RANDOMIZED NUMERICAL LINEAR ALGEBRA FOR LARGE-SCALE MATRIX DATA

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

Kun Dong

August 2019

RANDOMIZED NUMERICAL LINEAR ALGEBRA FOR LARGE-SCALE MATRIX DATA

Kun Dong, Ph.D.

Cornell University 2019

Your abstract goes here. Make sure it sits inside the brackets. If not, your biosketch page may not be roman numeral iii, as required by the graduate school.

BIOGRAPHICAL SKETCH

Kun Dong was born in Shaoxing, Zhejiang Province, China on September 26, 1992 to Jian Dong and Chamei Sang. Kun became interest in puzzles and mathematics in early elementary school, and participated in mathematics competitions until the end of high school.

In 2008, Kun moved to Newmarket, Ontario, Canada and attended Sir William Mulock Secondary School. He spent the next two years learning the new language and culture.

After graduation, Kun attended the University of California, Los Angeles in 2010 to study applied mathematics. During the summers of 2012 and 2013, he participated in the UCLA Applied Math REU program to work on dynamical system and crime modeling. He received tremendous help and encouragement on his way to graduate school from his REU mentors, Scott McCalla and James von Brecht. In 2014, Kun completed the departmental scholar program in mathematics, earning his B.S. and M.A. concurrently. He graduated Summa Cum Laude, receiving the departmental highest honor in applied mathematics and the Daus memorial award for his achievement in mathematics as an undergraduate.

In 2014, Kun was admitted to the Ph.D. program at Center for Applied Mathematics in Cornell University. He soon started working with Professor David Bindel, who later became his advisor. During the summer of 2017, he interned at the Lawrence Berkeley National Laboratory, where he worked under the mentorship of Professor Lin Lin. For the last three years at Cornell, he was partially supported by a National Science Foundation grant. In September 2019, Kun will move to Seattle and become a research scientist at Facebook, Inc.

This document is dedicated to my parents and wife.

ACKNOWLEDGEMENTS

Your acknowledgements go here. Make sure it sits inside the brackets.

TABLE OF CONTENTS

Biographical Sketch iii
Dedication iv
Acknowledgements v
Table of Contents vi
List of Tables vii
List of Figures viii

1 Introduction 1

2 Network Density of States 2

3 Scalable Gaussian Processes 3

4 Robust Large-Vocabulary Topic Modeling 4
4.1 Abstract 4
4.2 Introduction 4
4.3 Background and Related Work 8
4.4 Low-rank Rectification and Compression 11
4.4.1 Epsilon Non-Negative Rectification (ENN) 11
4.4.2 Proximal Alternating Linearized Minimization Rectification (PALM) 14
4.5 Low-rank Anchor Word Algorithm 15
4.6 Experimental Results 19
4.7 Conclusion 24

5 Weighted K-Means for Electronic Structure Calculation 26

6 Conclusion 27

LIST OF TABLES

LIST OF FIGURES

4.1 Experiment on four datasets. ENN and LR-JSMF mostly agree with AP, whereas PALM has slight inconsistency. The general information of each dataset is above the corresponding row. Recovery, approximation, and runtimes are in $\log_{10}$ scale. Note that ENN and LR-JSMF are almost two orders of magnitude faster than AP. The x-axis indicates the number of clusters K. Lower numbers in y-axis are better except Specificity and Dissimilarity. 19

4.2 As we increase the vocabulary size for four collections, anchor quality and sparsity improve, but running time is stable. The x-axis indicates the vocabulary size N in thousands. Values above 15k will not fit in memory on standard hardware with previous algorithms. 22

4.3 Losses or gains in topic words depending on the vocabulary size. Each row represents a topic from the NeurIPS dataset, with the top 6 topical words shown in the middle column. The red and green cells denote topic words that are lost or gained by shifting the vocabulary size from the default size 5000, respectively. The intensities of the colors indicate the words' contributions towards the specific topic. 24

CHAPTER 1

INTRODUCTION

CHAPTER 2

NETWORK DENSITY OF STATES

CHAPTER 3

SCALABLE GAUSSIAN PROCESSES

CHAPTER 4

ROBUST LARGE-VOCABULARY TOPIC MODELING

4.1 Abstract

Across many data domains, co-occurrence statistics about the joint appearance of objects are powerfully informative. By transforming unsupervised learning problems into decompositions of co-occurrence statistics, spectral inference provides transparent algorithms and optimality guarantees for non-linear dimensionality reduction or latent topic analysis. As object vocabularies grow, however, it becomes rapidly more expensive to store and run inference algorithms on co-occurrence statistics. Current rectification techniques, which preprocess real data in order to overcome model-data mismatch, are even more expensive, as they require iterative projections that destroy sparsity of the co-occurrence data. In this paper, we propose novel approaches that can simultaneously compress and rectify the co-occurrence statistics, scaling gracefully with the size of vocabulary and the dimension of latent space. We also present new algorithms that are capable of learning latent variables from the compressed statistics without losing visible precision, and verify that they perform comparably to previous approaches on both textual and non-textual data.

4.2 Introduction

Understanding the low-dimensional geometry of noisy and complex data is a fundamental problem of unsupervised learning. Probabilistic models explain data generation processes in terms of low-dimensional latent variables. Inferring a posterior distribution for these latent variables provides us with a compact representation for various exploratory analyses and downstream tasks. However, exact inference is often intractable due to entangled interactions between the latent variables [9, 2, 1, 29]. Variational inference transforms the posterior approximation into an optimization problem over simpler distributions with independent parameters [19, 31, 10], while Markov Chain Monte Carlo enables users to sample from the desired posterior distribution [26, 27, 30]. However, these likelihood-based methods require numerous iterations without any guarantee beyond local improvement at each step [20].

When the data consists of collections of discrete objects, co-occurrence statistics summarize interactions between objects. Collaborative filtering learns low-dimensional representations of individual items, which are useful for recommendation systems, by explicitly decomposing the co -occurrence of items that are jointly consumed by certain users [22, 25]. Word-vector models learn low -dimensional embeddings of individual words, which encode useful linguistic biases for neural networks, by implicitly decomposing the co-occurrence of words that appear together [28, 24]. If co -occurrence provides a rich enough set of unbiased moments about an underlying generative model, spectral methods can provably learn posterior configurations from co-occurrence information alone, without iterating through individual training examples [7, 5, 17, 3].

However, two major limitations hinder users from taking advantage of spectral inference based on co-occurrence. First, the second-order co -occurrence matrix grows quadratically in the size of the objects (e.g. items, words, products). Pruning these objects is an option, but for a retailer selling millions of products, a low-dimensional representation of a small subset of the products is inadequate. Second, inference quality is poor in real data that does not necessarily follow our computational model. Whereas likelihood-based methods have an intrinsic capability to fit the data to the model despite their mismatch, sample noise can destroy the performance of spectral methods even if the data is synthesized from the model [20]. *Rectification*, a process of projecting empirical co-occurrence onto a set consistent with the geometry of the model, can improve the performance of spectral inference in the face of model mismatch [22]. But running multiple projections dominates overall inference cost even when the vocabulary is small. In addition, the rectification process makes the co-occurrence dense, exacerbating storage costs when dealing with large vocabularies.

