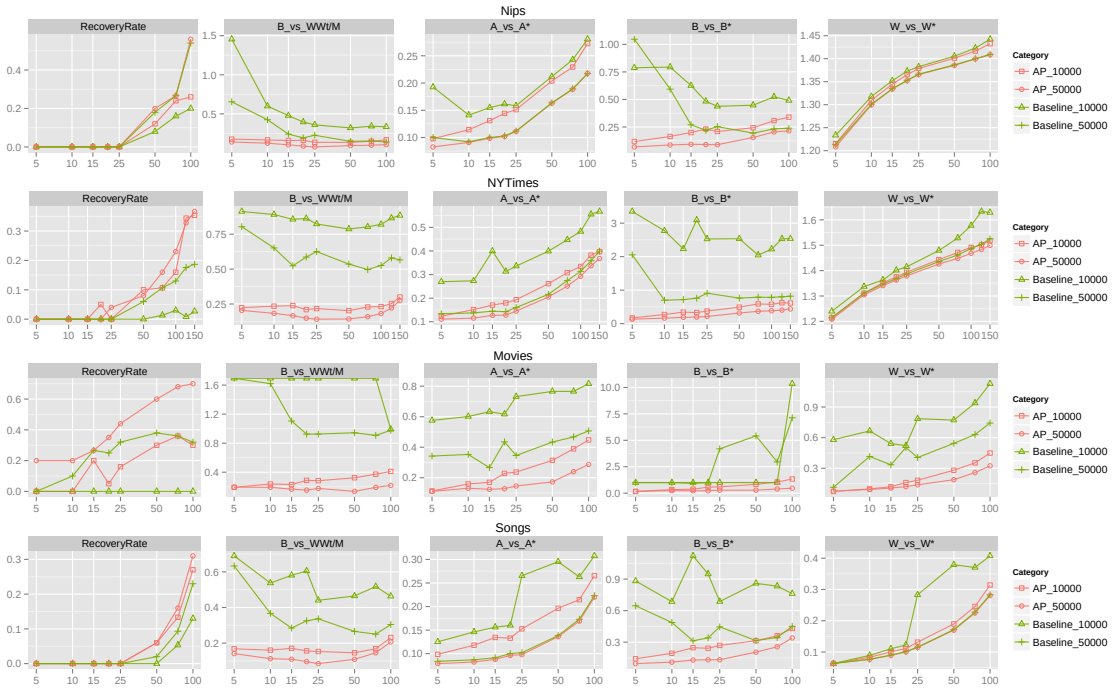

Figure A.3: Gaps between the learned and the truth parameters.

column. (Difference measure is Frobenius norm, but symmetric KL-divergence shows the same behaviors.)

A.5 Analysis of Algorithm

AP Convergence. Figure A.4 shows the actual convergence behavior on NYTimes. For 100 iterations, red solid line and blue dashed line illustrate $\log_{10}(\|\boldsymbol{C}' - \boldsymbol{C}\|_F)$ and the logarithm of the average distance to each of three sets, respectively.[1]

[1]While AP performs an alternating projection, we also keep track of the average of the projected points and the distance to each set per iteration for validation purpose.

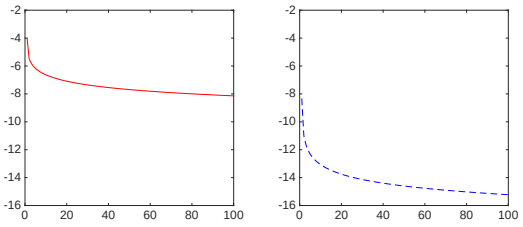

Figure A.4: Locally linear convergence of AP.

