On-the-Fly Rectification for Robust Large-Vocabulary Topic Inference

Moontae Lee[1] **Sungjun Cho**[2] **Kun Dong**[3] **David Mimno**[4] **David Bindel**[5]

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 algorithms provide transparent and efficient algorithms for posterior inference such as latent topic analysis and community detection. As object vocabularies grow, however, it becomes rapidly more expensive to store and run inference algorithms on co-occurrence statistics. Rectifying co-occurrence, the key process to uphold model assumptions, becomes increasingly more vital in the presence of rare terms, but current techniques cannot scale to large vocabularies. We propose novel methods that simultaneously compress and rectify co-occurrence statistics, scaling gracefully with the size of vocabulary and the dimension of latent space. We also present new algorithms learning latent variables from the compressed statistics, and verify that our methods perform comparably to previous approaches on both textual and non-textual data.

1. Introduction

Understanding the underlying 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 (Bengio et al., 2013). However, exact inference is often intractable due to entangled interactions between the latent variables (Blei et al., 2003; Airoldi et al., 2008; A. Erosheva, 2003; Pritchard et al., 2000). Variational inference transforms the posterior approximation into an optimization problem over simpler distributions with independent parameters (Jordan et al., 1999; Wainwright & Jordan, 2008; Blei et al., 2017), while Markov Chain Monte Carlo enables users to sample from the desired posterior distribution (Neal, 1993; Neal et al., 2011; Robert & Casella, 2013). However, these likelihood-based methods require numerous iterations without any guarantee beyond local improvement at each step (Kulesza et al., 2014).

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 (Lee et al., 2015; Liang et al., 2016). 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 in contexts (Pennington et al., 2014; Levy & Goldberg, 2014). 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 (Arora et al., 2013; Anandkumar et al., 2012c; Hsu et al., 2012; Anandkumar et al., 2012b).

However, two major limitations hinder users from taking advantage of spectral inference based on co-occurrence. First, the second-order co-occurrence matrix already grows quadratically in the number of words (e.g. objects, items, products). Pruning the vocabulary is an option, but for a retailer selling millions of long-tailed products, learning representations of only a subset of the products is inadequate. Second, inference quality is poor in real data that does not necessarily follow our generative model. Whereas likelihood-based methods (Blei et al., 2003; Airoldi et al., 2008; A. Erosheva, 2003; Pritchard et al., 2000) have an intrinsic capability to fit the data to the model despite their mismatch, sample noise can easily destroy the performance of spectral methods even if the data is synthesized from the model (Kulesza et al., 2014; Lee et al., 2015).

[1] Information and Decision Sciences, University of Illinois at Chicago, Chicago, Illinois, USA (also affiliated in Microsoft Research at Redmond, Redmond, Washington, USA) [2] Computational Science and Engineering, Georgia Tech, Atlanta, Georgia, USA [3] Applied Mathematics, Cornell University, Ithaca, New York, USA [4] Information Science, Cornell University, Ithaca, New York, USA [5] Computer Science, Cornell University, Ithaca, New York, USA. Correspondence to: Moontae Lee <moontae@uic.edu>.

Proceedings of the 38^{th} *International Conference on Machine Learning*, PMLR 139, 2021. Copyright 2021 by the author(s).

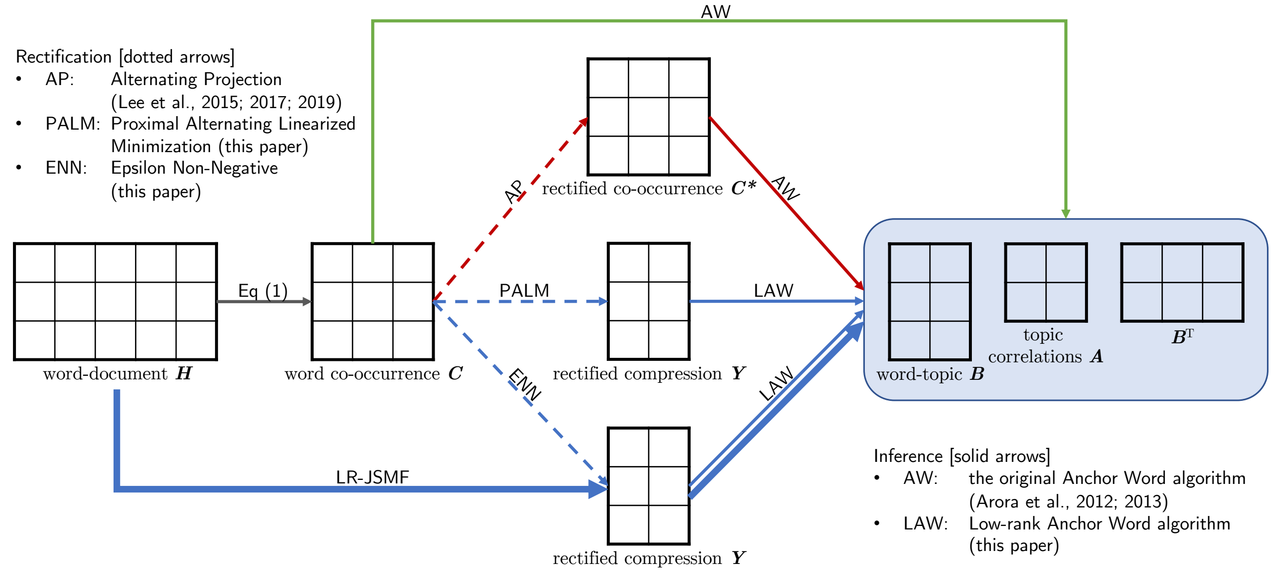
AW
Rectification [dotted arrows]
• AP: Alternating Projection (Lee et al., 2015; 2017; 2019)
• PALM: Proximal Alternating Linearized Minimization (this paper)
• ENN: Epsilon Non-Negative (this paper)
AP
rectified co-occurrence C*
AW
Eq (1)
PALM
LAW
word-document H
word co-occurrence C
ENN
rectified compression Y
LAW
word-topic B
topic correlations A
B^T
LR-JSMF
rectified compression Y
Inference [solid arrows]
• AW: the original Anchor Word algorithm (Arora et al., 2012; 2013)
• LAW: Low-rank Anchor Word algorithm (this paper)

Figure 1. Overall framework. The Rectified Anchor Word algorithm (RAW=AP+AW) (Lee et al., 2015; 2017; 2019) significantly improves the original Anchor Word algorithm (AW) (Arora et al., 2012; 2013) by adopting AP-rectification before running the inference. Developing two novel rectification methods (ENN/PALM) and a low-rank inference algorithm (LAW), we propose a robust and scalable pipeline (**LR-JSMF**) that constructs the ENN-rectified and compressed co-occurrence directly from a bag-of-words raw corpus possibly with large vocabulary (1st bold blue arrow), then running our LAW to efficiently learn high-quality topics and their correlations (2nd bold blue arrow).
.

Rectification, a process of projecting empirical co-occurrence onto a manifold consistent with the posterior geometry of the model, provides a principled treatment that improves the performance of spectral inference in the face of model-data mismatch (Lee et al., 2015). *Alternating Projection rectification (AP)* has been used to rectify the input co-occurrence matrix to the *Anchor Word algorithm (AW)*, a second-order spectral topic model (Lee et al., 2015; 2017; 2019), but running multiple projections dominates overall inference cost even when the vocabulary is small. AP makes the co-occurrence dense as well, exacerbating storage costs when operating on large vocabularies.

