arXiv:1204.1956v2 [cs.LG] 10 Apr 2012

Learning Topic Models — Going beyond SVD

Sanjeev Arora * Rong Ge † Ankur Moitra ‡

April 11, 2012

Abstract

Topic Modeling is an approach used for automatic comprehension and classification of data in a variety of settings, and perhaps the canonical application is in uncovering thematic structure in a corpus of documents. A number of foundational works both in machine learning [15] and in theory [27] have suggested a probabilistic model for documents, whereby documents arise as a convex combination of (i.e. distribution on) a small number of *topic* vectors, each topic vector being a distribution on words (i.e. a vector of word-frequencies). Similar models have since been used in a variety of application areas; the *Latent Dirichlet Allocation* or LDA model of Blei et al. is especially popular.

Theoretical studies of topic modeling focus on learning the model's parameters *assuming the data is actually generated from it.* Existing approaches for the most part rely on *Singular Value Decomposition* (SVD), and consequently have one of two limitations: these works need to either assume that each document contains only one topic, or else can only recover the *span* of the topic vectors instead of the topic vectors themselves.

This paper formally justifies *Nonnegative Matrix Factorization* (NMF) as a main tool in this context, which is an analog of SVD where all vectors are nonnegative. Using this tool we give the first polynomial-time algorithm for learning topic models without the above two limitations. The algorithm uses a fairly mild assumption about the underlying topic matrix called *separability*, which is usually found to hold in real-life data. A compelling feature of our algorithm is that it generalizes to models that incorporate topic-topic correlations, such as the *Correlated Topic Model* (CTM) and the *Pachinko Allocation Model* (PAM).

We hope that this paper will motivate further theoretical results that use NMF as a replacement for SVD – just as NMF has come to replace SVD in many applications.

*Princeton University, Computer Science Department and Center for Computational Intractability. Email: arora@cs.princeton.edu. This work is supported by the NSF grants CCF-0832797 and CCF-1117309.

†Princeton University, Computer Science Department and Center for Computational Intractability. Email: rongge@cs.princeton.edu. This work is supported by the NSF grants CCF-0832797 and CCF-1117309.

‡Institute for Advanced Study, School of Mathematics. Email: moitra@ias.edu. Research supported in part by NSF grant No. DMS-0835373 and by an NSF Computing and Innovation Fellowship.

1 Introduction

Developing tools for automatic comprehension and classification of data —web pages, newspaper articles, images, genetic sequences, user ratings — is a holy grail of machine learning. *Topic Modeling* is an approach that has proved successful in all of the aforementioned settings, though for concreteness here we will focus on uncovering thematic structure of a corpus of documents (see e.g. [4], [6]).

In order to learn structure one has to posit the *existence* of structure, and in topic models one assumes a *generative model* for a collection of documents. Specifically, each document is represented as a vector of word-frequencies (the *bag of words* representation). Seminal papers in theoretical CS (Papadimitriou et al. [27]) and machine learning (Hofmann's *Probabilistic Latent Semantic Analysis* [15]) suggested that documents arise as a convex combination of (i.e. distribution on) a small number of *topic* vectors, where each topic vector is a distribution on words (i.e. a vector of word-frequencies). Each convex combination of topics thus is itself a distribution on words, and the document is assumed to be generated by drawing N independent samples from it. Subsequent work makes specific choices for the distribution used to generate topic combinations —the well-known *Latent Dirichlet Allocation* (LDA) model of Blei et al [6] hypothesizes a *Dirichlet* distribution (see Section 4).

In machine learning, the prevailing approach is to use local search (e.g. [11]) or other heuristics [32] in an attempt to find a *maximum likelihood* fit to the above model. For example, fitting to a corpus of newspaper articles may reveal 50 topic vectors corresponding to, say, politics, sports, weather, entertainment etc., and a particular article could be explained as a $(1/2, 1/3, 1/6)$-combination of the topics politics, sports, and entertainment. Unfortunately (and not surprisingly), the maximum likelihood estimation is NP-hard (see Section 6) and consequently when using this paradigm, it seems necessary to rely on unproven heuristics even though these have well-known limitations (e.g. getting stuck in a local minima [11, 28]).

The work of Papadimitriou et al [27] (which also formalized the topic modeling problem) and a long line of subsequent work have attempted to give *provable* guarantees for the problem of learning the model parameters *assuming the data is actually generated from it.* This is in contrast to a maximum likelihood approach, which asks to find the closest-fit model for arbitrary data. The principal algorithmic problem is the following (see Section 1.1 for more details):

> **Meta Problem in Topic Modeling:** *There is an unknown topic matrix A with nonnegative entries that is dimension $n \times r$, and a stochastically generated unknown matrix W that is dimension $r \times m$. Each column of AW is viewed as a probability distribution on rows, and for each column we are given $N \ll n$ i.i.d. samples from the associated distribution.*
>
> **Goal:** *Reconstruct A and parameters of the generating distribution for W.*

The challenging aspect of this problem is that we wish to recover *nonnegative* matrices A, W with small inner-dimension r. The general problem of finding nonnegative factors A, W of specified dimensions when given the matrix AW (or a close approximation) is called the *Nonnegative Matrix Factorization* (NMF) problem (see [22], and [2] for a longer history) and it is NP-hard [31]. Lacking a tool to solve such problems, theoretical work has generally relied on the *Singular Value Decomposition* (SVD) which given the matrix AW will instead find factors U, V with both positive and negative entries. SVD can be used as a tool for clustering – in which case one needs to assume that each document has only *one* topic. In Papadimitriou et al [27] this is called the *pure documents* case and is solved under strong assumptions about the topic matrix A (see also [26] and [1] which

uses the method of moments instead). Alternatively, other papers use SVD to recover the *span* of the columns of A (i.e. the topic vectors) [3], [20], [19], which suffices for some applications such as computing the inner product of two document vectors (in the space spanned by the topics) as a measure of their *similarity*.

These limitations of existing approaches —either restricting to one topic per document, or else learning only the span of the topics instead of the topics themselves—are quite serious. In practice documents are much more faithfully described as a distribution on topics and indeed for a wide range of applications one needs the actual topics and not just their span – such as when browsing a collection of documents without a particular query phrase in mind, or tracking how topics evolve over time (see [4] for a survey of various applications). Here we consider what we believe to be a much weaker assumption – *separability*. Indeed, this property has already been identified as a natural one in the machine learning community [12] and has been empirically observed to hold in topic matrices fitted to various types of data [5].

Separability requires that each topic has some near-perfect indicator word – a word that we call the *anchor word* for this topic— that appears with reasonable probability in that topic but with negligible probability in all other topics (e.g., "soccer" could be an anchor word for the topic "sports"). We give a formal definition in Section 1.1. This property is particularly natural in the context of topic modeling, where the number of distinct words (dictionary size) is very large compared to the number of topics. In a typical application, it is common to have a dictionary size in the thousands or tens of thousands, but the number of topics is usually somewhere in the range from 50 to 100. Note that separability does *not* mean that the anchor word always occurs (in fact, a typical document may be very likely to contain *no* anchor words). Instead, it dictates that when an anchor word does occur, it is a strong indicator that the corresponding topic is in the mixture used to generate the document.