How does DC work? Diagonal completion is a special case of matrix completion, which is often solved by minimizing the nuclear norm consistent over matrices with the known data. While first-order methods like [20] generally provide an effective solution to general matrix completion, our alternative algorithm takes advantage of the specific structure of the diagonal completion problem. Suppose we bisects vocabulary into $\mathcal{V}_1$ and $\mathcal{V}_2$. Then it yields a block factorization.

$$\begin{bmatrix} \boldsymbol{C}_{11} & \boldsymbol{C}_{12} \\ \boldsymbol{C}_{21} & \boldsymbol{C}_{22} \end{bmatrix} = \begin{bmatrix} \boldsymbol{B}_1 \\ \boldsymbol{B}_2 \end{bmatrix} A \begin{bmatrix} \boldsymbol{B}_1^T & \boldsymbol{B}_2^T \end{bmatrix} = \begin{bmatrix} \boldsymbol{B}_1 \boldsymbol{A} \boldsymbol{B}_1^T & \boldsymbol{B}_1 \boldsymbol{A} \boldsymbol{B}_2^T \\ \boldsymbol{B}_2 \boldsymbol{A} \boldsymbol{B}_1^T & \boldsymbol{B}_2 \boldsymbol{A} \boldsymbol{B}_2^T \end{bmatrix}$$

Assume that each cluster associates with a sufficient number of objects, being distinguishable only with either $\mathcal{V}_1$ or $\mathcal{V}_2$. It means neither $\boldsymbol{B}_1$ nor $\boldsymbol{B}_2$ should be (nearly) rank deficient. Under this assumption, we find a basis for the range space of $\boldsymbol{C}_{12}$ from the leading singular vectors, then fill the diagonal elements of $\boldsymbol{C}_{11}$ to minimize the distance from $\boldsymbol{B}$ to this subspace. However, in practice, the co-occurrence matrix is contaminated: rather having $\boldsymbol{C}_{12} = \boldsymbol{U}_1 \boldsymbol{\Sigma}_{12} \boldsymbol{V}_2^T$, we actually have $\hat{\boldsymbol{C}}_{12} = \boldsymbol{C}_{12} + \boldsymbol{E}_{12} = \hat{\boldsymbol{U}}_1 \hat{\boldsymbol{\Sigma}}_{12} \hat{\boldsymbol{V}}_2^T$.

Error analysis for DC. In practice, the co-occurrence matrix is contaminated: rather having $\boldsymbol{C}_{12} = \boldsymbol{U}_1 \boldsymbol{\Sigma}_{12} \boldsymbol{V}_2^T$, we actually have $\hat{\boldsymbol{C}}_{12} = \boldsymbol{C}_{12} + \boldsymbol{E}_{12} = \hat{\boldsymbol{U}}_1 \hat{\boldsymbol{\Sigma}}_{12} \hat{\boldsymbol{V}}_2^T$.

Using Wedin's second $\sin\Theta$ theorem (Stewart, V.4.1, Th 4.4) and norm bounds, we have that the maximum sine between the desired space $\boldsymbol{U}_1$ and the computed space $\hat{\boldsymbol{U}}_1$ is $\|\sin\Theta\|_2 \leq \|\boldsymbol{E}_{12}\|/\hat{\sigma}_k \equiv \gamma$ where $\hat{\sigma}_k$ is the smallest retained singular value of the empirical block $\hat{\boldsymbol{C}}_{12}$ and $\|\boldsymbol{E}_{12}\|$ is the magnitude of the difference. This leads to the bounds

$$\min_{\boldsymbol{W}^T\boldsymbol{W}=I} \|\boldsymbol{U}_1\boldsymbol{W} - \hat{\boldsymbol{U}}_1\| \leq \sqrt{\frac{2\|\boldsymbol{E}_{12}\|}{\hat{\sigma}_k}} \equiv \sqrt{2\gamma},$$

$$\min_{\boldsymbol{W}^T\boldsymbol{W}=I} \|e_j^T\boldsymbol{U}_1\boldsymbol{W} - e_j^T\hat{\boldsymbol{U}}_1\| \leq \gamma.$$