In this paper, we propose two efficient methods that simultaneously compress and rectify the co-occurrence matrix, **Epsilon Non-Negative rectification (ENN)** and **Proximal Alternating Linearized Minimization rectification (PALM)**. We also propose the **Low-rank Anchor Word algorithm (LAW)** that learns the latent topics and their correlations only from the compressed statistics, guaranteeing the same performance as the original Anchor Word algorithm under a certain condition. Our experiments show that applying LAW after ENN learns topics of quality comparable to using AW after AP based on the full co-occurrence. We then introduce the **Low-Rank Joint Stochastic Matrix Factorization pipeline (LR-JSMF)** that first adopts a randomized algorithm to construct a low-rank approximation of the full co-occurrence $\boldsymbol{C}$ directly from the raw data; then performs ENN and LAW. While PALM needs access to the full co-occurrence, ENN can work solely with a low-rank initialization, eliminating the burden to ever construct a full co-occurrence matrix. This new pipeline scales to large vocabularies that were previously intractable for spectral inference, and offers a 10x∼100x speedup over previous methods on various textual and non-textual datasets.

Note that second-order spectral topic models often rely on the *separability assumption* that forces at least one anchor word for each topic. This has led to criticism in theory despite their superior performance in practice compared to probabilistic counterparts (Lee et al., 2017) and third-order tensor models (Lee et al., 2019). As most topic models with large vocabularies are proven separable (Ding et al., 2015), we show that our capability to process large vocabularies not only fits for modern datasets, but also alleviates the theoretical limitation. In addition, we also develop a new approach that helps better interpretation of topics by jointly reading characteristic words as well as traditional prominent words. By defining the characteristic words as the terms that are highly associated with each anchor word, we design a graph-based metric that can measure the degree of incoherence in individual topics. To the best of our knowledge, this work makes the first principled attempt to utilize anchor words for quantitative and qualitative interpretations of topics with the prominent words. Given our on-the-fly methods, users are now capable of efficiently understanding latent topics and their correlations from noisy co-occurrence statistics within time and space complexity linear in the size of vocabulary.

Algorithm 1 Anchor Word algorithm (AW)

```
Input: Word co-occurrence $\boldsymbol{C} \in \mathbb{R}^{N \times N}$
       Number of topics $K$
Output: Anchor words $\boldsymbol{S} = \{s_1, ..., s_K\}$
        Latent topics $\boldsymbol{B} \in \mathbb{R}^{N \times K}$
        Topic 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 column pivoted QR on $\overline{\boldsymbol{C}}^{\mathsf{T}}$.
    Find $\breve{\boldsymbol{B}}$ with $\breve{\boldsymbol{B}}_{ki} = p(\text{topic } k \mid \text{word } i)$ by
      solving $N$ simplex-constrained least squares
      in parallel to minimize $\|\overline{\boldsymbol{C}} - \breve{\boldsymbol{B}}^{\mathsf{T}} \overline{\boldsymbol{C}}_{\boldsymbol{S}*}\|_F$.
    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 $t = 0, 1, 2, ...$
        $(\boldsymbol{U}, \boldsymbol{\Lambda}_K) \leftarrow$ Truncated-Eig$(\boldsymbol{C}_t, K)$
        $\boldsymbol{\Lambda}_K^+ \leftarrow \max(\boldsymbol{\Lambda}_K, 0)$
        $\boldsymbol{C}^{\mathcal{PSD}_K} \leftarrow \boldsymbol{U} \boldsymbol{\Lambda}_K^+ \boldsymbol{U}^{\mathsf{T}}$
        $\boldsymbol{C}^{\mathcal{NOR}} \leftarrow \boldsymbol{C}^{\mathcal{PSD}_K} + \frac{1 - \sum_{ij} \boldsymbol{C}_{ij}^{\mathcal{PSD}_K}}{N^2} \boldsymbol{e}\boldsymbol{e}^{\mathsf{T}}$
        $\boldsymbol{C}^{\mathcal{NN}} \leftarrow \max(\boldsymbol{C}^{\mathcal{NOR}}, 0)$
        $\boldsymbol{C}_{t+1} \leftarrow \boldsymbol{C}^{\mathcal{NN}}$
    until converging to a certain $\boldsymbol{C}^*$
    $(\boldsymbol{S}, \boldsymbol{B}, \boldsymbol{A}) \leftarrow$ AW$(\boldsymbol{C}^*, K)$ (Algorithm 1)
end
```

2. Foundations and Rectification

Instead of using Variational inference (Jordan et al., 1999; Wainwright & Jordan, 2008; Blei et al., 2017) or Markov Chain Monte Carlo (Neal, 1993; Neal et al., 2011; Robert & Casella, 2013), our new algorithms build upon the **Joint-Stochastic Matrix Factorization (JSMF)** (Lee et al., 2015). Let $\boldsymbol{H} \in \mathbb{R}^{N \times M}$ be the word-document matrix whose m-th column vector $\boldsymbol{h}_m$ counts the occurrences of each of the N words in the vocabulary in document m. We denote the total number of words in document m by n_m. Given a user-specified number of topics K, we seek to learn a word-topic matrix $\boldsymbol{B} \in \mathbb{R}^{N \times K}$ where $\boldsymbol{B}_{ik}$ is the conditional probability of observing word i given latent topic k. Instead of learning $\boldsymbol{B}$ directly from the sparse and noisy observations $\boldsymbol{H}$, JSMF begins with constructing the joint-stochastic co-occurrence $\boldsymbol{C} \in \mathbb{R}^{N \times N}$ as an unbiased estimator for the underlying generative topic model by

$$\boldsymbol{C} = \hat{\boldsymbol{H}} \hat{\boldsymbol{H}}^{\mathsf{T}} - \hat{\boldsymbol{H}}_{diag} \quad \text{where} \quad \hat{\boldsymbol{h}}_m = \frac{\boldsymbol{h}_m}{\sqrt{n_m(n_m - 1)M}},$$

$$\hat{\boldsymbol{H}}_{diag} = \text{diag}\left(\sum_{m=1}^{M} \frac{\boldsymbol{h}_m}{n_m(n_m - 1)M} \right). \qquad (1)$$

The Anchor Word algorithm (AW) decomposes $\boldsymbol{C}$ into $\boldsymbol{BAB}^{\mathsf{T}}$ by Algorithm 1 (Arora et al., 2013; Lee et al., 2015), where $\boldsymbol{A} \in \mathbb{R}^{K \times K}$ is the topic correlation matrix whose entry $\boldsymbol{A}_{kl}$ captures the joint probability between two latent topics k and l.[1] In the limit, using infinite data generated from the correct probabilistic model, $\boldsymbol{A}$ must agree with the second-moment of the topic proportions (Arora et al., 2012), the Bayesian prior in the model (Lee et al., 2020).

As popular spectral algorithms (Hsu et al., 2012; Anandkumar et al., 2012a) often fail to learn high-quality latent variables beyond synthetic data, the decomposition described above frequently fails to learn high-quality topics due to *model-data mismatch* (Kulesza et al., 2014). Under the probabilistic model assumed to generate the data, $\mathbb{E}[C]$ 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 topics K ($\mathcal{PSD}_K$) (Lee et al., 2015). However, the empirical $\boldsymbol{C}$ from real data is often indefinite and full-rank due to sample noise[2] and the unbiased construction of $\boldsymbol{C}$ in Equation (1) that penalizes all diagonal entries. The **Rectified Anchor Word algorithm (RAW)** has an additional rectification step that forces $\boldsymbol{C}$ to enjoy the expected structures of $\mathbb{E}[\boldsymbol{C}]$ before running the main factorization. The Alternating Projection rectification (AP), as given in Algorithm 2, has been used to overcome the gap between the underlying assumptions of our models and the actual data (Lee et al., 2015; 2017; 2019; 2020).