In this paper, we propose the Epsilon Non-Negative rectification (ENN) and the Low-rank Anchor Word algorithm (LAW). Given a vocabulary of N objects and the user-specified latent dimension K, ENN simultaneously compresses and rectifies the co-occurrence matrix $\boldsymbol{C} \in \mathbb{R}^{N \times N}$ into $\boldsymbol{Y}\boldsymbol{Y}^T$ with $\boldsymbol{Y} \in \mathbb{R}^{N \times K}$. Each entry of the decompression $(\boldsymbol{Y}\boldsymbol{Y}^T)_{ij}$ tightly approximates the rectified co-occurrence $\boldsymbol{C}^*_{ij}$ but is allowed to be a tiny negative value above $-\epsilon$. Then LAW learns the latent clusters (e.g., topics of documents or genres of items) and their correlations provided only with $\boldsymbol{Y}$, guaranteeing the same performance as running the original Anchor Word algorithm on $\boldsymbol{C}^*$ if $\boldsymbol{Y}\boldsymbol{Y}^T \geq 0$. In contrast, we also formulate the Proximal Alternating Linearized Minimization rectification algorithm (PALM) that approximates the rectified co -occurrence $\boldsymbol{C}^*$ by $\boldsymbol{Y}\boldsymbol{Y}^T$ with $\boldsymbol{Y} \geq 0$. While the non-negativity of $\boldsymbol{Y}$ in the PALM approach allows LAW to perform exact inference, the fact that PALM enforces more constraints than ENN means that PALM also provides a less faithful approximation to the original co-occurrence data. Our experiments on various textual and non-textual datasets show that ENN learns a high-quality factor $\boldsymbol{Y}$ for which LAW provides results of quality comparable to those based on the full co-occurrence $\boldsymbol{C}^*$; in contrast, while PALM works comparably in some settings, in others there is a visible loss of accuracy.

We also adopt a randomized algorithm that constructs a low-rank approximation of the full co-occurrence $\boldsymbol{C}$ directly from the raw data. While PALM requires the full co-occurrence, ENN can work directly with the low-rank initialization, eliminating the need to ever store a full co-occurrence matrix. Note also that the second-order methods rely on the *separability assumption*,[1] which has been another criticism in theory despite their superior performance in practice [23]. Our analyses show that models based on large vocabularies find more separable anchor objects, learning stable latent clusters without much sensitivity to sample noise. Overall, we complete a robust and scalable pipeline: that efficiently performs quality posterior inference for unsupervised learning from co-occurrence information within time and space complexity linear in N.

The major contributions of our paper are:

- We introduce two efficient rectifications, ENN and PALM, that compress quadratic and noisy co-occurrence information on the fly with a linear space rectified representation.
- We develop a low-rank algorithm (LAW) for anchor-word-based topic modeling that works directly on the compressed rectified representations and provides near-exact performance.
- We propose a robust and scalable pipeline, LR-JSMF, that learns topic models

[1] Each cluster has at least one anchor object which dedicates exclusively to that cluster, nothing else.

with a small number of passes directly over the data. This new pipeline scales to large vocabularies that were previously intractable for spectral inference, and offers a ~100 × speedup over previous methods for general data sets.

4.3 Background and Related Work

Our new algorithms build on the the **Joint-Stochastic Matrix Factorization (JSMF)** framework [22], which we now describe. Let $\boldsymbol{H} \in \mathbb{R}^{N \times M}$ be the object-example matrix whose m-th column vector $\boldsymbol{h}_m$ counts the occurrences of each of the N objects in the vocabulary in example m. We denote the total number of objects in example m by n_m. Given a user-specified number of clusters K, we seek to learn an object-cluster matrix $\boldsymbol{B} \in \mathbb{R}^{N \times K}$ where $\boldsymbol{B}_{ik}$ is the conditional probability of observing object i given latent cluster k. Instead of learning $\boldsymbol{B}$ directly from the sparse and noisy observations $\boldsymbol{H}$, JSMF begins with constructing the joint-stochastic co-occurrence $\boldsymbol{C} \in \mathbb{R}^{N \times N}$ by

$$\boldsymbol{C} = \hat{\boldsymbol{H}}\hat{\boldsymbol{H}}^T - \hat{\boldsymbol{H}}_{diag}, \quad \hat{\boldsymbol{h}}_m = \frac{\boldsymbol{h}_m}{\sqrt{n_m(n_m - 1)M}}, \quad \hat{\boldsymbol{H}}_{diag} = diag\left(\sum_{m=1}^{M} \frac{\boldsymbol{h}_m}{n_m(n_m - 1)M}\right). \tag{4.1}$$

Then the original Anchor Word algorithm decomposes $\boldsymbol{C}$ into $\boldsymbol{B}\boldsymbol{A}\boldsymbol{B}^T$ by Algorithm 1, where $\boldsymbol{A} \in \mathbb{R}^{K \times K}$ is the cluster correlation matrix, whose entry $\boldsymbol{A}_{kl}$ captures the joint probability between two latent clusters k and l.[2] In the limit, using data generated according to the correct probabilistic models, $\boldsymbol{A}$ must agree with the second-moment of the cluster proportions, which is given as a Bayesian prior in the models.

As with other spectral algorithms for latent variable models [17, 4], the de-

[2] $\boldsymbol{C}$ is proven to be a by far more robust estimator than $\boldsymbol{H}$ in [6]. But actual construction of $\boldsymbol{C}$ in [7] is slightly misleading without dividing by M. We report the full equations in this paper.

Algorithm 1: Anchor Word algorithm (AW)

```
Input: Object co-occurrence $\boldsymbol{C} \in \mathbb{R}^{N \times N}$
       The number of clusters $K$
Output: Anchor objects $\boldsymbol{S} = \{s_1, ..., s_K\}$
        Latent clusters $\boldsymbol{B} \in \mathbb{R}^{N \times K}$
        Cluster correlations $\boldsymbol{A} \in \mathbb{R}^{K \times K}$
begin
    $L_1$-normalize the rows of $\boldsymbol{C}$ to form $\overline{\boldsymbol{C}}$.
    Find $\boldsymbol{S}$ via the column pivoted QR on $\overline{\boldsymbol{C}}^T$.
    Find $\breve{\boldsymbol{B}}$ with $\breve{\boldsymbol{B}}_{ki} = p(\text{cluster } k \mid \text{object } i)$ by solving $N$
      simplex-constrained least squares in parallel to minimize
      $\|\overline{\boldsymbol{C}} - \breve{\boldsymbol{B}}^T \overline{\boldsymbol{C}}_{\boldsymbol{S}*}\|_F$.
    Recover $\boldsymbol{B}$ from $\breve{\boldsymbol{B}}$ by the Bayes' rule.
    Recover $\boldsymbol{A}$ by $\boldsymbol{B}_{\boldsymbol{S}*}^{-1} \boldsymbol{C}_{\boldsymbol{SS}} \boldsymbol{B}_{\boldsymbol{S}*}^{-1}$.
end
```

Algorithm 2: Rectified AW algorithm (RAW)

```
Input/Output: Same as Algorithm 1
begin
    $\boldsymbol{C}_0 \leftarrow \boldsymbol{C}$
    repeat with $t = 0, 1, 2, ...$
        $(\boldsymbol{U}, \boldsymbol{\Lambda}_K) \leftarrow \text{Truncated-Eig}(\boldsymbol{C}_t, K)$
        $\boldsymbol{\Lambda}_K^+ \leftarrow \max(\boldsymbol{\Lambda}_K, 0)$
        $\boldsymbol{C}^{\mathcal{PSD}_K} \leftarrow \boldsymbol{U} \boldsymbol{\Lambda}_K^+ \boldsymbol{U}^T$
        $\boldsymbol{C}^{\mathcal{NOR}} \leftarrow \boldsymbol{C}^{\mathcal{PSD}_K} + \frac{1 - \sum_{ij} \boldsymbol{C}_{ij}^{\mathcal{PSD}_K}}{N^2} \boldsymbol{e}\boldsymbol{e}^T$
        $\boldsymbol{C}^{\mathcal{NN}} \leftarrow \max(\boldsymbol{C}^{\mathcal{NOR}}, 0)$
        $\boldsymbol{C}_{t+1} \leftarrow \boldsymbol{C}^{\mathcal{NN}}$
    until converging to a certain $\boldsymbol{C}^*$
    $(\boldsymbol{S}, \boldsymbol{B}, \boldsymbol{A}) \leftarrow \text{AW}(\boldsymbol{C}^*, K)$   (Algorithm 1)
end
```

composition as described so far may fail to learn high-quality clusters due to *model-data mismatch* [20]. Under the probabilistic model assumed to generate the data, the expected value of the co-occurrence should not only be normalized to sum to one ($\mathcal{NOR}$) and be entry-wise non-negative ($\mathcal{NN}$), but it should also be positive semi-definite with rank equal to the number of clusters K ($\mathcal{PSD}_K$) [22].