Based on the diagonal reconstruction formula, our noisy diagonal will be $\hat{d}_j = \hat{u}_j^T\hat{L}_{*j}/(1 - \hat{u}_j^T\hat{u}_j)$. We write the ratio between the approximate and true values as

$$\frac{\hat{d}_j}{d_j} = \left(\frac{\hat{u}_j^T\hat{\boldsymbol{L}}_{*j}}{u_j^T\boldsymbol{L}_{*j}}\right)\left(\frac{1 - u_j^T u_j}{1 - \hat{u}_j^T\hat{u}_j}\right).$$

The former term in the product can be written $1 + \delta_1$ with

$$|\delta_1| \lesssim \frac{\|\boldsymbol{L}_{*j} - \hat{\boldsymbol{L}}_{*j}\| + \|\boldsymbol{E}_{11}\|\|\boldsymbol{L}_{*j}\|}{u_j^T\boldsymbol{L}_{*j}}$$

and the latter term can be bounded as $1 + \delta_2$ with $|\delta_2| \leq \gamma/(1 - \|\hat{u}_j\|^2)$.

A.6 Related and Future Work

Through this paper, we examine why rectification is necessary, proposing two novel rectification algorithms. Whereas AP enforces every desirable property, Diagonal Completion (DC) enforces only low-rank property on top of joint-stochasticity without requiring positive semi-definiteness. While the results show AP is the appropriate method for our configuration, DC can be also useful

for other tasks based on the co-occurrence matrix. For example, many different embeddings based on the co-occurrence statistics have their own treatment to the diagonal entries, but most of them are based on simple heuristics or trial-and-error approaches rather than strictly enforcing certain mathematical structures. Therefore one might test our DC toward their co-occurrence statistics if low-rank structure is suitable for their tasks.

On the other side, AP finds better anchors as well as performs better inference for learning topics and their interactions. We conjecture that AP's treatment on bursty and rare words smoothen noisy eccentric vertices on the co-occurrence space $\boldsymbol{C}$, making most objects to be well spread out inside the convex rather than being crowded. (The first figure on the main paper shows that rectified space is significantly smaller than the original space in terms of the area, but object vertices in general well spread out through the space, giving better and clear cue of a convex shape.) Therefore, extreme vertices in this smooth space are likely to be truly informative, well summarizing other objects based on the underlying topic interactions.