Rectification is also important for addressing the issue of *outlier bias*. Real data often exhibit rare words that are only present in a few documents. Co-occurrences of these words are inevitably sparse with large variance, but the greedy anchor selection favors choosing these outliers. Previous work tried to bypass this problem by oversampling topics by the number of outliers under additional identifiability assumptions (Gillis & Vavasis, 2014). This approach is not always feasible, especially for a large vocabulary with many rare words. When synonyms and short documents cause undesirable sparsity to Latent Semantic Analysis (Landauer et al., 1998), projection onto the leading eigenspace blurs sparse co-occurrences. Similarly, $\mathcal{PSD}_K$-projection significantly reduces outlier bias, and the remaining projections maintain the probabilistic structures of $\boldsymbol{C}$, which then allow users to recover $\boldsymbol{B}$ and $\boldsymbol{A}$ in Algorithm 1.

[1] Using $\boldsymbol{C}$ is proven to be by far more robust than using $\boldsymbol{H}$ (Arora et al., 2012). Our Eq (1) fixes slightly incorrect construction of $\boldsymbol{C}$ in (Arora et al., 2013).

[2] Rectification notably improves the quality of topics even if a finite amount of data is synthesized from the generative models.

Algorithm 3 ENN-rectification (ENN)

```
Input: Word co-occurrence $\boldsymbol{C} \in \mathbb{R}^{N\times N}$
       Number of topics $K$
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)
    $\boldsymbol{C}^{op} : \boldsymbol{x} \to \boldsymbol{C}\boldsymbol{x}$   (Implicit operator)
    repeat with $t = 0, 1, 2, ...$
        $(\boldsymbol{U}, \boldsymbol{\Lambda}_K) \leftarrow$ Truncated-Eig$(\boldsymbol{C}^{op}, K)$
        $\boldsymbol{\Lambda}_K^+ \leftarrow \max(\boldsymbol{\Lambda}_K, 0)$  $\boldsymbol{Y} \leftarrow \boldsymbol{U}(\boldsymbol{\Lambda}_K^+)^{1/2}$
        $\boldsymbol{E}_{ij} \leftarrow \max(-\boldsymbol{Y}_{i*}\boldsymbol{Y}_{j*}^{\intercal}, 0)$
        $r \leftarrow (1 - \|\boldsymbol{Y}^{\intercal}\boldsymbol{e}\|_2^2 - \sum_{ij}\boldsymbol{E}_{ij})/N^2$
        $\boldsymbol{C}^{op} : \boldsymbol{x} \to \boldsymbol{Y}(\boldsymbol{Y}^{\intercal}\boldsymbol{x}) + \boldsymbol{E}\boldsymbol{x} + r(\boldsymbol{e}^{\intercal}\boldsymbol{x})\boldsymbol{e}$
    until $\boldsymbol{E}$ converges
end
```

Algorithm 4 PALM-rectification (PALM)

```
Input: Word co-occurrence $\boldsymbol{C} \in \mathbb{R}^{N\times N}$
       Number of topics $K$
Output: Rectified compression $\boldsymbol{Y} \in \mathbb{R}^{N\times K}$
begin
    $(\boldsymbol{U}, \boldsymbol{\Lambda}_K) \leftarrow$ Truncated-Eig$(\boldsymbol{C}, K)$
    $(\boldsymbol{X}_0, \boldsymbol{Y}_0) \leftarrow (\boldsymbol{U}\sqrt{\boldsymbol{\Lambda}_K}, \boldsymbol{U}\sqrt{\boldsymbol{\Lambda}_K})$
    repeat with $t = 0, 1, 2, ...$
        $c_t \leftarrow \gamma L_1(\boldsymbol{Y}_t)$
        $\boldsymbol{X}'_{t+1} \leftarrow \boldsymbol{X}_t - (1/c_t)\nabla_{\boldsymbol{X}} J(\boldsymbol{X}_t, \boldsymbol{Y}_t)$
        $\boldsymbol{X}_{t+1} \leftarrow \max(\boldsymbol{X}'_{t+1}, 0)$
        $d_t \leftarrow \gamma L_2(\boldsymbol{X}_{t+1})$
        $\boldsymbol{Y}'_{t+1} \leftarrow \boldsymbol{Y}_t - (1/d_t)\nabla_{\boldsymbol{Y}} J(\boldsymbol{X}_{t+1}, \boldsymbol{Y}_t)$
        $\boldsymbol{Y}_{t+1} \leftarrow \max(\boldsymbol{Y}'_{t+1}, 0)$
    until $\boldsymbol{Y}$ converges
end
```

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 $\mathcal{O}(N^2)$. We are unable to exploit the high sparsity of $\boldsymbol{C}$ as a single iteration of AP makes $\boldsymbol{C}$ significantly denser. The three projections in AP and the rest of AW in Algorithm 1 have time complexities of $\mathcal{O}(N^2K)$, $\mathcal{O}(N^2)$, $\mathcal{O}(N^2)$ and $\mathcal{O}(N^2K)$, respectively. On the other hand, the *separability assumption* is crucial for second-order models. While a line of research has tried to relax this assumption (Bansal et al., 2014; Huang et al., 2016), it is formally shown that most topic models are indeed separable if their vocabulary sizes are sufficiently larger than the number of topics (Ding et al., 2015), again emphasizing the urgency of an approach with better time and space scaling in the vocabulary size.

3. Simultaneous Rectification & Compression

The rectified co-occurrence $\boldsymbol{C}^*$ in Section 2 must be of rank K and positive semidefinite, hinting at an opportunity to represent it as $\boldsymbol{Y}\boldsymbol{Y}^{\intercal}$ for some $\boldsymbol{Y} \in \boldsymbol{R}^{N\times K}$. One idea for achieving this structure is to use a low-rank representation $\boldsymbol{C}_t = \boldsymbol{Y}_t\boldsymbol{Y}_t^{\intercal}$ throughout the rectification in Algorithm 2. Another way to obtain this structure is to directly minimize $\|\boldsymbol{C} - \boldsymbol{Y}\boldsymbol{Y}^{\intercal}\|_F$ with the necessary constraints. Note that random projections cannot preserve algebraic properties of co-occurrence other than L2-norms. Using such projections does not have any rectification effect, decreasing the performance as reported in (Lee & Mimno, 2014). In this section, therefore, we propose two novel compression-plus-rectification algorithms based on these two ideas.

3.1. ENN: Epsilon Non-Negative Rectification

The Alternating Projection rectification (AP) in Algorithm 2 produces low-rank intermediate matrices from the positive semi-definite projection ($\mathcal{PSD}_K$) and the normalization projection ($\mathcal{NOR}$), but the final projection to enforce element-wise non-negativity ($\mathcal{NN}$) destroys this low-rank structure. However, the $\mathcal{NN}$ projection significantly changes only 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 (ENN) in Algorithm 3 has the same structure as Algorithm 2, but with a key difference that it returns a sparse-plus-low-rank representation of the $\mathcal{NN}$ projection rather than a dense representation. Matrix-vector products with this sparse-plus-low-rank representation require $\mathcal{O}(NK + \text{nnz}(\boldsymbol{E}_t))$ time, and $\mathcal{O}(K)$ such matrix-vector products can be used in a Lanczos eigensolver to compute the truncated eigendecomposition at the start of the next iteration.