However, the empirical $\boldsymbol{C}$ from real data is often indefinite and full-rank due to sample noise[3] and the unbiased construction of $\boldsymbol{C}$ in Equation (4.1), which penalizes all diagonal entries. The **Rectified Anchor Word (RAW)** algorithm has an additional rectification step that enforces that $\boldsymbol{C}$ should enjoy the expected structures before running the main algorithm. In [22], an Alternating Projection (AP) rectification as given in Algorithm 2 is used to overcome the gap between the underlying assumptions of our models and the actual data.

Rectification is also important for addressing the issue of *outlier bias*. Real data often exhibits rare objects that are only present in a few examples. The corresponding co-occurrence of these objects are inevitably sparse with large variance, but the greedy anchor selection favors choosing these outliers due to L_1 normalization of $\boldsymbol{C}$. Previous work tried to bypass this problem by oversampling clusters by the number of outliers under some additional identifiability assumptions [14]. This approach is not always feasible, especially for a large vocabulary that introduces many low frequency objects. When synonyms and short documents cause undesirable sparsity to Latent Semantic Analysis [21], projection onto the leading eigen-subspace blurs sparse co-occurrences. Similarly, $\mathcal{PSD}_K$-projection turns out to significantly reduce outlier bias, and the remaining projections are useful for maintaining the probabilistic structures of $\boldsymbol{C}$, which then allow users to recover $\boldsymbol{B}$ and $\boldsymbol{A}$ in Algorithm 1.

Handling a *large vocabulary* is another major challenge for spectral methods. Even if we limit our focus only to second-order models, the space complexity of RAW is already $O(N^2)$, growing rapidly with increasing vocabulary. We are unable to exploit the high sparsity of $\boldsymbol{C}$ as a single iteration of the AP-rectification

[3] Rectification still improves the quality of clusters on the synthetic data that is generated from our models.

makes $\boldsymbol{C}$ significantly denser. The three projections in AP-rectification and the rest of the anchor word algorithm in Algorithm 1 have time complexities of $O(N^2K)$, $O(N^2)$, $O(N^2)$ and $O(N^2K)$, respectively, and so pose a difficulty when scaling to a large vocabulary size N. On the other hand, the *separability assumption* is crucial for the second-order models, and while there has been a line of research that tries to relax this assumption [8, 18], it has been formally shown that most topic models are indeed separable if their vocabulary sizes are sufficiently larger than the number of clusters [13], again emphasizing the urgency of an approach with better time and space scaling in the vocabulary size.

4.4 Low-rank Rectification and Compression

The rectified co-occurrence $\boldsymbol{C}^*$ in Section 4.3 must be of rank K and positive semidefinite, hinting at an opportunity to represent it in terms of an outer product $\boldsymbol{Y}\boldsymbol{Y}^T$ for some $\boldsymbol{Y} \in \mathbb{R}^{N \times K}$. One idea for achieving this structure is to use a low-rank representation $\boldsymbol{C}_t = \boldsymbol{Y}_t\boldsymbol{Y}_t$ throughout the rectification in Algorithm 2. Another way to obtain this structure is to directly minimize $\|\boldsymbol{C} - \boldsymbol{Y}\boldsymbol{Y}^T\|_F$ with the necessary constraints. In this section, we propose two new algorithms to simultaneously compress and rectify the input by representing $\boldsymbol{C} \in \mathbb{R}^{N \times N}$ by a low-rank outer product $\boldsymbol{Y}\boldsymbol{Y}^T$.

4.4.1 Epsilon Non-Negative Rectification (ENN)

The alternating projection iteration in Algorithm 2 produces low-rank semidefinite intermediate matrices in factored form at each iteration. By construc-

```
Algorithm 3: ENN-rectification (ENN)
Input: Object co-occurrence $\boldsymbol{C} \in \mathbb{R}^{N \times N}$
       The number of clusters $K$
Output: Rectified compression $\boldsymbol{Y} \in \mathbb{R}^{N \times K}$
begin
    $\boldsymbol{E} \leftarrow \boldsymbol{0} \in \mathbb{R}^{N \times N}$   (sparse format)
    $\boldsymbol{C}^{op} : \boldsymbol{x} \rightarrow \boldsymbol{C}\boldsymbol{x}$   (Implicit operator)
    repeat with $t = 0, 1, 2, ...$
        $(\boldsymbol{U}, \boldsymbol{\Lambda}_K) \leftarrow$ Truncated-Eig$(\boldsymbol{C}^{op}, K)$
        $\boldsymbol{\Lambda}_K^+ \leftarrow \max(\boldsymbol{\Lambda}_K, 0)$
        $\boldsymbol{Y} \leftarrow \boldsymbol{U}(\boldsymbol{\Lambda}_K^+)^{1/2}$
        $\boldsymbol{E}_{ij} \leftarrow \max(-\boldsymbol{Y}_{i*}\boldsymbol{Y}_{j*}^T, 0)$
        $r \leftarrow (1 - \|\boldsymbol{Y}^T\boldsymbol{e}\|_2^2 - \sum_{ij} \boldsymbol{E}_{ij})/N^2$
        $\boldsymbol{C}^{op} : \boldsymbol{x} \rightarrow \boldsymbol{Y}(\boldsymbol{Y}^T\boldsymbol{x}) + \boldsymbol{E}\boldsymbol{x} + r(\boldsymbol{e}^T\boldsymbol{x})\boldsymbol{e}$
    until $\boldsymbol{E}$ converges
end
```

tion, the positive semi-definite projection ($\mathcal{PSD}_K$) and the normalization projection ($\mathcal{NOR}$) produce positive semi-definite matrices of rank K and $K + 1$, respectively. Unfortunately, the final projection to enforce elementwise non-negativity ($\mathcal{NN}$) destroys this low rank structure. However, the $\mathcal{NN}$ projection only makes significant changes to a few elements; that is, the output of the $\mathcal{NN}$ projection at step t is nearly rank $K + 1$ plus a sparse correction $\boldsymbol{E}_t$. The Epsilon Non-Negative Rectification algorithm (Algorithm 3) has the same structure as Algorithm 2, but with the key difference that it returns a sparse-plus-low-rank representation of the $\mathcal{NN}$ projection rather than materializing a dense representation. Matrix-vector products with this sparse-plus-low-rank representation require $O(NK + \text{nnz}(\boldsymbol{E}_t))$ time, and $O(K)$ such matrix-vector products can be used in a Lanczos eigen-solver to compute the truncated eigen-decomposition at the start of the next iteration.