Arora et al. 2013 (Baseline)	This paper (AP)	Probabilistic LDA (Gibbs)
Pulp Fiction (1994)	Aladdin (1992)	Beauty and the Beast (1991)
Silence of the Lambs (1991)	Toy Story (1995)	Aladdin (1992)
Shawshank Redemption (1994)	Beauty and the Beast (1991)	Mary Poppins (1964)
Forrest Gump (1994)	Babe (1995)	Lion King (1994)
The Fugitive (1993)	Lion King (1994)	Little Mermaid (1989)
Pulp Fiction (1994)	Shrek (2001)	Lord of the Rings I (2001)
Forrest Gump (1994)	Lord of the Rings II (2002)	Lord of the Rings II (2002)
Silence of the Lambs (1991)	Austin Powers (1990)	Matrix (1999)
Shawshank Redemption (1994)	Lord of the Rings I (2001)	Lord of the Rings III (2003)
The Fugitive (1993)	Lord of the Rings III (2003)	Shrek (2001)
Pulp Fiction (1994)	Star Wars V (1980)	Alien (1979)
Silence of the Lambs (1991)	Star Wars IV (1977)	Aliens (1986)
Shawshank Redemption (1994)	StarWars IV (1983)	Blade Runner (1982)
Forrest Gump (1994)	Indiana Jones (1981)	Army of Darkness (1993)
The Fugitive (1993)	Terminator (1984)	Star Trek II (1982)
Pulp Fiction (1994)	Independence Day (1996)	Independence Day (1996)
Shawshank Redemption (1994)	Twister (1996)	The Rock (1996)
Silence of the Lambs (1991)	The Rock (1996)	Mission Impossible (1996)
Forrest Gump (1994)	Mission (1996)	Twister (1996)
Braveheart (1995)	Broken Arrow (1996)	Toy Story (1995)
Pulp Fiction (1994)	Apollo 13 (1995)	Apollo 13 (1995)
Shawshank Redemption (1994)	Dances with Wolves (1990)	The Fugitive (1993)
Silence of the Lambs (1991)	True Lies (1994)	Dances with Wolves (1990)
Forrest Gump (1994)	Pulp Fiction (1994)	Forrest Gump (1994)
The Fugitive (1993)	Batman (1989)	Pulp Fiction (1994)
Pulp Fiction (1994)	Shawshank Redemption (1994)	Maltese Falcon (1941)
Silence of the Lambs (1991)	Matrix (1999)	African Queen (1951)
Shawshank Redemption (1994)	Silence of the Lambs (1991)	Key Largo (1948)
Forrest Gump (1994)	Pulp Fiction (1994)	Double Indemnity (1944)
Star Wars Episode IV (1977)	Lord of the Rings I (2001)	American Graffiti (1973)
Pulp Fiction (1994)	Pulp Fiction (1994)	Remains of the Day (1993)
Silence of the Lambs (1991)	Shawshank Redemption (1994)	Much Ado about Nothing (1993)
Shawshank Redemption (1994)	Usual Suspect (1995)	Copycat (1995)
Forrest Gump (1994)	The Piano (1993)	The Piano (1993)
The Fugitive (1993)	Sense and Sensibility (1995)	What's Eating Gilbert Grape (1993)
Pulp Fiction (1994)	American Beauty (1999)	American Beauty (1999)
Silence of the Lambs (1991)	Sixth Sense (1999)	Sixth Sense (1999)
Shawshank Redemption (1994)	Austin Powers (1999)	Gladiator (2000)
Forrest Gump (1994)	American Pie (1999)	American Pie (1999)
The Fugitive (1993)	Shakespeare in Love (1998)	Fight Club (1999)
Pulp Fiction (1994)	Forrest Gump (1994)	Amelie (2001)
Silence of the Lambs (1991)	Jurassic Park (1983)	Mulholland Drive (2001)
Shawshank Redemption (1994)	Mrs. Doubtfire (1993)	Lost in Translation (2003)
Forrest Gump (1994)	Pretty Woman (1990)	Adaptation (2003)
The Fugitive (1993)	Ghost (1990)	Memento (2000)
Pulp Fiction (1994)	Godfather (1972)	Godfather (1972)
Silence of the Lambs (1991)	One Flew Over the Cuckoo's (1975)	Indiana Jones (1981)
Shawshank Redemption (1994)	Casablanca (1942)	Casablanca (1942)
Forrest Gump (1994)	Godfather II (1974)	One Flew Over the Cuckoo's (1975)
The Fugitive (1993)	Annie Hall (1977)	Star Wars V (1980)