Maintaining a sparse correction matrix $\boldsymbol{E}_t$ at each step lets the ENN approach avoid storage overheads of the original AP. To overcome the quadratic time cost at each iteration, though, we need to avoid explicitly computing every element of the intermediate $\boldsymbol{Y}\boldsymbol{Y}^{\intercal}$ in the course of the $\mathcal{NN}$ projection. However, we can bound the magnitude of elements of $\boldsymbol{Y}\boldsymbol{Y}^{\intercal}$ by the Cauchy-Schwartz inequality: $|C_{ij}| \leq \|By_i\|_2\|By_j\|_2$ where $\boldsymbol{y}_i$ and $\boldsymbol{y}_j$ denote columns of $\boldsymbol{Y}^{\intercal}$. Let I denote the index set $\{i : \|\boldsymbol{y}_i\|_2^2 > \epsilon\} \subseteq [N]$ for given ϵ; then every large entry of $\boldsymbol{C}$ belongs to either $\boldsymbol{Y}_{I*}\boldsymbol{Y}^{\intercal}$ or $\boldsymbol{Y}(\boldsymbol{Y}^{\intercal})_{*I}$. As $\boldsymbol{C}$ is symmetric, checking the negative entries in $\boldsymbol{Y}_{I*}\boldsymbol{Y}^{\intercal}$ is sufficient to find a symmetric correction $\boldsymbol{E}$ that guarantees $\boldsymbol{Y}\boldsymbol{Y}^{\intercal} + \boldsymbol{E} \geq -\epsilon$. 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 I to look up. Instead of fixing ϵ and finding $I = \{i : \|\boldsymbol{y}_i\|_2^2 > \epsilon\}$, we set $\boldsymbol{Y}_{I*}$ to be $\mathcal{O}(K)$ rows of $\boldsymbol{Y}$ with the largest 2-norms based on the common sampling complexity of a suitable set of rows for a near-optimal rank-K approximation in subset selection.

Algorithm 5 Low-rank AW (LAW)

```
Input: Rectified compression $\boldsymbol{Y} \in \mathbb{R}^{N \times K}$
Output: Anchor words $\boldsymbol{S} = \{s_1, ..., s_K\}$
        Latent topics $\boldsymbol{B} \in \mathbb{R}^{N \times K}$
        Topic correlations $\boldsymbol{A} \in \mathbb{R}^{K \times K}$
begin
    Calculate row sums $\boldsymbol{d} = \boldsymbol{Y}(\boldsymbol{Y}^\intercal \boldsymbol{e})$.
    Compute QR decomposition of $\boldsymbol{Y} = \boldsymbol{Q}\boldsymbol{R}$.
    Form $\overline{\boldsymbol{Y}} = \text{diag}(\boldsymbol{d})^{-1}\boldsymbol{Y}$ and $\boldsymbol{X} = \overline{\boldsymbol{Y}}\boldsymbol{R}^\intercal$.
    Select $\boldsymbol{S}$ using column pivoted QR on $\boldsymbol{X}^\intercal$.
    Solve $n$ simplex-constrained least square problems
      to minimize $\|\boldsymbol{X} - \breve{\boldsymbol{B}}\boldsymbol{X}_{\boldsymbol{S}*}\|_F$.
    Recover $\boldsymbol{B}$ from $\breve{\boldsymbol{B}}$ using Bayes' rule.
    Recover $\boldsymbol{A} = \boldsymbol{B}_{\boldsymbol{S}*}^{-1}\boldsymbol{Y}_{\boldsymbol{S}*}\boldsymbol{Y}_{\boldsymbol{S}*}^\intercal\boldsymbol{B}_{\boldsymbol{S}*}^{-1}$.
end
```

3.2. PALM: Proximal Alternating Linearized Minimization Rectification

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

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

$\mathcal{PSD}_K$- 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 (Bolte et al., 2014) for learning $\boldsymbol{Y}$ given $\boldsymbol{C}$; the relevant proximal operator is $\mathcal{NN}$ projection of $\boldsymbol{Y}$, which takes $\mathcal{O}(NK)$ time at most. Note that J is semi-algebraic (as it is a real polynomial) with two partial derivatives: $\nabla_{\boldsymbol{X}} J = (\boldsymbol{X}\boldsymbol{Y}^\intercal - \boldsymbol{C})\boldsymbol{Y} + s(\boldsymbol{X} - \boldsymbol{Y})$ and $\nabla_{\boldsymbol{Y}} J = (\boldsymbol{Y}\boldsymbol{X}^\intercal - \boldsymbol{C})\boldsymbol{X} + s(\boldsymbol{Y} - \boldsymbol{X})$. Thus the following lemma guarantees global convergence.[3]

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

Proof. $\|\nabla_{\boldsymbol{X}} J(\boldsymbol{X}, \boldsymbol{Y}) - \nabla_{\boldsymbol{X}} J(\boldsymbol{X}', \boldsymbol{Y})\|_F = \|(\boldsymbol{Y}^\intercal\boldsymbol{Y} + s\boldsymbol{I}_K)(\boldsymbol{X} - \boldsymbol{X}')\|_F \leq \|\boldsymbol{Y}^\intercal\boldsymbol{Y} + s\boldsymbol{I}_k\|_2 \cdot \|\boldsymbol{X} - \boldsymbol{X}'\|_F$. The proof is symmetric for the other case with $L_2(\boldsymbol{X}) = \|\boldsymbol{X}^\intercal\boldsymbol{X} + s\boldsymbol{I}_K\|_2$. □

Algorithm 4 shows PALM with adaptive learning rate control based on our 2-norm Lipschitz modulus.

Algorithm 6 Low-rank JSMF (LR-JSMF)

```
Input: Raw word-document $\boldsymbol{H} \in \mathbb{R}^{N \times M}$
Output: Anchor words $\boldsymbol{S} = \{s_1, ..., s_K\}$
        Latent topics $\boldsymbol{B} \in \mathbb{R}^{N \times K}$
        Topic correlations $\boldsymbol{A} \in \mathbb{R}^{K \times K}$
begin
    Get $\hat{\boldsymbol{H}}, \hat{\boldsymbol{H}}_{diag}$ from $\boldsymbol{H}$ by (1).
    $\boldsymbol{C}_{op} : \boldsymbol{x} \rightarrow \hat{\boldsymbol{H}}(\hat{\boldsymbol{H}}^\intercal\boldsymbol{x}) - \hat{\boldsymbol{H}}_{diag}\boldsymbol{x}$
    $(\boldsymbol{V}, \boldsymbol{D}) \leftarrow$ Randomized-Eig$(\boldsymbol{C}_{op}, K)$
    Initialize ENN with $\boldsymbol{V}, \boldsymbol{D}$.
    $\boldsymbol{Y} \leftarrow$ ENN-rectification
    $(\boldsymbol{S}, \boldsymbol{B}, \boldsymbol{A}) \leftarrow$ LAW$(\boldsymbol{Y})$  (Algorithm 5)
end
```

4. Low-rank Anchor Word Algorithm and Scalable Pipeline

ENN and PALM both output a compressed co-occurrence matrix $\boldsymbol{Y}$ with $\boldsymbol{C} \approx \boldsymbol{Y}\boldsymbol{Y}^\intercal$. In this section, we present the **Low-rank Anchor Word algorithm (LAW)** to find anchor words in $\mathcal{O}(NK^2)$ (rather than $\mathcal{O}(N^2K)$) directly from $\boldsymbol{Y}$. Note that LAW applies whenever $\boldsymbol{C}$ is in a low-rank representation, which does not have to be derived from our methods. In addition, LAW performs an exact inference if $\boldsymbol{Y}\boldsymbol{Y}^\intercal \geq \boldsymbol{0}$ but robust in practice when it has small negative entries 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 $\boldsymbol{d} = \boldsymbol{Y}(\boldsymbol{Y}^\intercal\boldsymbol{e})$. To obtain the normalized $\boldsymbol{C}$, we simply scale the rows of $\boldsymbol{Y}$, and $\overline{\boldsymbol{C}} = (\text{diag}(\boldsymbol{d})^{-1}\boldsymbol{Y})\boldsymbol{Y}^\intercal = \overline{\boldsymbol{Y}}\boldsymbol{Y}^\intercal$. These steps cost $\mathcal{O}(NK)$. Next, we need to apply column pivoted QR to $\overline{\boldsymbol{C}}^\intercal$ in order to identify the pivots as our anchor words $\boldsymbol{S}$. By taking the QR decomposition $\boldsymbol{Y} = \boldsymbol{Q}\boldsymbol{R}$, $\overline{\boldsymbol{C}}^\intercal$ can be further transformed into $\boldsymbol{Q}\boldsymbol{R}\overline{\boldsymbol{Y}}^\intercal$. Notice that $\overline{\boldsymbol{C}}^\intercal$ is an orthogonal embedding of $\boldsymbol{X}^\intercal = \boldsymbol{R}\overline{\boldsymbol{Y}}^\intercal$ onto a higher-dimensional space, which preserves the column L_2-norms. Lemma 2 shows that column pivoted QR on $\overline{\boldsymbol{C}}^\intercal$ and on $\boldsymbol{R}\overline{\boldsymbol{Y}}^\intercal$ are equivalent, which allows us to lower the computation cost from $\mathcal{O}(N^2K)$ to $\mathcal{O}(NK^2)$.