Recently, we gave a polynomial time algorithm to solve NMF under the condition that the topic matrix A is separable [2]. The intuition that underlies this algorithm is that the set of anchor words can be thought of as extreme points (in a geometric sense) of the dictionary. This condition can be used to identify all of the anchor words and then also the nonnegative factors. Ideas from this algorithm are a key ingredient in our present paper, but our focus is on the question:

Question. *What if we are not given the true matrix AW, but are instead given a few samples (say, 100 samples) from the distribution represented by each column?*

The main technical challenge in adapting our earlier NMF algorithm is that each document vector is a *very poor* approximation to the corresponding column of AW —it is *too noisy* in any reasonable measure of noise. Nevertheless, the core insights of our NMF algorithm still apply. Note that it is impossible to learn the matrix W to within arbitrary accuracy. (Indeed, this is information theoretically impossible even if we knew the topic matrix A and the distribution from which the columns of W are generated.) So we *cannot* in general give an estimator that converges to the true matrix W, and yet we *can* give an estimator that converges to the true topic matrix A! (For an overview of our algorithm, see the first paragraph of Section 3.)

We hope that this application of our NMF algorithm is just a starting point and other theoretical results will start using NMF as a replacement for SVD – just as NMF has come to replace SVD in several applied settings.

1.1 Our Results

Now we precisely define the topic modeling (learning) problem which was informally introduced above. There is an unknown *topic matrix* A which is dimension $n \times r$ (i.e. n is the dictionary size)

and each column of A is a distribution on $[n]$. There is an unknown $r \times m$ matrix W whose each column is itself a distribution (aka convex combination) on $[r]$. The columns of W are i.i.d. samples from a distribution $\mathcal{T}$ which belongs to a known family, e.g., Dirichlet distributions, but whose parameters are unknown. Thus each column of AW (being a convex combination of distributions) is itself a distribution on $[n]$, and the algorithm's input consists of N i.i.d. samples for each column of AW. Here N is the document size and is assumed to be a constant for simplicity. Our algorithm can be easily adapted to work when the documents have different sizes.

The algorithm's running time will necessarily depend upon various model parameters, since distinguishing a very small parameter from 0 imposes a lower bound on the number of samples needed. The first such parameter is a quantitative version of *separability*, which was presented above as a natural assumption in context of topic modeling.

Definition 1.1 (p-Separable Topic Matrix)**.** An $n \times r$ matrix A is *p-separable* if for each i there is some row $\pi(i)$ of A that has a single nonzero entry which is in the ith column and it is at least p.

The next parameter measures the lowest probability with which a topic occurs in the distribution that generates columns of W.

Definition 1.2 (Topic Imbalance)**.** The *topic imbalance* of the model is the ratio between the largest and smallest expected entries in a column of W, in other words, $a = \max_{i,j \in [r]} \frac{\mathbf{E}[X_i]}{\mathbf{E}[X_j]}$ where $X \in \mathbb{R}^r$ is a random weighting of topics chosen from the distribution.

Finally, we require that topics stay identifiable despite sampling-induced noise. To formalize this, we define a matrix that will be important throughout this paper:

Definition 1.3 (Topic-Topic Covariance Matrix $R(\mathcal{T})$)**.** If $\mathcal{T}$ is the distribution that generates the columns of W, then $R(\mathcal{T})$ is defined as an $r \times r$ matrix whose (i,j)th entry is $E[X_i X_j]$ where $X_1, X_2, ...X_r$ is a vector chosen from $\mathcal{T}$.

Let $\gamma > 0$ be a lower bound on the ℓ_1-condition number of the matrix $R(\mathcal{T})$. This is defined in Section 2, but for a $r \times r$ matrix it is within a factor of $\sqrt{r}$ of the smallest singular value. Our algorithm will work for any γ, but the number of documents we require will depend (polynomially) on $1/\gamma$:

Theorem 1.4 (Main)**.** *There is a polynomial time algorithm that learns the parameters of a topic model if the number of documents is at least*

$$m = \max\left\{ O\left(\frac{\log n \cdot a^4 r^6}{\epsilon^2 p^6 \gamma^2 N}\right), O\left(\frac{\log r \cdot a^2 r^4}{\gamma^2}\right)\right\},$$

where the three numbers a, p, γ are as defined above. The algorithm learns the topic-term matrix A up to additive error ϵ. Moreover, when the number of documents is also larger than $O\left(\frac{\log r \cdot r^2}{\epsilon^2}\right)$ the algorithm can learn the topic-topic covariance matrix $R(\mathcal{T})$ up to additive error ϵ.

As noted earlier, we are able to recover the topic matrix even though we do not always recover the parameters of the column distribution $\mathcal{T}$. In some special cases we can also recover the parameters of $\mathcal{T}$, e.g. when this distribution is Dirichlet, as happens in the popular *Latent Dirichlet Allocation* (LDA) model [6, 4]. In Section 4.1 we compute a lower bound on the γ parameter for the Dirichlet distribution, which allows us to apply our main learning algorithm, and also the parameters of $\mathcal{T}$ can be recovered from the co-variance matrix $R(\mathcal{T})$ (see Section 4.2).

Recently the basic LDA model has been refined to allow correlation among different topics, which is more realistic. See for example the *Correlated Topic Model* (CTM) [7] and the *Pachinko Allocation Model* (PAM) [24]. A compelling aspect of our algorithm is that it extends to these models as well: we can learn the topic matrix, even though we cannot always identify $\mathcal{T}$. (Indeed, the distribution $\mathcal{T}$ in the Pachinko is not even identifiable: two different sets of parameters can generate exactly the same distribution)

Comparison with existing approaches. (i) We rely crucially on separability. But note that this assumption is weaker in some sense than the assumptions in *all* prior works that provably learn the topic matrix. They assume a single topic per document, which can be seen as a strong separability assumption about W instead of A —in *every* column of W only one entry is nonzero. By contrast, separability only assumes a similar condition for a negligible fraction —namely, r out of n— of rows of A. Besides, it is found to actually hold in topic matrices found using current heuristics. (ii) Needless to say, existing theoretical approaches for recovering topic matrix A couldn't handle topic correlations at all since they only allow one topic per document. (iii) We remark that prior approaches that learn the span of A instead of A needed strong concentration bounds on eigenvalues of random matrices, and thus require substantial document sizes (on the order of the number of words in the dictionary!). *By contrast we can work with documents of $O(1)$ size.*

2 Tools for (Noisy) Nonnegative Matrix Factorization

2.1 Various Condition Numbers

Central to our arguments will be various notions of matrices being "far" from being low-rank. The most interesting one for our purposes was introduced by Kleinberg and Sandler [19] in the context of collaborative filtering; and can be thought of as an ℓ_1-analogue to the smallest singular value of a matrix.