Maintaining a sparse correction matrix $\boldsymbol{E}_t$ at each step lets the ENN approach

Algorithm 4: PALM-rectification (PALM)

```
Input: Object co-occurrence $\boldsymbol{C} \in \mathbb{R}^{N\times N}$
       The number of clusters $K$
Output: Rectified compression $\boldsymbol{Y} \in \mathbb{R}^{N\times K}$
begin
    $(\boldsymbol{V}, \boldsymbol{D}) \leftarrow$ Truncated-Eig$(\boldsymbol{C}, K)$
    $(\boldsymbol{X}_0, \boldsymbol{Y}_0) \leftarrow (\boldsymbol{V}\sqrt{\boldsymbol{D}}, \boldsymbol{V}\sqrt{\boldsymbol{D}})$
    repeat
        $c_t \leftarrow \gamma L_1(\boldsymbol{Y}_t)$
        $\boldsymbol{X}'_{t+1} \leftarrow \boldsymbol{X}_t - (1/c_t)\nabla_{\boldsymbol{X}} J(\boldsymbol{X}_t, \boldsymbol{Y}_t)$
        $\boldsymbol{X}_{t+1} \leftarrow \max(\boldsymbol{X}'_{t+1}, 0)$
        $d_t \leftarrow \gamma L_2(\boldsymbol{X}_{t+1})$
        $\boldsymbol{Y}'_{t+1} \leftarrow \boldsymbol{Y}_t - (1/d_t)\nabla_{\boldsymbol{Y}} J(\boldsymbol{X}_{t+1}, \boldsymbol{Y}_t)$
        $\boldsymbol{Y}_{t+1} \leftarrow \max(\boldsymbol{Y}'_{t+1}, 0)$
    until $\boldsymbol{Y}$ converges
end
```

avoid the storage overheads of the original alternating projection algorithm. To overcome the quadratic time cost at each iteration, though, we need to avoid explicitly computing every element of the intermediate $\boldsymbol{Y}\boldsymbol{Y}^T$ in the course of the $\mathcal{NN}$ projection. However, we can bound the magnitude of many elements of $\boldsymbol{Y}\boldsymbol{Y}^T$ by the Cauchy-Schwartz inequality: $|\boldsymbol{C}_{ij}| \leq \|\boldsymbol{y}_i\|_2\|\boldsymbol{y}_j\|_2$ where $\boldsymbol{y}_i$ and $\boldsymbol{y}_j$ denote columns of $\boldsymbol{Y}^T$. Let $\boldsymbol{I}$ denote the index set indices $\{i : \|\boldsymbol{y}_i\|_2^2 > \epsilon\} \subseteq [N]$ for given ϵ; then every large entry of $\boldsymbol{C}$ belongs to either $\boldsymbol{Y}_{\boldsymbol{I}*}\boldsymbol{Y}^T$ or $\boldsymbol{Y}(\boldsymbol{Y}^T)_{*\boldsymbol{I}}$. As $\boldsymbol{C}$ is symmetric, checking the negative entries in $\boldsymbol{Y}_{\boldsymbol{I}*}\boldsymbol{Y}^T$ is sufficient to find a symmetric correction $\boldsymbol{E}$ that guarantees $\boldsymbol{Y}\boldsymbol{Y}^T + \boldsymbol{E} \geq -\epsilon$. We refer to this property as *Epsilon Non-Negativity*: ϵ balances the trade-off between the effect of leaving small negative entries versus increasing the size of $\boldsymbol{I}$ to look up. We limit $|\boldsymbol{I}|$ to be $\mathcal{O}(K)$ based on the common sampling complexity of a suitable set of rows for a near-optimal rank-K approximation[4].

[4] This choice is standard in literature on low-rank approximation via column subset selection.

4.4.2 Proximal Alternating Linearized Minimization Rectification (PALM)

To avoid small negative entries, we investigate another rectified compression algorithm that directly minimizes $\|\boldsymbol{C} - \boldsymbol{Y}\boldsymbol{Y}^T\|_F$ subject to the stronger $\mathcal{NN}$-constraint $\boldsymbol{Y} \geq 0$ and the usual $\mathcal{NOR}$-constraint $\|\boldsymbol{Y}^T\boldsymbol{e}\|_2 = 1$. Concretely, we try to

$$\text{minimize } \; J(\boldsymbol{X}, \boldsymbol{Y}) := \frac{1}{2}\|\boldsymbol{C} - \boldsymbol{X}\boldsymbol{Y}^T\|_F^2 + \frac{s}{2}\|\boldsymbol{X} - \boldsymbol{Y}\|_F^2 \quad \text{subject to } \; \boldsymbol{X} \geq 0, \boldsymbol{Y} \geq 0. \tag{4.2}$$

$\mathcal{PSD}_K$- and $\mathcal{NOR}$-constraints are implicitly satisfied by jointly minimizing the two terms in the objective function J, whereas $\mathcal{NN}$-constraint is explicit in the formulation. Thus we can apply the Proximal Alternating Linearized Minimization [11] for learning $\boldsymbol{Y}$ given $\boldsymbol{C}$; the relevant proximal operator is $\mathcal{NN}$ projection of $\boldsymbol{Y}$, which takes $O(NK)$ at most.

Note that J is semi-algebraic (as it is a real polynomial) with two partial derivatives: $\nabla_{\boldsymbol{X}} J = (\boldsymbol{X}\boldsymbol{Y}^T - \boldsymbol{C})\boldsymbol{Y} + s(\boldsymbol{X} - \boldsymbol{Y})$ and $\nabla_{\boldsymbol{Y}} J = (\boldsymbol{Y}\boldsymbol{X}^T - \boldsymbol{C})\boldsymbol{X} + s(\boldsymbol{Y} - \boldsymbol{X})$. So, the following lemma guarantees the global convergence.

Lemma 1. *For any fixed $\boldsymbol{Y}$, $\nabla_{\boldsymbol{X}} J(\boldsymbol{X}, \boldsymbol{Y})$ is globally Lipschitz continuous with the moduli $L_1(\boldsymbol{Y}) = \|\boldsymbol{Y}^T\boldsymbol{Y} + s\boldsymbol{I}_K\|_2$. So is $\nabla_{\boldsymbol{Y}} J(\boldsymbol{X}, \boldsymbol{Y})$ given any fixed $\boldsymbol{X}$ with $L_2(\boldsymbol{X}) = \|\boldsymbol{X}^T\boldsymbol{X} + s\boldsymbol{I}_K\|_2$.*

Proof.

$$\begin{aligned} &\|\nabla_{\boldsymbol{X}} J(\boldsymbol{X}, \boldsymbol{Y}) - \nabla_{\boldsymbol{X}} J(\boldsymbol{X}', \boldsymbol{Y})\|_F \\ &= \|(\boldsymbol{Y}^T\boldsymbol{Y} + s\boldsymbol{I}_K)(\boldsymbol{X} - \boldsymbol{X}')\|_F \end{aligned}$$

$$\leq \|\boldsymbol{Y}^T\boldsymbol{Y} + s\boldsymbol{I}_k\|_2 \cdot \|\boldsymbol{X} - \boldsymbol{X}'\|_F$$

The proof is symmetric for the other case with $L_2(\boldsymbol{X}) = \|\boldsymbol{X}^T\boldsymbol{X} + s\boldsymbol{I}_K\|_2$. □

Algorithm 4 shows the PALM-rectification with the adaptive control of the learning rates based on the tight 2-norm Lipschitz modulis at each step t.