Lemma 2. *Let* $\boldsymbol{S}$ *be the set of pivots that have been selected by column pivoted QR on* $\overline{\boldsymbol{C}}^\intercal = \boldsymbol{Q}\boldsymbol{X}^\intercal$. *Given the QR decomposition,* $\overline{\boldsymbol{X}}_{\boldsymbol{S}*}^\intercal = \boldsymbol{P}\boldsymbol{T}$, *then* $\overline{\boldsymbol{C}}_{\boldsymbol{S}*}^\intercal = (\boldsymbol{Q}\boldsymbol{P})\boldsymbol{T}$ *is the corresponding QR decomposition for the columns of* $\overline{\boldsymbol{C}}$. *For any remaining row* $i \in [N] \setminus \boldsymbol{S}$ *where* $[N] = \{1, 2, ..., N\}$,

$$\|(\boldsymbol{I} - \boldsymbol{P}\boldsymbol{P}^\intercal)\overline{\boldsymbol{X}}_{i*}^\intercal\|_2 = \|(\boldsymbol{I} - (\boldsymbol{Q}\boldsymbol{P})(\boldsymbol{Q}\boldsymbol{P})^\intercal)\overline{\boldsymbol{C}}_{i*}^\intercal\|_2 \tag{3}$$

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

[3] One could further improve the performance of PALM-rectification by using an inertial version like iPALM in (Pock & Sabach, 2016). As PALM is not our main rectification method for the scalable pipeline, we leave this as a space for future work.

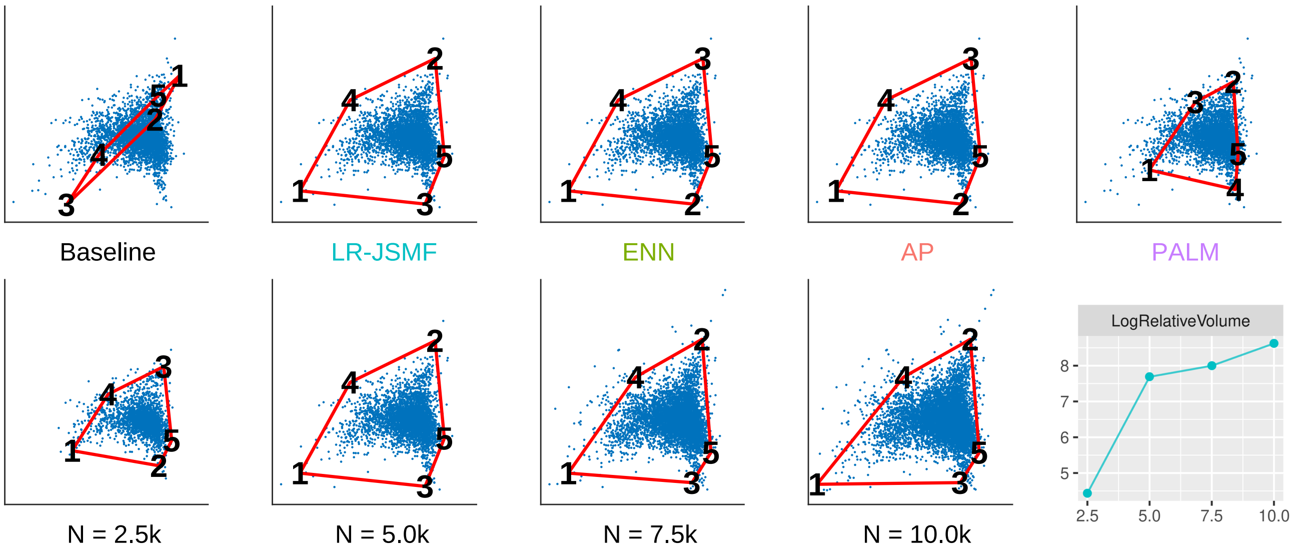

Figure 2. 2D visualizations of co-occurrence spaces in NeurIPS. Blue dots are words. Each vertex on the convex hulls corresponds to the k-th anchor word. First row shows that ENN and LR-JSMF find same anchors as AP (Lee et al., 2015). PALM is off but outperforms Baseline, which is AW without any rectification (Arora et al., 2013). Second row shows growing anchor convex hull volumes relative to random convex hulls when expanding vocabularies.

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

$$(\boldsymbol{QP})^\intercal(\boldsymbol{QP}) = \boldsymbol{P}^\intercal(\boldsymbol{Q}^\intercal\boldsymbol{Q})\boldsymbol{P} = \boldsymbol{P}^\intercal\boldsymbol{P} = \boldsymbol{I}$$

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

$$\begin{aligned}
&(\boldsymbol{I} - (\boldsymbol{QP})(\boldsymbol{QP})^\intercal)\overline{\boldsymbol{C}}^\intercal_{i*} \\
=&(\boldsymbol{I} - (\boldsymbol{QP})(\boldsymbol{QP})^\intercal)\boldsymbol{Q}\overline{\boldsymbol{X}}^\intercal_{i*} \\
=&\boldsymbol{Q}\overline{\boldsymbol{X}}^\intercal_{i*} - \boldsymbol{QPP}^\intercal\boldsymbol{Q}^\intercal\boldsymbol{Q}\overline{\boldsymbol{X}}^\intercal_{i*} \\
=&\boldsymbol{Q}(\boldsymbol{I} - \boldsymbol{PP}^\intercal)\overline{\boldsymbol{X}}^\intercal_{i*}
\end{aligned}$$

Finally,

$$\begin{aligned}
&\|(\boldsymbol{I} - (\boldsymbol{QP})(\boldsymbol{QP})^\intercal)\overline{\boldsymbol{C}}^\intercal_{i*}\|_2^2 \\
=&\overline{\boldsymbol{X}}_{i*}(\boldsymbol{I} - \boldsymbol{PP}^\intercal)\boldsymbol{Q}^\intercal\boldsymbol{Q}(\boldsymbol{I} - \boldsymbol{PP}^\intercal)\overline{\boldsymbol{X}}^\intercal_{i*} \\
=&\|(\boldsymbol{I} - \boldsymbol{PP}^\intercal)\overline{\boldsymbol{X}}^\intercal_{i*}\|_2^2
\end{aligned} \tag{4}$$

Because the next pivot is selected as the column whose residual has the largest L_2-norm, Eq. 4 indicates that the same pivot will be selected for $\overline{\boldsymbol{C}}^\intercal$ and $\overline{\boldsymbol{X}}^\intercal$. 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 $\|\overline{\boldsymbol{C}}_{i*} - \breve{\boldsymbol{B}}^\intercal_{i*}\overline{\boldsymbol{C}}_{\boldsymbol{S}*}\|_2$. Again we can leverage the L_2-norm preserving property,

$$\begin{aligned}
\|\overline{\boldsymbol{C}}_{i*} - \breve{\boldsymbol{B}}^\intercal_{i*}\overline{\boldsymbol{C}}_{\boldsymbol{S}*}\|_2 &= \|\boldsymbol{X}_{i*}\boldsymbol{Q}^\intercal - \breve{\boldsymbol{B}}^\intercal_{i*}\boldsymbol{X}_{\boldsymbol{S}*}\boldsymbol{Q}^\intercal\|_2 \\
&= \|\boldsymbol{X}_{i*} - \breve{\boldsymbol{B}}^\intercal_{i*}\boldsymbol{X}_{\boldsymbol{S}*}\|_2
\end{aligned} \tag{5}$$

and reduce the dimension of the least-square problems from N to K, thereby reducing the complexity from $\mathcal{O}(N^2K)$ to $\mathcal{O}(NK^2)$. 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 statistics by introducing a direct initialization method for ENN from the raw word-document data. This allows us to avoid creating and storing $\boldsymbol{C}$, which is a burden of memory when N becomes large. In Algorithm 3, $\boldsymbol{C}$ only appears in the initial truncated eigendecomposition, 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 iterative methods in initialization. Using the generative formula in Equation (1), we are able to implicitly apply $\boldsymbol{C}$ to vectors as an outer-product plus diagonal operator in terms of $\boldsymbol{H}$, at $\mathcal{O}(NMK)$ computation cost.