Arora et al. 2013 (Baseline)	**This paper (AP)**	**Probabilistic LDA (Gibbs)**
Pulp Fiction (1994)	Titanic (1997)	Hunt for Red October (1990)
Silence of the Lambs (1991)	The Game (1997)	The Rock (1996)
Shawshank Redemption (1994)	Liar, Liar (1997)	Die Hard 2 (1990)
Forrest Gump (1994)	Chasing Amy (1997)	Face Off (1997)
The Fugitive (1993)	Scream (1996)	Air Force One (1997)
Pulp Fiction (1994)	Tombstone (1993)	Ferris Bueller's Day Off (1986)
Silence of the Lambs (1991)	The Specialist (1994)	Breafast Club (1985)
Shawshank Redemption (1994)	Judge Dredd (1995)	Airplane (1980)
Forrest Gump (1994)	Leon (1994)	Big (1988)
The Fugitive (1993)	Species (1995)	Christmas Story (1983)
Pulp Fiction (1994)	Pulp Fiction (1994)	Tee Departed (2006)
Silence of the Lambs (1991)	Silence of the Lambs (1991)	Casino Royale (2006)
Shawshank Redemption (1994)	Usual Suspects (1995)	Little Miss Sunshine (2006)
Forrest Gump (1994)	12 Monkeys (1995)	V for Vendetta (2006)
The Fugitive (1993)	Seven (1995)	Batman Begins (2005)
Pulp Fiction (1994)	Star Wars IV (1977)	Spider-Man (2002)
Silence of the Lambs (1991)	Star Wars Episode IV (1983)	Ocean's Eleven (2001)
Shawshank Redemption (1994)	Jerry Maguire (1996)	Harry Potter I (2001)
Forrest Gump (1994)	Godfather (1972)	Lord Of the Rings I (2001)
The Fugitive (1993)	Time to Kill (1996)	My Big Fat Greek Wedding (2002)
Pulp Fiction (1994)	Fargo (1996)	Fargo (1996)
Silence of the Lambs (1991)	Leaving Las Vegas (1995)	Shakespeare in Love (1998)
Shawshank Redemption (1994)	Dead Man Walking (1995)	Good Will Hunting (1997)
Forrest Gump (1994)	The Postman (1994)	L. A. Confidential (1997)
The Fugitive (1993)	Trainspotting (1996)	Full Monty (1997)

BIBLIOGRAPHY

[1] Dimitris Achlioptas. Database-friendly random projections. In *SIGMOD*, pages 274–281, 2001.

[2] Loulwah AlSumait, Daniel Barbar, James Gentle, and Carlotta Domeniconi. Topic significance ranking of lda generative models. In *ECML*, 2009.

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

[4] 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.

[5] 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.

[6] F. Arabshahi and A. Anandkumar. Spectral methods for correlated topic models. *AISTATS*, 2017.

[7] Forough Arabshahi and Animashree Anandkumar. Spectral methods for correlated topic models. *AISTATS*, 2017.

[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] Sanjeev Arora, Rong Ge, Frederic Koehler, Tengyu Ma, and Ankur Moitra. Provable algorithms for inference in topic models. In *ICML*, pages 2859–2867, 2016.

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

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

[13] José M Bioucas-Dias and Mário AT Figueiredo. Alternating direction algorithms for constrained sparse regression: Application to hyperspectral unmixing. In *Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS), 2010 2nd Workshop on*, pages 1–4, 2010.

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

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

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

[17] Christos Boutsidis, Michael W. Mahoney, and Petros Drineas. An improved approximation algorithm for the column subset selection problem. In *SODA*, pages 968–977, 2009.

[18] JamesP. Boyle and RichardL. Dykstra. A method for finding projections onto the intersection of convex sets in Hilbert spaces. In *Advances in Order Restricted Statistical Inference*, volume 37 of *Lecture Notes in Statistics*, pages 28–47. Springer New York, 1986.

[19] Mary E Broadbent, Martin Brown, Kevin Penner, I Ipsen, and R Rehman. Subset selection algorithms: Randomized vs. deterministic. *SIAM Undergraduate Research Online*, 3:50–71, 2010.

[20] Jian-Feng Cai, Emmanuel J. Cands, and Zuowei Shen. A singular value thresholding algorithm for matrix completion, 2008.

[21] Jonathan Chang, Jordan Boyd-Graber, Chong Wang, Sean Gerrish, and David M. Blei. Reading tea leaves: How humans interpret topic models. In *NIPS*, 2009.

[22] Jianfei Chen, Jun Zhu, Zi Wang, Xun Zheng, and Bo Zhang. Scalable inference for logistic-normal topic models. In *NIPS*, pages 2445–2453, 2013.

[23] Shuo Chen, J. Moore, D. Turnbull, and T. Joachims. Playlist prediction via metric embedding. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, pages 714–722, 2012.