Definition 2.1 (ℓ_1 Condition Number)**.** If matrix B has nonnegative entries and all rows sum to 1 then its ℓ_1 Condition Number $\Gamma(B)$ is defined as:

$$\Gamma(B) = \min_{\|x\|_1=1} \|xB\|_1.$$

If B does not have row sums of one then $\Gamma(B)$ is equal to $\Gamma(DB)$ where D is the diagonal matrix such that DB has row sums of one.

For example, if the rows of B have disjoint support then $\Gamma(B) = 1$ and in general the quantity $\Gamma(B)$ can be thought of a measure of how close two distributions on *disjoint* sets of rows can be. Note that, if x is an n-dimensional real vector, $\|x\|_2 \leq \|x\|_1 \leq \sqrt{n}\|x\|_2$ and hence (if $\sigma_{min}(B)$ is the smallest singular value of B), we have:

$$\frac{1}{\sqrt{n}}\sigma_{min}(B) \leq \Gamma(B) \leq \sqrt{m}\sigma_{min}(B).$$

The above notions of condition number will be most relevant in the context of the topic-topic covariance matrix $R(\mathcal{T})$. We shall always use γ to denote the ℓ_1 condition number of $R(\mathcal{T})$. The definition of condition number will be preserved even when we estimate the topic-topic covariance matrix using random samples.

Lemma 2.2. *When $m > 5\log r/\epsilon_0^2$, with high probability the matrix $R = \frac{1}{m}WW^T$ is entry-wise close to $R(\mathcal{T})$ with error ϵ_0. Further, when $\epsilon_0 < 1/4\gamma a r^2$ where a is topic imbalance, the matrix R has ℓ_1 condition number at least $\gamma/2$.*

Proof: Since $\mathbf{E}[W_i W_i^T] = R(\mathcal{T})$, the first part is just by Chernoff bound and union bound. The further part follows because $R(\mathcal{T})$ is γ robustly simplicial, and the error can change the ℓ_1 norm of vR for any unit v by at most $ar \cdot r\epsilon_0$. The extra factor ar comes from the normalization to make rows of R sum up to 1. ■

In our previous work on nonnegative matrix factorization [2] we defined a different measure of "distance" from singular which is essential to the polynomial time algorithm for NMF:

Definition 2.3 (β-robustly simplicial)**.** If each column of a matrix A has unit ℓ_1 norm, then we say it is *β-robustly simplicial* if no column in A has ℓ_1 distance smaller than β to the convex hull of the remaining columns in A.

The following claim clarifies the interrelationships of these latter condition numbers.

Claim 2.4. *(i) If A is p-separable then A^T has ℓ_1 condition number at least p. (ii) If A^T has all row sums equal to 1 then A is β-robustly simplicial for $\beta = \Gamma(A^T)/2$.*

We shall see that the ℓ_1 condition number for product of matrices is at least the product of ℓ_1 condition number. The main application of this composition is to show that the matrix $R(\mathcal{T})A^T$ (or the empirical version RA^T) is at least $\Omega(\gamma p)$-robustly simplicial. The following lemma will play a crucial role in analyzing our main algorithm:

Lemma 2.5 (Composition Lemma)**.** *If B and C are matrices with ℓ_1 condition number $\Gamma(B) \geq \gamma$ and $\Gamma(C) \geq \beta$, then $\Gamma(BC)$ is at least $\beta\gamma$. Specificially, when A is p-separable the matrix $R(\mathcal{T})A^T$ is at least $\gamma p/2$-robustly simplicial.*

Proof: The proof is straight forward because for any vector x, we know $\|xBC\|_1 \leq \Gamma(C)\|xB\|_1 \leq \Gamma(C)\Gamma(B)\|x\|_1$. For the matrix $R(\mathcal{T})A^T$, by Claim 2.4 we know the matrix A^T has ℓ_1 condition number at least p. Hence $\Gamma(R(\mathcal{T})A^T)$ is at least γp and again by Claim 2.4 the matrix is $\gamma p/2$-robustly simplicial. ■

2.2 Noisy Nonnegative Matrix Factorization under Separability

A key ingredient is an approximate NMF algorithm from [2], which can recover an approximate nonnegative matrix factorization $\tilde{M} \approx AW$ when the ℓ_1 distance between each row of $\tilde{M}$ and the corresponding row in AW is small. We emphasize that this is not enough for our purposes, since the term-by-document matrix $\tilde{M}$ will have a substantial amount of noise (when compared to its expectation) precisely because the number of words in a document N is much smaller than the dictionary size n. Rather, we will apply the following algorithm (and an improvement that we give in Section 5) to the Gram matrix $\tilde{M}\tilde{M}^T$.

Theorem 2.6 (Robust NMF Algorithm [2])**.** *Suppose $M = AW$ where W and M are normalized to have rows sum up to 1, A is separable and W is γ-robustly simplicial. Let $\epsilon = O(\gamma^2)$. There is a polynomial time algorithm that given $\tilde{M}$ such that for all rows $\left\|\tilde{M}^i - M^i\right\|_1 < \epsilon$, finds a W' such that $\left\|W'^i - W^i\right\|_1 < 10\epsilon/\gamma + 7\epsilon$. Further every row W'^i in W' is a row in $\tilde{M}$. The corresponding row in M can be represented as $(1 - O(\epsilon/\gamma^2))W^i + O(\epsilon/\gamma^2)W^{-i}$. Here W^{-i} is a vector in the convex hull of other rows in W with unit length in ℓ_1 norm.*

In this paper we need a slightly different goal here than in [2]. Our goal is not to recover estimates to the anchor words that are close in ℓ_1-norm but rather to recover almost anchor words (word whose row in A has almost all its weight on a single coordinate). Hence, we will be able to achieve better bounds by treating this problem directly, and we give a substitute for the above theorem. We defer the proof to Section 5.

Theorem 2.7. *Suppose $M = AW$ where W and M are normalized to have rows sum up to 1, A is separable and W is γ-robustly simplicial. When $\epsilon < \gamma/100$ there is a polynomial time algorithm that given $\tilde{M}$ such that for all rows $\|\tilde{M}^i - M^i\|_1 < \epsilon$, finds r row (almost anchor words) in $\tilde{M}$. The i-th almost anchor word corresponds to a row in M that can be represented as $(1 - O(\epsilon/\gamma))W^i + O(\epsilon/\gamma)W^{-i}$. Here W^{-i} is a vector in the convex hull of other rows in W with unit length in ℓ_1 norm.*

3 Algorithm for Learning a Topic Model: Proof of Theorem 1.4

First it is important to understand why separability helps in nonnegative matrix factorization, and specifically, the exact role played by the anchor words. Suppose the NMF algorithm is given a matrix AB. If A is p-separable then this means that A contains a diagonal matrix (up to row permutations). Thus a scaled copy of each row of B is present as a row in AB. In fact, if we knew the anchor words of A, then by looking at the corresponding rows of AB we could"read off" the corresponding row of B (up to scaling), and use these in turn to recover all of A. Thus the anchor words constitute the "key" that "unlocks" the factorization, and indeed the main step of our earlier NMF algorithm was a geometric procedure to identify the anchor words. When one is given a noisy version of AB, the analogous notion is "almost anchor" words, which correspond to rows of AB that are "very close" to rows of B; see Theorem 2.7.