Note that even when $M > N$, using $\boldsymbol{H}$ is still more efficient than using $\boldsymbol{C}$ due to higher sparsity of $\boldsymbol{H}$. To further reduce the number of times the operator is applied, we adopt the one-pass randomized eigendecomposition by Halko et al. (Halko et al., 2011). This technique enables initialization with a single pass over the dataset, without concurrently storing the entire $\boldsymbol{H}$ in memory. A limitation is when the number of topics 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 vocabularies and parallelization-friendly.

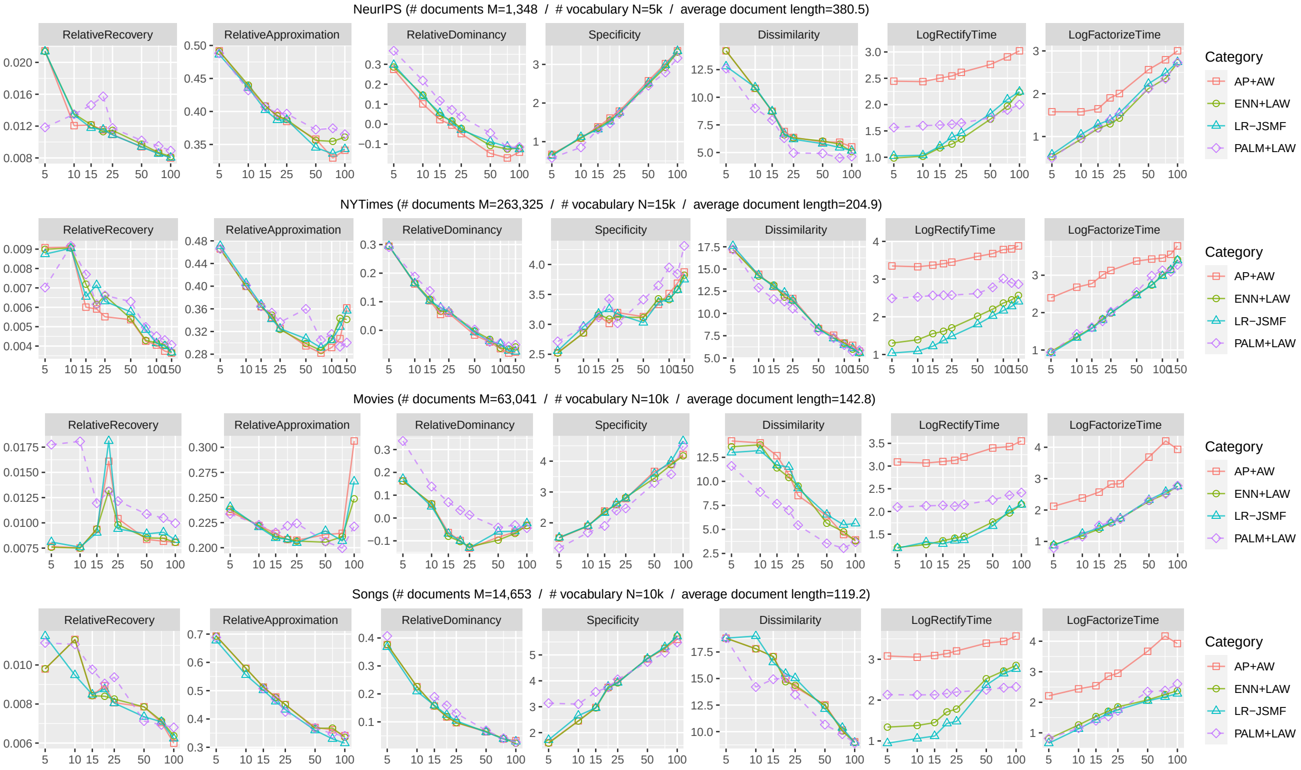

Figure 3. Experiment on four datasets. ENN and LR-JSMF agree with AP, while PALM has slight inconsistency. The basic dataset statistics is above each row. Runtimes are in $\log_{10}$ seconds. Note that ENN and LR-JSMF are almost two orders of magnitude faster than AP. The x-axis indicates the number of topics K. In y-axes, lower is better except for Specificity and Dissimilarity.

5. Experimental Results

A good factorization should be accurate, meaningful, and fast. The first row of Figure 2 illustrates that using ENN with LAW or LR-JSMF pipeline directly with the raw data correctly recover the anchor convex hulls. The second row shows that the volumes of the anchor convex hulls relative to the volumes of the random convex hulls grow over increasing vocabularies.[4] In the next two series of experiments, we demonstrate that our simultaneous rectification and compression maintains model quality while running in a fraction of the space and time needed for the original JSMF framework. The code is publicly available.[5]

For the first series of experiments, we measure the accuracy of each rectification component as well as the entire pipeline of LR-JSMF. For thorough comparisons, we construct the full co-occurrence $\boldsymbol{C}$ from each of our datasets $\boldsymbol{H}$ by (1), and we produce the rectified $\boldsymbol{C}_{AP}$ by running AP on $\boldsymbol{C}$. Next we compress $\boldsymbol{C}$ into $\boldsymbol{Y}_{ENN}$ and $\boldsymbol{Y}_{PALM}$ by running ENN (with $|I| = 10K + 1000$) and PALM (with $s = 1e^{-4}$) until convergence. To test the complete low-rank pipeline, we also construct $(\boldsymbol{V}, \boldsymbol{D})$ from the raw data $\boldsymbol{H}$ by the randomized eigendecomposition in Algorithm 6, learning the rectified and compressed statistics $\boldsymbol{Y}_{LR-JSMF}$ by running ENN initialized with $\boldsymbol{V}\sqrt{\boldsymbol{D}}$. Then we run AW on $\boldsymbol{C}_{AP}$ and LAW on each of $\boldsymbol{Y}_{ENN}$, $\boldsymbol{Y}_{PALM}$, and $\boldsymbol{Y}_{LR-JSMF}$.

The goal of rectification is to ensure robust spectral inference to data that does not follow our modeling assumptions, so we evaluate on real data: two standard textual datasets from the UCI Machine Learning repository (NeurIPS papers and New York Times articles) as well as two non-textual datasets (Movies from Movielens 10M star-ratings and Songs from Yes.com complete playlists) previously used to show the performance of JSMF with AP in (Lee et al., 2015). We apply identical vocabulary curation with (Lee et al., 2015) for fair comparisons. Data statistics are given in each figure.