[24] Jason Chuang, Christopher D. Manning, and Jeffrey Heer. Termite: Visualization techniques for assessing textual topic models. In *International Working Conference on Advanced Visual Interfaces (AVI)*, pages 74–77, 2012.

[25] A. Daniilidis, A. S. Lewis, J. Malick, and H. Sendov. Prox-regularity of spectral functions and spectral sets. *Journal of Convex Analysis*, 15(3):547–560, 2008.

[26] W. Ding, P. Ishwar, and V. Saligrama. Most large topic models are approximately separable. In *2015 Information Theory and Applications Workshop (ITA)*, 2015.

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

[28] John Duchi, Shai Shalev-Shwartz, Yoram Singer, and Tushar Chandra. Efficient projections onto the l 1-ball for learning in high dimensions. In *Proceedings of the 25th international conference on Machine learning*, pages 272–279, 2008.

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

[30] Jacob Eisenstein and Eric Xing. The CMU 2008 political blog corpus. Technical report, CMU, March 2010.

[31] Matt Erlin. Topic modeling, epistemology, and the english and german novel. *Cultural Analytics*, May 2017.

[32] Andrew Goldstone and Ted Underwood. The quiet transformations of literary studies: What thirteen thousand scholars could tell us. *New Literary History*, 45(3), Summer 2014.

[33] Cécile Gomez, H. Le Borgne, Pascal Allemand, Christophe Delacourt, and Patrick Ledru. N-FindR method versus independent component analysis for lithological identification in hyperspectral imagery. *International Journal of Remote Sensing*, 28(23):5315–5338, 2007.

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

[35] Ming Gu and Stanley C. Eisenstat. Efficient algorithms for computing a strong rank-revealing QR factorization. In *SIAM J. Sci Comput*, pages 848–869, 1996.

[36] David Hall, Daniel Jurafsky, and Christopher D. Manning. Studying the history of ideas using topic models. In *EMNLP*, pages 363–371, 2008.

[37] Frank. L. Hitchcock. The expression of a tensor or a polyadic as a sum of products. *Journal of Mathematics and Physics*, 6(1):164–189, 1927.

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

[39] T. Hofmann. Probabilistic latent semantic analysis. In *UAI*, pages 289–296, 1999.

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

[41] 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.

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

[43] William B. Johnson and Joram Lindenstrauss. Extensions of lipschitz mappings into a hilbert space. *Contemporary Mathematics*, 26:189–206, 1984.

[44] Nikos Komodakis, Nikos Paragios, and Georgios Tziritas. Mrf energy minimization and beyond via dual decomposition. *IEEE transactions on pattern analysis and machine intelligence*, 33(3):531–552, 2011.

[45] Da Kuang, Haesun Park, and Chris H. Q. Ding. Symmetric nonnegative matrix factorization for graph clustering. In *SDM*. SIAM / Omnipress, 2012.

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

[47] Abhishek Kumar, Vikas Sindhwani, and Prabhanjan Kambadur. Fast conical hull algorithms for near-separable non-negative matrix factorization. *CoRR*, pages –1–1, 2012.

[48] Simon Lacoste-Julien, Fei Sha, and Michael I. Jordan. DiscLDA: Discriminative learning for dimensionality reduction and classification. In *NIPS*, 2008.

[49] Jey Han Lau, Karl Grieser, David Newman, and Timothy Baldwin. Automatic labelling of topic models. In *HLT*, pages 1536–1545, 2011.

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

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

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

[53] Omer Levy and Yoav Goldberg. Neural word embedding as implicit matrix factorization. In *NIPS*, 2014.

[54] Adrian S. Lewis, D. R. Luke, and Jrme Malick. Local linear convergence for alternating and averaged nonconvex projections. *Foundations of Computational Mathematics*, 9:485–513, 2009.

[55] Guoyin Li and Ting Kei Pong. Douglas–rachford splitting for nonconvex optimization with application to nonconvex feasibility problems. *Mathematical Programming*, 159(1-2):371–401, 2016.