Now we sketch how to apply these insights to learning topic models. Let M denote the provided term-by-document matrix, whose each column describes the empirical word frequencies in the documents. It is obtained from sampling AW and thus is an extremely noisy approximation to AW. Our algorithm starts by forming the Gram matrix MM^T, which can be thought of as an empirical word-word covariance matrix. In fact as the number of documents increases $\frac{1}{m}MM^T$ tends to a limit $Q = \frac{1}{m}E[AWW^TA]$, implying $Q = AR(\mathcal{T})A^T$. (See Lemma 3.7.) Imagine that we are given the exact matrix Q instead of a noisy approximation. Notice that Q is a product of *three* nonnegative matrices, the first of which is p-separable and the last is the transpose of the first. NMF at first sight seems too weak to help find such factorizations. However, if we think of Q as a product of *two* nonnegative matrices, A and $R(\mathcal{T})A^T$, then our NMF algorithm [2] can at least identify the anchor words of A. As noted above, these suffice to recover $R(\mathcal{T})A^T$, and then (using the anchor words of A again) all of A as well. See Section 3.1 for details.

Of course, we are not given Q but merely a good approximation to it. Now our NMF algorithm allows us to recover "almost anchor" words of A, and the crux of the proof is Section 3.2 showing that these suffice to recover provably good estimates to A and WW^T. This uses (mostly) bounds from matrix perturbation theory, and interrelationships of condition numbers mentioned in Section 2.

For simplicity we assume the following condition on the topic model, which we will see in Section 3.5 can be assumed without loss of generality:

(*) *The number of words, n, is at most $4ar/\epsilon$.*

Please see Algorithm 1: Main Algorithm for description of the algorithm. Note that R is our shorthand for $\frac{1}{m}WW^T$, which as noted converges to $R(\mathcal{T})$ as the number of documents increases.

Algorithm 1. MAIN ALGORITHM, **Output:** R and A

1. Query the oracle for m documents, where

$$m = \max\left\{O\left(\frac{\log n \cdot a^4 r^6}{\epsilon^2 p^6 \gamma^2 N}\right), O\left(\frac{\log r \cdot a^2 r^4}{\gamma^2}\right), O\left(\frac{\log r \cdot r^2}{\epsilon^2}\right)\right\}$$

2. Split the words of each document into two halves, and let $\tilde{M}$, $\tilde{M}'$ be the term-by-document matrix with first and second half of words respectively.
3. Compute word-by-word matrix $Q = \frac{4}{N^2 m}\tilde{M}\tilde{M}'^T$
4. Apply the "Robust NMF" algorithm of Theorem 2.7 to Q which returns r words that are "almost" the anchor words of A.
5. Use these r words as input to RECOVER WITH ALMOST ANCHOR WORDS to compute $R = \frac{1}{m}WW^T$ and A

Algorithm 2. RECOVER WITH TRUE ANCHOR WORDS
Input: r anchor words, **Output:** R and A

1. Permute the rows and columns of Q so that the anchor words appear in the first r rows and columns
2. Compute $DRA^T\vec{1}$ (which is equal to $DR\vec{1}$)
3. Solve for $\vec{z}$: $DRD\vec{z} = DR\vec{1}$.
4. Output $A^T = ((DRD\text{Diag}(z))^{-1}DRA^T)$.
5. Output $R = (\text{Diag}(z)DRD\text{Diag}(z))$.

3.1 Recover R and A with Anchor Words

We first describe how the recovery procedure works in an "idealized" setting (Algorthm 2,RECOVER WITH TRUE ANCHOR WORDS), when we are given the exact value of ARA^T and a set of anchor words – one for each topic. We can permute the rows of A so that the anchor words are exactly the first r words. Therefore $A^T = (D, U^T)$ where D is a diagonal matrix. Note that D is not necessarily the identity matrix (nor even a scaled copy of the identity matrix), but we do know that the diagonal entries are at least p. We apply the same permutation to the rows and columns of Q. As shown in Figure 1, if we look at the submatrix formed by the first r rows and r columns, it is exactly DRD. Similarly, the submatrix consisting of the first r rows is exactly DRA^T. We can use these two matrices to compute R and A, in this idealized setting (and we will use the same basic strategy in the general case, but need only be more careful about how we analyze how errors compound in our algorithm).

Our algorithm has exact knowledge of the matrices DRD and DRA^T, and so the main task is to recover the diagonal matrix D. Given D, we can then compute A and R (for the Dirichlet Allocation we can also compute its parameters - i.e. the $\vec{\alpha}$ so that $R(\alpha) = R$). The key idea to this algorithm is that the row sums of DR and DRA^T are the same, and we can use the row sums of DR to set up a system of linear constraints on the diagonal entries of D^{-1}.

Lemma 3.1. *When the matrix Q is exactly equal to ARA^T and we know the set of anchor words,* RECOVER WITH TRUE ANCHOR WORDS *outputs A and R correctly.*

Proof: The Lemma is straight forward from Figure 1 and the procedure. By Figure 1 we can

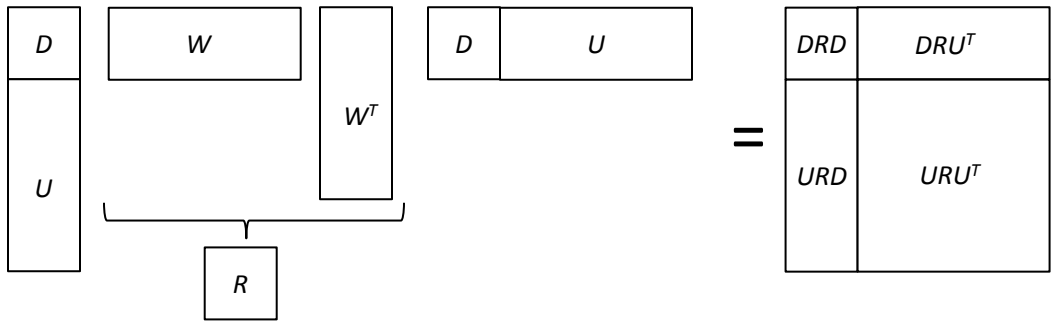

Figure 1: The matrix Q

find the exact value of DRA^T and DRD in the matrix Q. Step 2 of recover computes $DR\vec{1}$ by computing $DRA^T\vec{1}$. The two vectors are equal because A is the topic-term matrix and its columns sum up to 1, in particular $A^T\vec{1} = \vec{1}$.