Figure 3 shows the overall performance of the learned topics from the four datasets with increasing number of topics K. For fair comparisons across increasing vocabulary sizes later, we extend the intrinsic quality metrics used in (Lee et al., 2015) to be relative to vocabulary scales. Low **RelativeRecovery** error $(\frac{1}{N}\sum_i \|\overline{\boldsymbol{C}}_{i*} - \breve{\boldsymbol{B}}_{i*}\overline{\boldsymbol{C}}_{\boldsymbol{S}*}\|_2 / \|\overline{\boldsymbol{C}}\|_F)$ implies that the learned anchor words successfully recon-

[4] We build a set of random convex hulls by uniformly sampling each row of $\overline{\boldsymbol{C}}$ from the corresponding simplex, then finding the anchors by the same column-pivoted QR on $\overline{\boldsymbol{C}}^{\mathsf{T}}$.

[5] https://github.com/moontae/JSMF

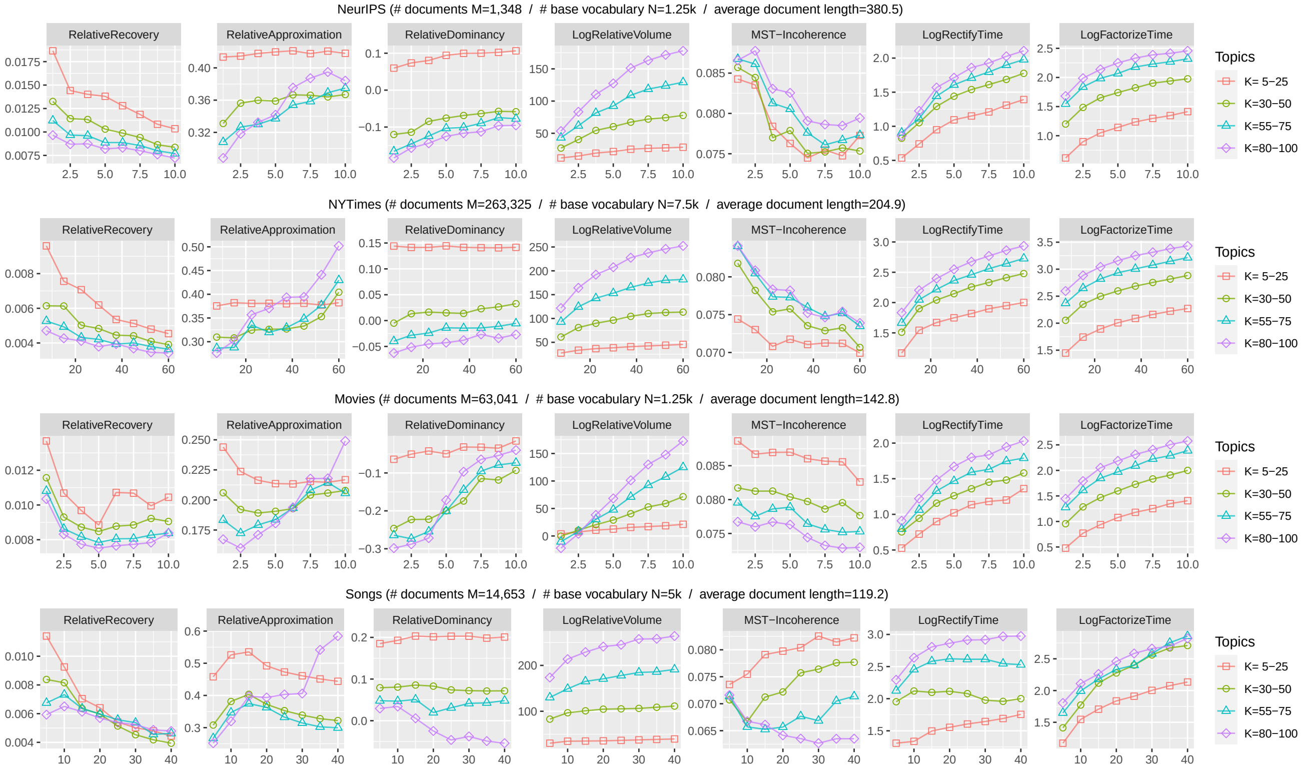

Figure 4. As we increase the vocabulary size N, the relative volume of anchor convex hulls grows. MST-Incoherence decreases in general, but runtimes are stable. The x-axis shows N in thousands. Vocabularies above 15k do not fit in memory on standard hardware with previous algorithms.

struct the co-occurrence space (as in Figure 2) of the entire words. Low **RelativeApproximation** error ($\|\boldsymbol{C} - \boldsymbol{B}\boldsymbol{A}\boldsymbol{B}^{\mathsf{T}}\|_F / \|\boldsymbol{C}\|_F$) means that our factorization captures most of information given in the unbiased co-occurrence statistics. Topics in real data often exhibit correlations, and low **RelativeDominancy** ($\frac{1}{K}\sum_k \boldsymbol{A}_{kk}/\|\boldsymbol{A}\|_F$) implies that our models learn more correlations between different topics. High **Specificity** ($\frac{1}{K}\sum_k \mathrm{KL}(\boldsymbol{B}_{*k}\|\sum_i \boldsymbol{C}_{*i})$) indicates that the learned topics are distinct from the corpus unigram distribution, and high **Dissimilarity** tells that most top words in each topic do not occur within the top 20 words of the other topics, showing interpretable difference across the learned topics. We do not report a traditional topic Coherence (Chang et al., 2009; Mimno et al., 2011) as it often measures deceptively if a model learns many duplicated topics whose top words are mostly high-frequency unigrams (Huang et al., 2016).

The first five columns show that using ENN+LAW or LR-JSMF learn approximately same topics as using AP+AW without any visible loss in accuracy across all settings. More importantly, the randomness in LR-JSMF produces very low variance over a number of runs. This is important as the stability of spectral inference is a major advantage over MCMC or Variational Inference. Although using PALM+LAW deviates slightly from the other three methods, it mostly achieves the same level of accuracy and follows the overall trend closely. Note that all of our methods have clear advantage over AP+AW, gaining $1 \sim 2$ orders of magnitude speedup in most situations. Even with the relatively small vocabularies, our algorithms show notable improvements in efficiency.

Despite the success of anchor-based topic modeling, the anchor words themselves are too rare terms to help interpretation of topics. In addition to the traditional *Prominent Words (PWs)* selected from $\boldsymbol{B}_{*k}$ for each topic k, we define *Characteristic Words (CWs)* as the most co-occurring terms with the anchor word s_k with respect to $\overline{\boldsymbol{C}}_{s_k *}$. As shown in Table 1, reading *biological* (never chosen as a CW in the smaller vocabularies) clarifies that the first topic is more about neuroscience than computer science. Similarly reading {*character, kanji, radical*} together with the PWs hints that the second topic describes written/spoken language recognition. By reading PWs and CWs altogether, users can better understand both general and specific details of the topic, also inspiring a new coherence metric.

For the second series of experiments, we create corpora $\{\boldsymbol{H}_{kN}\}_{k=1}^{8}$ by choosing vocabularies of kN words with greatest tf-idf scores. Here we do not compare LR-JSMF

Table 1. 5 topics from NeurIPS with vocabulary of size 10k. Left column: each line shows the top 6 words that contribute the most to the topic k in $\boldsymbol{B}_{*k}$. Right column: each line shows the top 6 words that co-occur most frequently with the anchor word s_k in $\overline{\boldsymbol{C}}_{s_k*}$. Using characteristic words in bold in addition to prominent words enables more specific and definitive interpretation of topics.