4.5 Low-rank Anchor Word Algorithm

The output for both methods in §4.4 is a compressed co -occurrence matrix $\boldsymbol{C} = \boldsymbol{Y}\boldsymbol{Y}^T$. In this section, we present the **Low-rank Anchor Word algorithm (LAW)** that reduces the time complexity of finding anchor objects from $O(N^2K)$ to $O(NK^2)$ by taking advantage of this form. We note that LAW applies whenever $\boldsymbol{C}$ is in a low-rank representation, which does not have to be derived from our methods. Moreover, it is exact for non-negative $\boldsymbol{C}$, but it is robust in practice when we allow small negative entries in $\boldsymbol{C}$, as in the case with ENN.

The first step is to L_1-normalize the rows of $\boldsymbol{C}$. Given $\boldsymbol{C} \geq 0$, the L_1-norm of each row is simply the sum of all its entries, so we can calculate the row norms by $\boldsymbol{d} = \boldsymbol{Y}(\boldsymbol{Y}^T\boldsymbol{e})$. To obtain the normalized $\boldsymbol{C}$, we simply scale the rows of $\boldsymbol{Y}$, and $\overline{\boldsymbol{C}} = (\text{diag}(\boldsymbol{d})^{-1}\boldsymbol{Y})\boldsymbol{Y}^T = \overline{\boldsymbol{Y}}\boldsymbol{Y}^T$. These steps cost $O(NK)$.

Next, we need to apply column pivoted QR to $\overline{\boldsymbol{C}}^T$ in order to identify the pivots as our anchor objects $\boldsymbol{S}$. By taking the QR decomposition $\boldsymbol{Y} = \boldsymbol{Q}\boldsymbol{R}$, $\overline{\boldsymbol{C}}^T$ can be further transformed into $\boldsymbol{Q}\boldsymbol{R}\overline{\boldsymbol{Y}}^T$. Notice that $\overline{\boldsymbol{C}}^T$ is an orthogonal embedding of $\boldsymbol{X}^T = \boldsymbol{R}\overline{\boldsymbol{Y}}^T$ onto a higher-dimensional space, which preserves the column L_2-norms. Lemma 2 shows that column pivoted QR on $\overline{\boldsymbol{C}}^T$ and on $\boldsymbol{R}\overline{\boldsymbol{Y}}^T$ are equivalent, which allows us to lower the computation cost from $O(N^2K)$ to $O(NK^2)$.

Algorithm 5: Low-rank AW (LAW)

```
Input: Object co-occurrence $C = YY^T$
Output: Anchor objects $S = \{s_1, ..., s_K\}$
        Latent clusters $B \in \mathbb{R}^{N \times K}$
        Cluster correlations $A \in \mathbb{R}^{K \times K}$
begin
    Calculate row sums $d = Y(Y^T e)$.
    Normalize $\overline{Y} \leftarrow \mathrm{diag}(d)^{-1} Y$.
    Compute QR decomposition of $Y = QR$.
    Form $X = \overline{Y} R^T$.
    Select $S$ using column pivoted QR on $X^T$.
    Solve $n$ simplex-constrained least square problems to minimize
      $\|X - \breve{B} X_{S*}\|_F$.
    Recover $B$ from $\breve{B}$ using Bayes' rule.
    Recover $A = B_{S*}^{-1} Y_{S*} Y_{S*}^T B_{S*}^{-1}$.
end
```

Algorithm 6: Low-rank JSMF (LR-JSMF)

```
Input: Raw object-example $H \in \mathbb{R}^{N \times M}$
Output: Anchor objects $S = \{s_1, ..., s_K\}$
        Latent clusters $B \in \mathbb{R}^{N \times K}$
        Cluster correlations $A \in \mathbb{R}^{K \times K}$
begin
    Get $\hat{H}, \hat{H}_{diag}$ from $H$ by (4.1).
    $C_{op} : x \to \hat{H}(\hat{H}^T x) - \hat{H}_{diag} x$
    $(U, \Lambda_K) \leftarrow$ Randomized-Eig($C_{op}, K$)
    Initialize ENN with $U, \Lambda_K$.
    $Y \leftarrow$ ENN-rectification
    $(S, B, A) \leftarrow$LAW($Y$)   (Algorithm 5)
end
```

Lemma 2. *Let $\boldsymbol{S}$ be the set of pivots that have been selected by column pivoted QR on $\overline{\boldsymbol{C}}^T = \boldsymbol{Q}\boldsymbol{X}^T$. Given the QR decomposition, $\overline{\boldsymbol{X}}^T_{\boldsymbol{S}*} = \boldsymbol{P}\boldsymbol{T}$, then $\overline{\boldsymbol{C}}^T_{\boldsymbol{S}*} = (\boldsymbol{Q}\boldsymbol{P})\boldsymbol{T}$ is the corresponding QR decomposition for the columns of $\overline{\boldsymbol{C}}$. For any remaining row $i \in [N] \setminus S$,*

$$\|(I - \boldsymbol{P}\boldsymbol{P}^T)\overline{\boldsymbol{X}}^T_{i*}\|_2 = \|(I - (\boldsymbol{Q}\boldsymbol{P})(\boldsymbol{Q}\boldsymbol{P})^T)\overline{\boldsymbol{C}}^T_{i*}\|_2 \tag{4.3}$$

Therefore, the next column pivot is identical for $\overline{\boldsymbol{C}}^T$ and $\overline{\boldsymbol{X}}^T$. By induction, column pivoted QR on $\overline{\boldsymbol{C}}^T$ and $\overline{\boldsymbol{X}}^T$ return the same pivots.

Proof. Because both $\boldsymbol{Q}$ and $\boldsymbol{P}$ have orthonormal columns,

$$(\boldsymbol{Q}\boldsymbol{P})^T(\boldsymbol{Q}\boldsymbol{P}) = \boldsymbol{P}^T(\boldsymbol{Q}^T\boldsymbol{Q})\boldsymbol{P} = \boldsymbol{P}^T\boldsymbol{P} = I \tag{4.4}$$

Thus, $\boldsymbol{Q}\boldsymbol{P}$ and $\boldsymbol{T}$ forms the QR decomposition of $\overline{\boldsymbol{C}}^T_{\boldsymbol{S}*}$. The residual of a remaining column $i \in [N] \setminus S$ is $(I - (\boldsymbol{Q}\boldsymbol{P})(\boldsymbol{Q}\boldsymbol{P})^T)\overline{\boldsymbol{C}}^T_{i*}$ and $(I - \boldsymbol{P}\boldsymbol{P}^T)\overline{\boldsymbol{X}}^T_{i*}$ for $\overline{\boldsymbol{C}}^T$ and $\overline{\boldsymbol{X}}^T$, respectively. Simplify the former gives us

$$\begin{aligned}
&(I - (\boldsymbol{Q}\boldsymbol{P})(\boldsymbol{Q}\boldsymbol{P})^T)\overline{\boldsymbol{C}}^T_{i*} \\
&= (I - (\boldsymbol{Q}\boldsymbol{P})(\boldsymbol{Q}\boldsymbol{P})^T)\boldsymbol{Q}\overline{\boldsymbol{X}}^T_{i*} \\
&= \boldsymbol{Q}\overline{\boldsymbol{X}}^T_{i*} - \boldsymbol{Q}\boldsymbol{P}\boldsymbol{P}^T\boldsymbol{Q}^T\boldsymbol{Q}\overline{\boldsymbol{X}}^T_{i*} \\
&= \boldsymbol{Q}(I - \boldsymbol{P}\boldsymbol{P}^T)\overline{\boldsymbol{X}}^T_{i*}
\end{aligned}$$