In Step 3, since R is invertible by Lemma 2.2, D is a diagonal matrix with entries at least p, the matrix DRD is also invertible. Therefore there is a unique solution $\vec{z} = (DRD)^{-1}DR\vec{1} = D^{-1}\vec{1}$. Also $D\vec{z} = \vec{1}$ and hence $D\text{Diag}(z) = I$. Finally, using the fact that $D\text{Diag}(z) = I$, the output in step 4 is just $(DR)^{-1}DRA^T = A^T$, and the output in step 5 is equal to R. ■

3.2 Recover R and A with Almost Anchor Words

What if we are not given the exact anchor words, but are given words that are "close" to anchor words? As we noted, in general we cannot hope to recover the true anchor words, but even a good approximation will be enough to recover R and A.

When we restrict A to the rows corresponding to "almost" anchor words, the submatrix will not be diagonal. However, it will be close to a diagonal in the sense that the submatrix will be a diagonal matrix D multiplied by E, and E is close to the identity matrix (and the diagonal entries of D are at least $\Omega(p)$). Here we analyze the same procedure as above and show that it still recovers A and R (approximately) even when given "almost" anchor words instead of true anchor words. For clarity we state the procedure again in Algorithm 3: Recover with Almost Anchor Words. The guarantees at each step are different than before, but the implementation of the procedure is the same. Notice that here we permute the rows of A (and hence the rows and columns of Q) so that the "almost" anchor words returned by Theorem 2.6 appear first and the submatrix A on these rows is equal to DE.

Here, we still assume that the matrix Q is exactly equal to ARA^T and hence the first r rows of Q form the submatrix $DERA^T$ and the first r rows and columns are $DERE^TD$. The complication here is that $\text{Diag}(z)$ is not necessarily equal to D^{-1}, since the matrix E is not necessarily the identity. However, we can show that $\text{Diag}(z)$ is "close" to D^{-1} if E is suitably close to the identity matrix – i.e. given good enough proxies for the anchor words, we can bound the error of the above recovery procedure. We write $E = I + Z$. Intuitively when Z has only small entries E should behave like the identity matrix. In particular, E^{-1} should have only small off-diagonal entries. We make this precise through the following lemmas:

Lemma 3.2. *Let $E = I + Z$ and $\sum_{i,j} |Z_{i,j}| = \epsilon < 1/2$, then $E^{-1}\vec{1}$ is a vector with entries in the range $[1 - 2\epsilon, 1 + 2\epsilon]$.*

Algorithm 3. Recover with Almost Anchor Words
Input: r "almost" anchor words, **Output:** R and A

1. Permute the rows and columns of Q so that the "almost" anchor words appear in the first r rows and columns.
2. Compute $DERA^T\vec{1}$ (which is equal to $DER\vec{1}$)
3. Solve for $\vec{z}$: $DERE^TD\vec{z} = DER\vec{1}$.
4. Output $A^T = ((DERE^TD\text{Diag}(z))^{-1}DERA^T)$.
5. Output $R = (\text{Diag}(z)DERE^TD\text{Diag}(z))$.

Proof: E is clearly invertible because the spectral norm of Z is at most $1/2$. Let $\vec{b} = E^{-1}\vec{1}$. Since $E = I + Z$ we multiply E on both sides to get $\vec{b} + Z\vec{b} = \vec{1}$. Let b_{max} be the largest absolute value of any entry of b ($b_{max} = \max|b_i|$). Consider the entry i where b_{max} is achieved, we know $b_{max} = |b_i| \leq 1 + |(Zb)_i| \leq 1 + \sum_j |Z_{i,j}||b_j| \leq 1 + \epsilon b_{max}$. Thus $b_{max} \leq 1/(1-\epsilon) \leq 2$. Now all the entries in $Z\vec{b}$ are within 2ϵ in absolute value, and we know that $\vec{b} = \vec{1} + Z\vec{b}$. Hence all the entries of b are in the range $[1-2\epsilon, 1+2\epsilon]$, as desired. ■

Lemma 3.3. *Let $E = I + Z$ and $\sum_{i,j}|Z_{i,j}| = \epsilon < 1/2$, then the columns of $E^{-1} - I$ have ℓ_1 norm at most 2ϵ.*

Proof: Without loss of generality, we can consider just the first column of $E^{-1} - I$, which is equal to $(E^{-1} - I)\vec{e_1}$, where $\vec{e_1}$ is the indicator vector that is one on the first coordinate and zero elsewhere.

The approach is similar to that in Lemma 3.2. Let $\vec{b} = (E^{-1} - I)\vec{e_1}$. Left multiply by $E = (I+Z)$ and we obtain $\vec{b} + Z\vec{b} = -Z\vec{e_1}$. Hence $\vec{b} = -Z(\vec{b} + \vec{e_1})$. Let b_{max} be the largest absolute value of entries of $\vec{b}$ ($b_{max} = \max|b_i|$). Let i be the entry in which b_{max} is achieved. Then

$$b_{max} = |b_i| \leq |(Z\vec{b})_i| + |(Z\vec{e_1})_i| \leq \epsilon b_{max} + \epsilon$$

Therefore $b_{max} \leq \epsilon/(1-\epsilon) \leq 2\epsilon$. Further, the $\|\vec{b}\|_1 \leq \|Z\vec{e_1}\|_1 + \|Z\vec{b}\|_1 \leq \epsilon + 2\epsilon^2 \leq 2\epsilon$. ■

Now we are ready to show that the procedure Recover with Almost Anchor Words succeeds when given "almost" anchor words:

Lemma 3.4. *When the matrix Q is exactly equal to ARA^T, the matrix A restricted to almost anchor words is DE where $E - I$ has ℓ_1 norm $\epsilon < 1/10$ when viewed as a vector, procedure* Recover with Almost Anchor Words *outputs A such that each column of A has ℓ_1 error at most 6ϵ. The matrix R has additive error Z_R whose ℓ_1 norm when viewed as a vector is at most 8ϵ.*

Proof: Since Q is exactly ARA^T, our algorithm is given $DERA^T$ and $DERE^TD$ with no error. In Step 3, since D, E and R are all invertible, we have

$$\vec{z} = (DERE^TD)^{-1}DER\vec{1} = D^{-1}(E^T)^{-1}\vec{1}$$

Ideally we would want $\text{Diag}(z) = D^{-1}$, and indeed $D\text{Diag}(z) = \text{Diag}((E^T)^{-1}\vec{1})$. From Lemma 3.2, the vector $(E^T)^{-1}\vec{1}$ has entries in the range $[1-2\epsilon, 1+2\epsilon]$, thus each entry of $\text{Diag}(z)$ is within a $(1 \pm 2\epsilon)$ multiplicative factor from the corresponding entry in D^{-1}.

Consider the output in Step 4. Since D, E, R are invertible, the first output is

$$(DERE^T D\text{Diag(z)})^{-1} DERA^T = (D\text{Diag}(z))^{-1}(E^T)^{-1}A^T$$

Our goal is to bound the ℓ_1 error of the columns of the output compared to the corresponding columns of A. Notice that it is sufficient to show that the jth row of $(D\text{Diag}(z))^{-1}(E^T)^{-1}$ is close (in ℓ_1 distance) to the indicator vector $\vec{e_j}^T$.