Top Prominent Words from B by LR-JSMF	Top Characteristic Words from $\overline{C}$
neuron cell circuit synaptic layer signal	neuron synaptic **potential** cell **biological** circuit
layer recognition hidden word speech net	**character samples kanji** recognition **radical** layer
control action dynamic optimal policy reinforcement	**tpdp** control **states** optimal action dynamic
cell field visual image motion direction	motion cell direction **contrast signal region**
gaussian noise approximation bound hidden matrix	**conditional** bound **likelihood cem log** gaussian

to ENN/PALM/AP as we cannot store full co-occurrence matrices. Grouping results from $K = 5$ to 100 into four categories[6], Figure 4 shows the performance of the topics with increasing vocabulary size. High **LogRelativeVolume** means that the volume of an anchor convex hull (as in Figure 2) is large relative to the average volume of random convex hulls in the same N-dimensional space. We measure incoherence of each topic as the minimum spanning tree cost of the associated graph where nodes are the union of 7 PWs and 7 CWs. Every node pair (i, j) on the graph is linked with an undirected edge with weight $\frac{1}{2}(1 - \text{NPMI}(i, j))$ where $\text{NPMI}(i, j) = -\text{PMI}(i, j)/\log \boldsymbol{C}_{ij}$ as in (Röder et al., 2015). Low **MST-Incoherence** implies that every topic has at least one path of top words that allows coherent understanding. When a topic consists of two relevant subtopics bridged by a few top words, our MST-Incoherence is less sensitive to possibly large pairwise distances between the top words of the two subtopics.[7]

As N increases, the relative volumes of anchor convex hulls grow, while relative recovery errors decrease. This means inference quality improves because LR-JSMF chooses better anchor words from larger vocabularies, thereby better representing non-anchor words inside the convex hulls. As each anchor word corresponds to a vertex of the growing anchor convex hulls, users are provided with more informative characteristic words over increasing N. The decreasing MST-Incoherence supports our intuition, again emphasizing the power of using large vocabulary. Most excitingly, the running times of ENN and LAW show the scalability of our new rectification and decomposition algorithms, thereby demonstrating the efficient and robust pipeline of LR-JSMF. Figure 4 does not report Specificity and Dissimilarity because we cannot directly compare distributional distances measured on the different supports. The supplementary material includes all the missing panels in Figures 3 and 4.

6. Conclusion

Spectral algorithms provide appealing alternatives for identifying interpretable low-rank subspaces by simple factorizations of higher-order co-occurrence data. But this simplicity is also a weakness: the size of the co-occurrence limit us to small vocabularies, and these methods perform poorly without rectifications that previously suffered quadratic scaling. Anchor words are guaranteed to be exclusive to the corresponding topics, but they are rarely used for topic interpretations because they are often chosen as too rare terms.

We develop a robust and scalable pipeline: Low-Rank Joint Stochastic Matrix Factorization based on our two complementary on-the-fly rectification methods (ENN/PALM) and a sufficiently general low-rank inference algorithm (LAW). These methods simultaneously compress and rectify the co-occurrence from raw data; learn high-quality topics from the compressed matrix factorization; and achieve low-rank non-negative approximations without quadratic blowup. They also provide orders of magnitude speedups for rectification even on small vocabularies. In addition, we verify that using large vocabularies benefits inference quality by better satisfying the separability assumption. It also improves model interpretability by jointly understanding the prominent words with the characteristic words, and by measuring our MST-Incoherence metric for individual topics. Given all these new development, we can now learn and evaluate useful low-dimensional structures in high-dimensional datasets on laptop-grade hardware, massively increasing the applicability and potential use of the spectral algorithms.[8]

Acknowledgements

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

[6] For example, K=30-50 in Figure 4 means that individual metrics are averaged over multiple runs using K=30, 35, 40, 45, and 50. It is to evaluate mean performance over varying ranges of K.

[7] The graphs in MST-Incoherence are complete graphs of limited size (max 14 nodes), regardless of the vocab size. Adding more words does not allow more paths. Thus the metric relies solely on the associations amongst the prominent and characteristic words.

[8] The **supplementary material** of this paper consists of various friendly answers to potential questions from readers. Reading it together with the main paper will boost your understanding.

References

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

Airoldi, E. M., Blei, D. M., Fienberg, S. E., and Xing, E. P. Mixed membership stochastic blockmodels. *Journal of machine learning research*, 9(Sep):1981–2014, 2008.

Anandkumar, A., Foster, D. P., Hsu, D., Kakade, S., and Liu, Y. A spectral algorithm for latent Dirichlet allocation. In *NIPS*, 2012a.

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

Anandkumar, A., Hsu, D., and Kakade, S. M. A method of moments for mixture models and hidden markov models. In *Conference on Learning Theory*, pp. 33–1, 2012c.

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

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

Bansal, T., Bhattacharyya, C., and Kannan, R. A provable SVD-based algorithm for learning topics in dominant admixture corpus. In *NIPS*, 2014.

Bengio, Y., Courville, A., and Vincent, P. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8): 1798–1828, 2013.

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

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

Bolte, J., Sabach, S., and Teboulle, M. Proximal alternating linearized minimization for nonconvex and nonsmooth problems. *Mathematical Programming*, 146(1-2):459–494, 2014.

Chang, J., Boyd-Graber, J., Wang, C., Gerrish, S., and Blei, D. M. Reading tea leaves: How humans interpret topic models. In *NIPS*, 2009.

Ding, W., Ishwar, P., and Saligrama, V. Most large topic models are approximately separable. In *ITA, 2015*, pp. 199–203. IEEE, 2015.

Gillis, N. and Vavasis, S. A. Fast and robust recursive algorithmsfor separable nonnegative matrix factorization. *IEEE transactions on pattern analysis and machine intelligence*, 36(4):698–714, 2014.

Halko, N., Martinsson, P.-G., and Tropp, J. A. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM review*, 53(2):217–288, 2011.

Hsu, D., Kakade, S. M., and Zhang, T. A spectral algorithm for learning hidden markov models. *Journal of Computer and System Sciences*, 78(5):1460–1480, 2012.

Huang, K., Fu, X., and Sidiropoulos, N. D. Anchor-free correlated topic modeling: Identifiability and algorithm. In *NIPS*, 2016.

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

Kulesza, A., Rao, N. R., and Singh, S. Low-rank spectral learning. In *Artificial Intelligence and Statistics*, pp. 522–530, 2014.

Landauer, T. K., Foltz, P. W., and Laham, D. An introduction to latent semantic analysis. *Discourse processes*, 25(2-3): 259–284, 1998.

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

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

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

Lee, M., Cho, S., Bindel, D., and Mimno, D. Practical correlated topic modeling and analysis via the rectified anchor word algorithm. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 4992–5002, 2019.

Lee, M., Bindel, D., and Mimno, D. Prior-aware composition inference for spectral topic models. In *Artificial Intelligence and Statistics 2020 (AISTATS)*, 2020.

Levy, O. and Goldberg, Y. Neural word embedding as implicit matrix factorization. In *NIPS*, 2014.

Liang, D., Altosaar, J., Charlin, L., and Blei, D. M. Factorization meets the item embedding: Regularizing matrix factorization with item co-occurrence. In *Proceedings of the 10th ACM conference on recommender systems*, pp. 59–66. ACM, 2016.

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

Neal, R. M. *Probabilistic inference using Markov chain Monte Carlo methods*, 1993.

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

Pennington, J., Socher, R., and Manning, C. D. GloVe: Global vectors for word representation. In *EMNLP*, 2014.

Pock, T. and Sabach, S. Inertial proximal alternating linearized minimization (ipalm) for nonconvex and nonsmooth problems. *SIAM Journal on Imaging Sciences*, 9 (4):1756–1787, 2016.

Pritchard, J. K., Stephens, M., and Donnelly, P. Inference of population structure using multilocus genotype data. *Genetics*, 155(2):945–959, 2000.

Robert, C. and Casella, G. *Monte Carlo statistical methods*. Springer Science & Business Media, 2013.

Röder, M., Both, A., and Hinneburg, A. Exploring the space of topic coherence measures. In *Proceedings of the eighth ACM international conference on Web search and data mining*, pp. 399–408, 2015.

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