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

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

[58] Michael W. Mahoney and Petros Drineas. CUR matrix decompositions for improved data analysis. *National Academy of Sciences*, 106(3):697–702, 2009.

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

[60] Qiaozhu Mei, Xuehua Shen, and ChengXiang Zhai. Automatic labeling of multinomial topic models. In *KDD*, pages 490–499, 2007.

[61] David Mimno, Matt Hoffman, and David Blei. Sparse stochastic inference for latent dirichlet allocation. 2012.

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

[63] David Mimno, Hanna Wallach, Edmund Talley, Miriam Leenders, and Andrew McCallum. Optimizing semantic coherence in topic models. In *EMNLP*, 2011.

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

[65] Alan Mislove, Bimal Viswanath, Krishna P. Gummadi, and Peter Druschel. You are who you know: Inferring user profiles in Online Social Networks. In *Proceedings of the 3rd ACM International Conference of Web Search and Data Mining (WSDM'10)*, New York, NY, February 2010.

[66] Jos M. P. Nascimento, Student Member, and Jos M. Bioucas Dias. Vertex component analysis: A fast algorithm to unmix hyperspectral data. *IEEE Transactions on Geoscience and Remote Sensing*, pages 898–910, 2005.

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

[68] Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. A three-way model for collective learning on multi-relational data. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, ICML, pages 809–816. ACM, 2011.

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

[70] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. GloVe: Global vectors for word representation. In *EMNLP*, 2014.

[71] Alexander M Rush and Michael Collins. A tutorial on dual decomposition and lagrangian relaxation for inference in natural language processing. *J. Artif. Intell. Res.(JAIR)*, 45:305–362, 2012.

[72] D. Sontag and D. Roy. Complexity of inference in latent Dirichlet allocation. In *NIPS*, pages 1008–1016, 2011.

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

[74] GW Stewart. Four algorithms for the the efficient computation of truncated pivoted QR approximations to a sparse matrix. *Numerische Mathematik*, 83(2):313–323, 1999.

[75] Mark Steyvers and Thomas L Griffiths. Rational analysis as a link between human memory and information retrieval. *The probabilistic mind: Prospects for Bayesian cognitive science*, pages 329–349, 2008.

[76] Edmund M Talley, David Newman, David Mimno, Bruce W Herr II, Hanna M Wallach, Gully A P C Burns, Miriam Leenders, and Andrew McCallum. Database of nih grants using machine-learned categories and graphical clustering. *Nature Methods*, 8(7):443–444, June 2011.

[77] Christian Thurau, Kristian Kersting, and Christian Bauckhage. Yes we can: simplex volume maximization for descriptive web-scale matrix factorization. In *CIKM'10*, pages 1785–1788, 2010.

[78] Ledyard R. Tucker. Some mathematical notes on three-mode factor analysis. *Psychometrika*, 31:279–311, 1966.

[79] L.J.P. van der Maaten and G.E. Hinton. Visualizing high-dimensional data using t-SNE. *JMLR*, 9:2579–2605, Nov 2008.

[80] Hanna Wallach, Iain Murray, Ruslan Salakhutdinov, and David Mimno. Evaluation methods for topic models. In *ICML*, 2009.

[81] Hanna M. Wallach, David M. Mimno, and Andrew McCallum. Rethinking lda: Why priors matter. In *NIPS*. 2009.

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

[83] Yining Wang and Jun Zhu. Spectral methods for supervised topic models. In *NIPS*, 2014.

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

[85] Tianyi Zhou, Jeff A Bilmes, and Carlos Guestrin. Divide-and-conquer learning by anchoring a conical hull. In *Advances in Neural Information Processing Systems 27*, pages 1242–1250, 2014.

[86] Jun Zhu, Amr Ahmed, and Eric P. Xing. MedLDA: Maximum margin supervised topic models for regression and classication. In *ICML*, 2009.