Claim 3.5. *For each j, $\|\vec{e_j}^T(DDiag(z))^{-1}(E^T)^{-1} - \vec{e_j}^T\|_1 \leq 5\epsilon$*

Proof: Again, without loss of generality we can consider just the first row. From Lemma 3.3 $\vec{e_1}^T(E^T)^{-1}$ has ℓ_1 distance at most 2ϵ to $\vec{e_1}^T$. $(D\text{Diag}(z))^{-1}$ has entries in the range $[1-3\epsilon, 1+3\epsilon]$. And so

$$\|\vec{e_1}^T(D\text{Diag}(z))^{-1}(E^T)^{-1} - \vec{e_1}^T\|_1 \leq \|\vec{e_1}^T(D\text{Diag}(z))^{-1}(E^T)^{-1} - \vec{e_1}^T(E^T)^{-1}\|_1 + \|\vec{e_1}^T(E^T)^{-1} - \vec{e_1}^T\|_1$$

The last term can be bounded by 2ϵ. Consider the first term on the right hand side: The vector $\vec{e_1}^T(D\text{Diag}(z))^{-1} - \vec{e_1}^T$ has one non-zero entry (the first one) whose absolute value is at most 3ϵ. Hence, from Lemma 3.3 the first term can be bounded by $6\epsilon^2 \leq 3\epsilon$, and this implies the claim. ■

The first row of $(D\text{Diag}(z))^{-1}(E^T)^{-1}A^T$ is $A_1 + z^T A$ where z is a vector with ℓ_1 norm at most 5ϵ. So every column of A is recovered with ℓ_1 error at most 6ϵ.

Consider the second output of the algorithm. The output is $\text{Diag}(z)DERE^T D\text{Diag}(z)$ and we can write $\text{Diag}(z)D = I + Z_1$ and $E = I + Z_2$. The leading error are $Z_1R + Z_2R + RZ_1 + RZ_2$ and hence the ℓ_1 norm of the leading error term (when treated as a vector) is at most 6ϵ and other terms are of order ϵ^2 and can safely be bounded by 2ϵ for suitably small ϵ). ■

Finally we consider the general case (in which there is additive noise in Step 1): we are not given ARA^T exactly. We are given Q which is close to ARA^T (by Lemma 3.7). We will bound the accumulation of this last type of error. Suppose in Step 1 of RECOVER we obtain $DERA^T + U$ and $DERE^T D + V$ and furthermore the entries of U and $U\vec{1}$ have absolute value at most ϵ_1 and the matrix V has ℓ_1 norm ϵ_2 when viewed as a vector.

Lemma 3.6. *If $\epsilon, \epsilon_1, \epsilon_2$ are sufficiently small, RECOVER outputs A such that each entry of A has additive error at most $O(\epsilon + (ra\epsilon_2/p^3 + \epsilon_1 r/p^2)/\gamma)$. Also the matrix R has additive error Z_R whose ℓ_1 norm when viewed as a vector is at most $O(\epsilon + (ra\epsilon_2/p^3 + \epsilon_1 r/p^2)/\gamma)$.*

The main idea of the proof is to write $DERE^T D + V$ as $DER(E^T + V')D$. In this way the error V can be translated to an error V' on E and Lemma 3.4 can be applied. The error U can be handled similarly.

Proof: We shall follow the proof of Lemma 3.4. First can express the error term V instead as $V = (DER)V'(D)$. This is always possible because all of D, E, R are invertible. Moreover, the ℓ_1 norm of V' when viewed as a vector is at most $8ra\epsilon_2/\gamma p^3$, because this norm will grow by a factor of at most $1/p$ when multiplied by D^{-1}, a factor of at most 2 when multiplied by E^{-1} and at most $ra/\Gamma(R)$ when multiplied by R^{-1}. The bound of $\Gamma(R)$ comes from Lemma 2.2, we lose an extra ra because R may not have rows sum up to 1.

Hence $DERE^T D + V = DER(E^T + V')D$ and the additive error for $DERE^T D$ can be transformed into error in E, and we will be able to apply the analysis in Lemma 3.4.

Similarly, we can express the error term U as $U = DERU'$. Entries of U' have absolute value at most $8\epsilon_1 r/\gamma p^2$. The right hand side of the equation in step 3 is equal to $DER\vec{1} + U\vec{1}$ so the

error is at most ϵ_1 per entry. Following the proof of Lemma 3.4, we know $\mathrm{Diag}(z)D$ has diagonal entries within $1 \pm \left(2\epsilon + 16\epsilon_2/\gamma p^3 + 2\epsilon_1\right)$.

Now we consider the output. The output for A^T is equal to

$$(DER(E^T + V')D\mathrm{Diag}(z))^{-1}DER(A^T + U') = (D\mathrm{Diag}(z))^{-1}(E^T + V')^{-1}(A^T + U').$$

Here we know $(E^T + V')^{-1} - I$ has ℓ_1 norm at most $O(\epsilon + ra\epsilon_2/\gamma p^3)$ per row, $(D\mathrm{Diag}(z))$ is a diagonal matrix with entries in $1 \pm O(\epsilon + ra\epsilon_2/\gamma p^3 + \epsilon_1)$, entries of U' has absolute value $O(\epsilon_1 r/\gamma p^2)$. Following the proof of Lemma 3.4 the final entry-wise error of A is roughly the sum of these three errors, and is bounded by $O(\epsilon + (ra\epsilon_2/p^3 + \epsilon_1 r/p^2)/\gamma)$ (Notice that Lemma 3.4 gives bound for ℓ_1 norm of rows, which is stronger. Here we switched to entry-wise error because the entries of U are bounded while the ℓ_1 norm of U might be large).

Similarly, the output of R is equal to $\mathrm{Diag}(z)(DERE^TD + V)\mathrm{Diag}(z)$. Again we write $\mathrm{Diag}(z)D = I + Z_1$ and $E = I + Z_2$. The extra term $\mathrm{Diag}(z)V\mathrm{Diag}(z)$ is small because the entries of z are at most to $2/p$ (otherwise $\mathrm{Diag}(z)D$ won't be close to identity). The error can be bounded by $O(\epsilon + (ra\epsilon_2/p^3 + \epsilon_1 r/p^2)/\gamma)$. ■

Now in order to prove our main theorem we just need to show that when number of documents is large enough, the matrix Q is close to the ARA^T, and plug the error bounds into Lemma 3.6.

3.3 Error Bounds for Q

Here we show that the matrix Q indeed converges to $\frac{1}{m}AWW^TA^T = ARA^T$ when m is large enough.

Lemma 3.7. *When $m > \frac{50 \log n}{N\epsilon_Q^2}$, with high probability all entries of $Q - \frac{1}{m}AWW^TA^T$ have absolute value at most ϵ_Q. Further, the ℓ_1 norm of rows of Q are also ϵ_Q close to the ℓ_1 norm of the corresponding row in $\frac{1}{m}AWW^TA^T$.*

Proof: We shall first show that the expectation of Q is equal to ARA^T where R is $\frac{1}{m}WW^T$. Then by concentration bounds we show that entries of Q are close to their expectations. Notice that we can also hope to show that Q converges to $AR(\mathcal{T})A^T$. However in that case we will not be able to get the inverse polynomial relationship with N (indeed, even if N goes to infinity it is impossible to learn $R(\mathcal{T})$ with only one document). Replacing $R(\mathcal{T})$ with the empirical R allows our algorithm to perform better when the number of words per document is larger.