Finally,

$$\|(I - (\boldsymbol{Q}\boldsymbol{P})(\boldsymbol{Q}\boldsymbol{P})^T)\overline{\boldsymbol{C}}^T_{i*}\|_2^2 = \overline{\boldsymbol{X}}_{i*}(I - \boldsymbol{P}\boldsymbol{P}^T)\boldsymbol{Q}^T\boldsymbol{Q}(I - \boldsymbol{P}\boldsymbol{P}^T)\overline{\boldsymbol{X}}^T_{i*} = \|(I - \boldsymbol{P}\boldsymbol{P}^T)\overline{\boldsymbol{X}}^T_{i*}\|_2^2 \tag{4.5}$$

Because the next pivot is selected as the column whose residual has the largest L_2-norm, Eq. 4.5 indicates that the same pivot will be selected for $\overline{\boldsymbol{C}}^T$ and $\overline{\boldsymbol{X}}^T$. Inductively, the anchors $\boldsymbol{S}$ recovered by column pivoted QR on those matrices are equivalent. □

Following the recovery of $\boldsymbol{S}$, AW solves N independent simplex-constrained least square problems $\|\overline{\boldsymbol{C}}_{i*} - \breve{\boldsymbol{B}}_{i*}^T \overline{\boldsymbol{C}}_{\boldsymbol{S}*}\|_2$. Again we can leverage the L_2-norm preserving property,

$$\|\overline{\boldsymbol{C}}_{i*} - \breve{\boldsymbol{B}}_{i*}^T \overline{\boldsymbol{C}}_{\boldsymbol{S}*}\|_2 = \|\boldsymbol{X}_{i*}\boldsymbol{Q}^T - \breve{\boldsymbol{B}}_{i*}^T \boldsymbol{X}_{\boldsymbol{S}*}\boldsymbol{Q}^T\|_2 = \|\boldsymbol{X}_{i*} - \breve{\boldsymbol{B}}_{i*}^T \boldsymbol{X}_{\boldsymbol{S}*}\|_2 \tag{4.6}$$

and reduce the dimension of the least-square problems from N to K, thereby the complexity from $O(N^2K)$ to $O(NK^2)$. The remaining part of the algorithm follows exactly as AW.

Low-rank Joint Stochastic Matrix Factorization (LR-JSMF) We complete our scalable framework of processing co-occurrence statistic by introducing a direct initialization method from the raw object-example data for ENN. This allows us to avoid creating and storing $\boldsymbol{C}$, which is a burden of memory when N becomes sufficiently large. In Algorithm 3, $\boldsymbol{C}$ only appears in the initial truncated eigen-decomposition, after which we maintain the compressed operator $\boldsymbol{C}_{op}$ independent of it. On the other hand, we just need the matrix-vector multiplication by $\boldsymbol{C}$ for the iterative methods in initialization. Using the generative formula in Equation (4.1), we are able to implicitly apply $\boldsymbol{C}$ to vectors as an outer-product plus diagonal operator, in terms of $\boldsymbol{H}$, at $O(NMK)$ computation cost. To further reduce the number of times the operator is applied, we adopt the one-pass randomized eigen-decomposition by Halko et al. [15]. This technique enables us to initialize with a single pass over the dataset, without concurrently storing the entire $\boldsymbol{H}$ in memory. A limitation is when the number of clusters is large and the gap between the K -th eigenvalue and the ones below is small, we will have to incorporate a few power iterations for refinement, as suggested by the original paper. This will result in a multi-pass method, but still far more efficient on large object size and parallelization-friendly.

4.6 Experimental Results

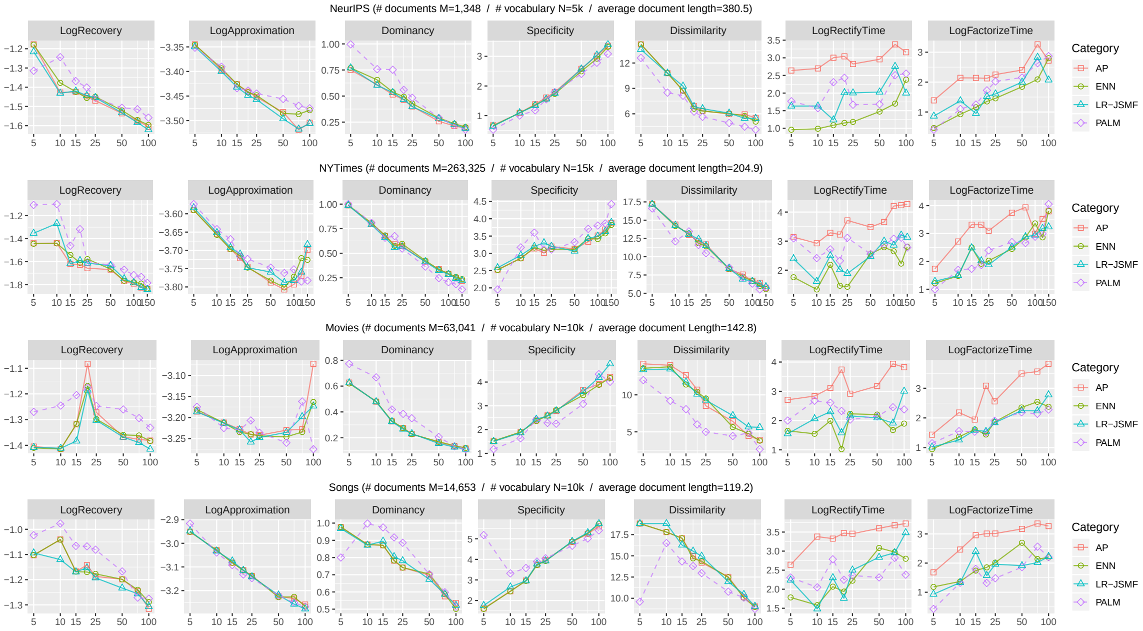

Figure 4.1: Experiment on four datasets. ENN and LR-JSMF mostly agree with AP, whereas PALM has slight inconsistency. The general information of each dataset is above the corresponding row. Recovery, approximation, and runtimes are in $\log_{10}$ scale. Note that ENN and LR-JSMF are almost two orders of magnitude faster than AP. The x-axis indicates the number of clusters K. Lower numbers in y-axis are better except Specificity and Dissimilarity.

A good factorization should be accurate, meaningful, and fast. In two series of experiments, we show that LR-JSMF maintains model quality while running in a fraction of the space and time needed for the original JSMF method. Previous methods have required truncation of the vocabulary even to run on consumer-grade computers. We show not only that we are able to handle increasingly large vocabularies without loss of speed, but that using larger vocabularies measurably improves model quality relative to truncated vocabularies.