To show the expectation is correct we observe that conditioned on W, the entries of two matrices $\tilde{M}$ and $\tilde{M}'$ are independent. Their expectations are both $\frac{N}{2}AW$. Therefore,

$$\mathbf{E}[Q] = \frac{4}{mN^2}\,\mathbf{E}[\tilde{M}\tilde{M}'^T] = \frac{1}{m}\left(\frac{2}{N}\,\mathbf{E}[\tilde{M}]\right)\left(\frac{2}{N}\,\mathbf{E}[\tilde{M}'^T]\right) = \frac{1}{m}AWW^TA^T = ARA^T.$$

We still need to show that Q is close to this expectation. This is not surprising because Q is the average of m independent samples (of $\frac{4}{N^2}\tilde{M}_i\tilde{M}_i'$). Further, the variance of each entry in $\frac{4}{N^2}\tilde{M}_i\tilde{M}_i'^T$ can be bounded because $\tilde{M}$ and $\tilde{M}'$ also come from independent samples. For any i, j_1, j_2, let $v = AW_i$ be the probability distribution that $\tilde{M}_i$ and $\tilde{M}_i'$ are sampled from, then $\tilde{M}_i(j_1)$ is distributed as $Binomial(N/2, v(j_1))$ and $\tilde{M}_i'(j_2)$ is distributed as $Binomial(N/2, v(j_2))$. The variance of these two variables are less than $N/8$ no matter what v is by the properties of binomial distribution. Conditioned on the vector v these two variables are independent, thus the variance of their product is at most $\mathbf{Var}\,\tilde{M}_i(j_1)\,\mathbf{E}\,\tilde{M}_i'(j_2)^2 + \mathbf{E}\,\tilde{M}_i(j_1)^2\,\mathbf{Var}\,\tilde{M}_i'(j_2) + \mathbf{Var}\,\tilde{M}_i(j_1)\,\mathbf{Var}\,\tilde{M}_i'(j_2) \le N^3/4 + N^2/64$.

The variance of any entry in $\frac{4}{N^2}\tilde{M}_i\tilde{M}_i'^T$ is at most $4/N + 1/16N^2 = O(1/N)$. Higher moments can be bounded similarly and they satisfy the assumptions of Bernstein inequalities. Thus by Bernstein inequalities the probability that any entry is more than ϵ_Q away from its true value is much smaller than $1/n^2$.

The further part follows from the observation that the ℓ_1 norm of a row in Q is proportional to the number of appearances of the word. As long as the number of appearances concentrates the error in ℓ_1 norm must be small. The words are all independent (conditioned on W) so this is just a direct application of Chernoff bounds. ■

3.4 Proving the Main Theorem

We are now ready to prove Theorem 1.4:

Proof: By Lemma 3.7 we know when we have at least $50 \log n/N\epsilon_Q^2$ documents, Q is entry wise close to ARA^T. In this case error per row for Theorem 2.6 is at most $\epsilon_Q \cdot O(a^2r^2/p^2)$ because in this step we can assume that there are at most $4ar/p$ words (see Section 3.5) and to normalize the row we need a multiplicative factor of at most $10ar/p$ (we shall only consider rows with ℓ_1 norm at least $p/10ar$, with high probability all the anchor words are in these rows). The γ parameter for Theorem 2.7 is $p\gamma/4$ by Lemma 2.2. Thus the almost anchor words found by the algorithm has weight at least $1 - O(\epsilon_Q a^2r^2/\gamma p^3)$ on diagonals. The error for $DERE^TD$ is at most $\epsilon_Q r^2$, the error for any entry of $DERA^T$ and $DERA^T\vec{1}$ is at most $O(\epsilon_Q)$. Therefore by Lemma 3.6 the entry-wise error for A is at most $O(\epsilon_Q a^2r^3/\gamma p^3)$.

When $\epsilon_Q < \epsilon p^3\gamma/a^2r^3$ the error is bounded by ϵ . In this case we need

$$m = \max\left\{O\left(\frac{\log n \cdot a^4r^6}{\epsilon^2p^6\gamma^2N}\right), O\left(\frac{\log r \cdot a^2r^4}{\gamma^2}\right)\right\}.$$

The latter constraint comes from Lemma 2.2.

To get within ϵ additive error for the parameter α, we further need R to be close enough to the variance-covariance matrix of the document-topic distribution, which means m is at least

$$m = \max\left\{O\left(\frac{\log n \cdot a^4r^6}{\epsilon^2p^6\gamma^2N}\right), O\left(\frac{\log r \cdot a^2r^4}{\gamma^2}\right), O\left(\frac{\log r \cdot r^2}{\epsilon^2}\right)\right\}.$$

■

3.5 Reducing Dictionary Size

Above we assumed that the number of distinct words is small. Here, we give a simple gadget that shows in the general case we can assume that this is the case at the loss of an additional additive ϵ in our accuracy:

Lemma 3.8. *The general case can be reduced to an instance in which there are at most $4ar/\epsilon$ words all of which (with at most one exception) occur with probability at least $\epsilon/4ar$.*

Proof: In fact, we can collect all words that occur infrequently and "merge" all of these words into a aggregate word that we will call the *runoff word*. To this end, we call a word large if it appears more than $\epsilon mN/3ar$ times in $m = \frac{100ar \log n}{N\epsilon}$ documents, and otherwise we call it small. Indeed, with high probability all large words are words that occur with probability at least $\epsilon/4ar$ in our model. Also, all words that has a entry larger than ϵ in the corresponding row of A will appear with at least ϵ/ar probability, and is thus a large word with high probability. We can merge all

small words (i.e. rename all of these words to a single, new word). Hence we can apply the above algorithm (which assumed that there are not too many distinct words). After we get a result with the modified documents we can ignore the *runoff words* and assign 0 weight for all the small words. The result will still be correct up to ϵ additive error. ■

4 The Dirichlet Subcase

Here we demonstrate that the parameters of a Dirichlet distribution can be (robustly) recovered from just the covariance matrix $R(\mathcal{T})$. Hence an immediate corollary is that our main learning algorithm can recover both the topic matrix A and the distribution that generates columns of W in a *Latent Dirichlet Allocation* (LDA) Model [6], provided that A is separable. We believe that this algorithm may be of practical use, and provides the first alternative to local search and (unproven) approximation procedures for this inference problem [32], [11], [6].