For the first series of experiments, we measure the accuracy of each rectification component as well as the entire pipeline of LR-JSMF. To produce a strong baseline, we begin with constructing the full co-occurrence $\boldsymbol{C}$ from each of our datasets $\boldsymbol{H}$ by (4.1), and produce the rectified $\boldsymbol{C}_{AP}$ by running Alternating Projection (AP) on $\boldsymbol{C}$. Next we compress $\boldsymbol{C}$ into $\boldsymbol{Y}_{ENN}$ and $\boldsymbol{Y}_{PALM}$ by running ENN (50 iterations, $|\boldsymbol{I}| = 10K + 1000$) and PALM (100 iterations, $s = 1e^{-4}$). For testing our complete low-rank pipeline, we also construct $(\boldsymbol{V}, \boldsymbol{D})$ directly from the raw data $\boldsymbol{H}$ by the randomized eigen-decomposition in Algorithm 6, learning the compressed statistics $\boldsymbol{Y}_{LR-JSMF}$ again by running ENN initialized with $\boldsymbol{V}\sqrt{\boldsymbol{D}}$. Then we run the Anchor Word algorithm (AW) on $\boldsymbol{C}_{AP}$ and the Low-rank Anchor Word algorithm (LAW) on each of $\boldsymbol{Y}_{ENN}$, $\boldsymbol{Y}_{PALM}$, and $\boldsymbol{Y}_{LR-JSMF}$.

The goal of rectification is to apply spectral inference to data that does not follow our modeling assumptions, so we evaluate on real data. In addition to two standard datasets from the UCI Machine Learning repository (NeurIPS papers and New York Times articles), we also use two non-textual datasets (Movies and Songs) previously used to demonstrate the performance of full algorithm with AP-rectification in [22]. Although our ultimate goal is to extend JSMF to large vocabularies, we use the same restricted vocabulary as [22] for a fair comparison in the first series of experiment.

Figure 4.1 shows the overall performance of the learned topic clusters from these four datasets with increasing number of clusters K. Low **Recovery** error $\frac{1}{N}\sum_i \|\overline{\boldsymbol{C}}_{i*} - \breve{\boldsymbol{B}}_{i*}\overline{\boldsymbol{C}}_{S*}\|_2$ implies that the learned anchor objects successfully reconstruct the co-occurrence space of the entire objects. Low **Approximation** error $\|\boldsymbol{C} - \boldsymbol{B}\boldsymbol{A}\boldsymbol{B}^T\|_2$ means that our factorization captures most of information given in the unbiased co -occurrence statistics. In real data, low **Dominancy** $\frac{1}{K}\sum_k \boldsymbol{A}_{kk}$ implies that our models learn more correlations between clusters. High **Specificity** $\frac{1}{K}\sum_k \mathrm{KL}(\boldsymbol{B}_{*k}\| \sum_i \boldsymbol{C}_{*i})$ indicates that the learned clusters are different enough from the corpus distribution, whereas high **Dissimilarity** counts the average number of objects in each cluster that do not occur among top 20 in other clusters, showing the interpretable difference across the learned clusters. We do not report the Cluster Coherence because it often measures deceptively [12]. The first five columns show that ENN and LR-JSMF learn approximately same clusters as JSMF with the full AP, showing no visible loss in accuracy across all settings. More importantly, the randomness we introduced into LR-JSMF results in a very low variance over a number of runs. This is important as the stability of spectral inference is a major advantage relative to MCMC or Variational Inference. Although PALM deviates a small amount from the other three methods in a few cases, it mostly achieves the same level of accuracy and follows the overall trend closely. In terms of runtimes, all of our methods have clear advantage over AP, gaining 1 $\sim$ 2 orders of magnitude speedup in most situations. Even for applications on relatively small vocabulary sizes, our algorithms shows a notable improvement in efficiency.

For the second series of experiments, we create eight corpora $\{\boldsymbol{H}_N, \boldsymbol{H}_{2N}, ..., \boldsymbol{H}_{8N}\}$ for each dataset $\boldsymbol{H}$ by tailoring their vocabulary sizes as multiples of a base vocabulary of N objects. In this case we are not able to compare LR-JSMF to previous methods because we cannot store the full co-occurrence matrices for the larger vocabulary: these models would be impossible. Figure 4.2 illustrates the overall performance of the learned small/medium/large size clusters with increasing vocabulary size N. Low **BaseRecovery** means that the anchor objects from the models with larger vocabulary better reconstruct the objects in our base vocabulary ($\boldsymbol{H}_N$). High **AnchorQuality** indicates that the average rank of the anchor objects s_k in every other topic clusters than k is high, implying the anchor objects rarely contribute to other clusters than their own. High **Spar-**

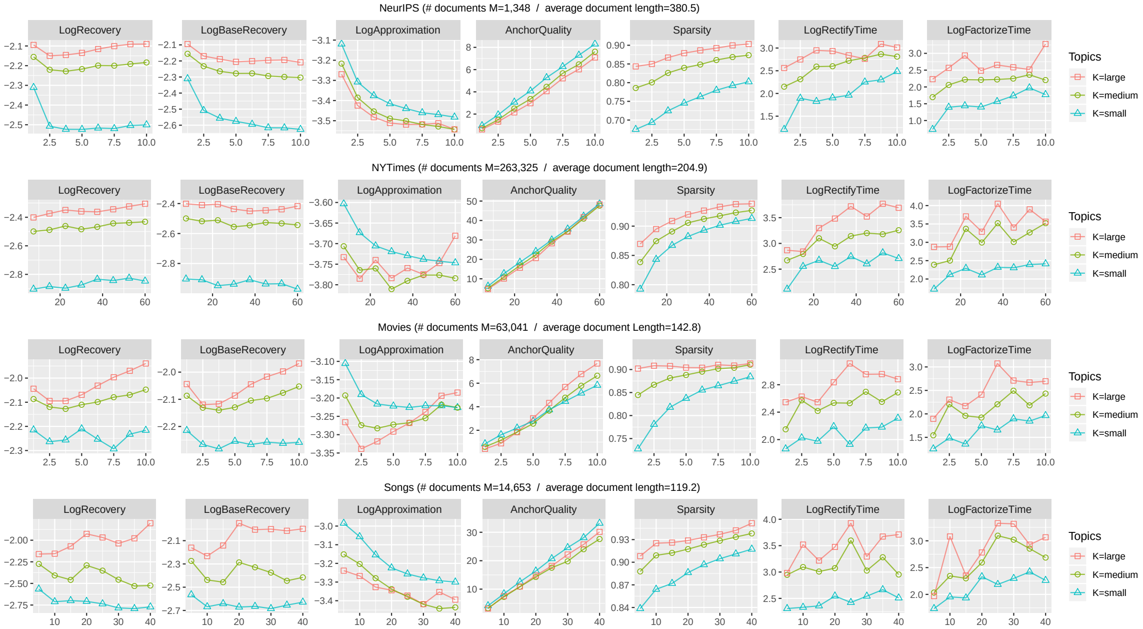

Figure 4.2: As we increase the vocabulary size for four collections, anchor quality and sparsity improve, but running time is stable. The x-axis indicates the vocabulary size N in thousands. Values above 15k will not fit in memory on standard hardware with previous algorithms.

sity ($\frac{1}{K}\sum_k \frac{\sqrt{N}-(\|\boldsymbol{b}_k\|_1/\|\boldsymbol{b}_k\|_2)}{\sqrt{N}-1}$) [16] says that our topic clusters are more concentrated on specific objects.

We observe the quality of anchors increases with increasing vocabulary size, verifying that using larger vocabularies helps better satisfy the separability assumption. We also verify that a large vocabulary often better approximates the co-occurrence statistics and better reconstructs the co-occurrence space of the *base* vocabulary, but these patterns are not always consistent in non-textual datasets. In contrast, Sparsity consistently improves, increasing the interpretability of the learned clusters. Most excitingly, the running times of ENN and LAW show the scalability of our new rectification and low-rank algorithm, thereby demonstrating that LR-JSMF is an efficient and robust pipeline.

Finally, we have also inspected the qualitative behavior of the recovered clusters, as we increase the vocabulary size. The topic clusters become significantly more specific, while the clustering of objects is more conspicuous. Figure 4.3 shows how using a larger vocabulary size can lead to more distinguishable topics, especially as it allows us to make use of words that are relatively rare, but used in much narrower contexts. Going from left to right, we can observe that the set of topic words become more and more specific: for instance, the topic corresponding to the third row is slightly vague when observing just the left half of the row, while as we increase the vocabulary size beyond 5000, we gain access to highly topic-specific words such as hjb (Hamilton-Jacobi-Bellman equation) or pid (Proportional Integral Derivative), which signifies the row's pertinence to dynamical and control systems. The flip side of the figure shows that words we would normally consider as non-topical can often be assigned high contributions towards certain topics. The strong red shade on the bottom left indicates that words such as "equivalent" or "cambridge" are strongly connected to the machine learning literature.

			Vocabulary Size			
1250	2500	3750	5000 (base)	6250	7500	8750
refractory	interspike	bursting	neuron	signalling	ipsp	tst
interconnection	marder	stomatogastric	circuit	meilijson	stg	tam
seen	abbott	konishi	synaptic	quiescent	substances	hyperpolarized
detail	acad	axonal	cell	ryckebusch	inactivation	memorized
transmission	pyloric	modulatory	layer	leech	depolarized	transposed
considered	bird	ionic	signal	silent	shapiro	tsividis
san	male	kanji	recognition	radical	sicl	subword
additional	henderson	phonemic	layer	npm	joe	phoneroes
amount	jackel	subsystem	hidden	demi	chinese	otherfilter
considered	recog	dtw	word	shikano	hanazawa	sdnn
developed	dictionary	strokes	speech	letterform	lexicon	perplexity
significant	ocr	gender	net	tebelskis	preprocessed	males
cambridge	discounted	tutor	control	hjb	jacobi	ovi
pendulum	bradtke	lqr	action	rein	forcement	pid
requires	discount	disturbances	dynamic	biped	viscosity	idm
con	eligibility	disturbance	optimal	trol	sel	umass
bellman	indirect	hamilton	reinforcement	handicapped	bizzi	missile
plan	amherst	smdp	controller	gullapalli	swinging	queueing
directional	transparent	luminance	cell	unoriented	aftereffect	moc
seen	adelson	ruderman	field	heeger	blast	ori
dark	geniculate	andersen	visual	thalamus	mae	taube
neuroscience	amacrine	bergen	motion	directionally	knierim	muller
soc	deg	selectively	direction	hayashi	mexican	swindale
supported	mcnaughton	lond	image	werblin	specimen	skagg
equivalent	fix	solvable	gaussian	birmingham	boxplot	pbr
computing	satisfying	distribu	noise	kolmogorov	dependences	owi
cambridge	royal	barber	approximation	gmm	cb2	danish
considered	opper	parametrized	hidden	winther	trigonometric	ylz
simply	leibler	const	bound	imation	colt	diabetes
detail	treatment	eter	matrix	statist	minimizer	devroye

Figure 4.3: Losses or gains in topic words depending on the vocabulary size. Each row represents a topic from the NeurIPS dataset, with the top 6 topical words shown in the middle column. The red and green cells denote topic words that are lost or gained by shifting the vocabulary size from the default size 5000, respectively. The intensities of the colors indicate the words' contributions towards the specific topic.

4.7 Conclusion

Spectral algorithms provide an appealing alternative for identifying interpretable low-rank subspaces by simple factorizations of higher-order moments. But this simplicity is also a weakness: violations of modeling assumptions destroy performance unless they are handled through rectification, and the size of the moment matrices limits us to small vocabularies. In this paper, we developed an efficient and scalable framework: Low-Rank Joint Stochastic Matrix Factorization. We provide theoretical advances in compressed matrix factorization, leading to high-quality low-rank non-negative approximations without quadratic blowup. The method provides orders of magnitude speedups for rectification even on small vocabularies. Perhaps most importantly, we can now apply reliable, high-quality factorizations of high-dimensional data sets on laptop-grade hardware, massively increasing the applicability and potential use of these algorithms.

CHAPTER 5

WEIGHTED K-MEANS FOR ELECTRONIC STRUCTURE CALCULATION

CHAPTER 6

CONCLUSION

BIBLIOGRAPHY

[1] E A. Erosheva. Bayesian estimation of the grade of membership model. *Bayesian Stat.*, 7, 01 2003.

[2] Edoardo M Airoldi, David M Blei, Stephen E Fienberg, and Eric P Xing. Mixed membership stochastic blockmodels. *Journal of machine learning research*, 9(Sep):1981–2014, 2008.

[3] A. Anandkumar, D. Hsu, and S. Kakade. A method of moments for mixture models and hidden markov models. In *COLT*, 2012.

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

[5] Animashree Anandkumar, Daniel Hsu, and Sham M Kakade. A method of moments for mixture models and hidden markov models. In *Conference on Learning Theory*, pages 33–1, 2012.

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

[7] 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.

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

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

[10] David M Blei, Alp Kucukelbir, and Jon D McAuliffe. Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518):859–877, 2017.

[11] Jérôme Bolte, Shoham Sabach, and Marc Teboulle. Proximal alternating linearized minimization for nonconvex and nonsmooth problems. *Mathematical Programming*, 146(1-2):459–494, 2014.

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

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

[14] Nicolas Gillis and Stephen A Vavasis. Fast and robust recursive algorithms-for separable nonnegative matrix factorization. *IEEE transactions on pattern analysis and machine intelligence*, 36(4):698–714, 2014.

[15] Nathan Halko, Per-Gunnar Martinsson, and Joel A Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM review*, 53(2):217–288, 2011.

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

[17] Daniel Hsu, Sham M Kakade, and Tong Zhang. A spectral algorithm for learning hidden markov models. *Journal of Computer and System Sciences*, 78(5):1460–1480, 2012.

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

[19] M. Jordan, Z. Ghahramani, T. Jaakola, and L. Saul. Introduction to variational methods for graphical models. *Machine Learning*, pages 183–233, 1999.

[20] Alex Kulesza, N Raj Rao, and Satinder Singh. Low-rank spectral learning. In *Artificial Intelligence and Statistics*, pages 522–530, 2014.

[21] Thomas K Landauer, Peter W Foltz, and Darrell Laham. An introduction to latent semantic analysis. *Discourse processes*, 25(2-3):259–284, 1998.

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

[23] Moontae Lee, David Bindel, and David Mimno. From correlation to hierarchy: Practical topic modeling via spectral inference. In *12th INFORMS Workshop on Data Mining and Decision Analytics*, 2017.

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

[25] Dawen Liang, Jaan Altosaar, Laurent Charlin, and David M Blei. Factorization meets the item embedding: Regularizing matrix factorization with item co-occurrence. In *Proceedings of the 10th ACM conference on recommender systems*, pages 59–66. ACM, 2016.

[26] Radford M Neal. *Probabilistic inference using Markov chain Monte Carlo methods*, 1993.

[27] Radford M Neal et al. Mcmc using hamiltonian dynamics. *Handbook of markov chain monte carlo*, 2(11):2, 2011.

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

[29] Jonathan K Pritchard, Matthew Stephens, and Peter Donnelly. Inference of population structure using multilocus genotype data. *Genetics*, 155(2):945–959, 2000.

[30] Christian Robert and George Casella. *Monte Carlo statistical methods*. Springer Science & Business Media, 2013.

[31] M. Wainwright and M. Jordan. Graphical models, exponential families, and variational inference. *Foundations and Trends in Machine Learning*, pages 1–305, 2008.