The Dirichlet distribution is parametrized by a vector α of positive reals is a natural family of continuous multivariate probability distributions. The support of the Dirichlet Distribution is the unit simplex whose dimension is the same as the dimension of α. Let α be a r dimensional vector. Then for a vector $\theta \in \mathbb{R}^r$ in the r dimensional simplex, its probability density is given by

$$Pr[\theta|\alpha] = \frac{\Gamma(\sum_{i=1}^r \alpha_i)}{\prod_{i=1}^r \Gamma(\alpha_i)} \prod_{i=1}^r \theta_i^{\alpha_i - 1},$$

where Γ is the Gamma function. In particular, when all the α_i's are equal to one, the Dirichlet Distribution is just the uniform random distribution over the probability simplex.

The expectation and variance of θ_i's are easy to compute given the parameters α. We denote $\alpha_0 = \|\alpha\|_1 = \sum_{i=1}^r \alpha_i$, then the ratio α_i/α_0 should be interpreted as the "size" of the i-th variable θ_i, and α_0 shows whether the distributions is concentrated in the interior (when α_0 is large) or near the boundary (when α_0 is small). The first two moments of Dirichlet Distribution is listed as below:

$$\mathbf{E}[\theta_i] = \frac{\alpha_i}{\alpha_0}.$$

$$\mathbf{E}[\theta_i \theta_j] = \begin{cases} \frac{\alpha_i \alpha_j}{\alpha_0(\alpha_0+1)} & \text{when } i \neq j \\ \frac{\alpha_i(\alpha_i+1)}{\alpha_0(\alpha_0+1)} & \text{when } i = j \end{cases}.$$

Suppose the Dirichlet distribution has $\max \alpha_i / \min \alpha_i = a$ and the sum of parameters is α_0; we give an algorithm that computes close estimates to the vector of parameters α given a sufficiently close estimate to the co-variance matrix $R(\mathcal{T})$ (Theorem 4.3). Combining this with Theorem 1.4, we obtain the following corollary:

Theorem 4.1. *There is an algorithm that learns the topic matrix A with high probability up to an additive error of ϵ from at most*

$$m = \max\left\{ O\left(\frac{\log n \cdot a^6 r^8 (\alpha_0+1)^4}{\epsilon^2 p^6 N}\right), O\left(\frac{\log r \cdot a^2 r^4 (\alpha_0+1)^2}{\epsilon^2}\right)\right\}$$

documents sampled from the LDA model and runs in time polynomial in n, m. Furthermore, we also recover the parameters of the Dirichlet distribution to within an additive ϵ.

Our main goal in this section is to bound the ℓ_1-condition number of the Dirichlet distribution (Section 4.1), and using this we show how to recover the parameters of the distribution from its covariance matrix (Section 4.2).

Algorithm 4. DIRICHLET(R), **Input:** R, **Output:** α (vector of parameters)

1. Set $\alpha/\alpha_0 = R\vec{1}$.
2. Let i be the row with smallest ℓ_1 norm, let $u = R_{i.i}$ and $v = \alpha_i/\alpha_0$.
3. Set $\alpha_0 = \frac{1-u/v}{u/v-v}$.
4. Output $\alpha = \alpha_0 \cdot (\alpha/\alpha_0)$.

4.1 Condition Number of a Dirichlet Distribution

There is a well-known meta-principle that if a matrix W is chosen by picking its columns independently from a fairly diffuse distribution, then it will be far from low rank. However, our analysis will require us to prove an explicit lower bound on $\Gamma(R(\mathcal{T}))$. We now prove such a bound when the columns of W are chosen from a Dirichlet distribution with parameter vector α. We note that it is easy to establish such bounds for other types of distributions as well. Recall that we defined $R(\mathcal{T})$ in Section 1, and here we will abuse notation and throughout this section we will denote by $R(\alpha)$ the matrix $R(\mathcal{T})$ where $\mathcal{T}$ is a Dirichlet distribution with parameter α.

Let $\alpha_0 = \sum_{i=1}^{r} \alpha_i$. The mean, variance and co-variance for a Dirichlet distribution are well-known, from which we observe that $R(\alpha)_{i,j}$ is equal to $\frac{\alpha_i\alpha_j}{\alpha_0(\alpha_0+1)}$ when $i \neq j$ and is equal to $\frac{\alpha_i(\alpha_i+1)}{\alpha_0(\alpha_0+1)}$ when $i = j$.

Lemma 4.2. *The ℓ_1 condition number of $R(\alpha)$ is at least $\frac{1}{2(\alpha_0+1)}$.*

Proof: As the entries $R(\alpha)_{i,j}$ is $\frac{\alpha_i\alpha_j}{\alpha_0(\alpha_0+1)}$ when $i \neq j$ and $\frac{\alpha_i(\alpha_i+1)}{\alpha_0(\alpha_0+1)}$ when $i = j$, after normalization $R(\alpha)$ is just the matrix $D' = \frac{1}{\alpha_0+1}(\alpha \times (1, 1, ..., 1) + I)$ where $\times$ is outer product and I is the identity matrix.

Let x be a vector such that $|x|_1 = 1$ and $|D'x|_1$ achieves the minimum in $\Gamma(R(\alpha))$ and let $I = \{i|x_i \geq 0\}$ and let $J = \bar{I}$ be the complement. We can assume without loss of generality that $\sum_{i\in I} x_i \geq |\sum_{i\in J} x_i|$ (otherwise just take $-x$ instead). The product $D'x$ is $\frac{\sum x_i}{\alpha_0+1}\alpha + \frac{1}{\alpha_0+1}x$. The first term is a nonnegative vector and hence for each $i \in I$, $(D'x)^i \geq 0$. This implies that

$$|D'x|_1 \geq \frac{1}{\alpha_0+1}\sum_{i\in I} x_i \geq \frac{1}{2(\alpha_0+1)}.$$

■

4.2 Recovering the Parameters of a Dirichlet Distribution

When the variance covariance matrix $R(\alpha)$ is recovered with error ϵ_R in ℓ_1 norm when viewed as a vector, we can use Algorithm 4: DIRICHLET to compute the parameters for the Dirichlet.

Theorem 4.3. *When the variance covariance matrix $R(\alpha)$ is recovered with error ϵ_R in ℓ_1 norm when viewed as a vector, the procedure* DIRICHLET(R) *learns the parameter of the Dirichlet distribution with error at most $O(ar(\alpha_0 + 1)\epsilon_R)$.*

Proof: The α_i/α_0's all have error at most ϵ_R. The value u is $\frac{\alpha_i}{\alpha_0}\frac{\alpha_i+1}{\alpha_0+1} \pm \epsilon_R$ and the value v is $\alpha_i/\alpha_0 \pm \epsilon_R$. Since $v \geq 1/ar$ we know the error for u/v is at most $2ar\epsilon_R$. Finally we need to bound the denominator $\frac{\alpha_i+1}{\alpha_0+1} - \frac{\alpha_i}{\alpha_0} > \frac{1}{2(\alpha_0+1)}$ (since $\frac{\alpha_i}{\alpha_0} \leq 1/r \leq 1/2$). Thus the final error is at most $5ar(\alpha_0 + 1)\epsilon_R$. ■

5 Obtaining Almost Anchor Words

In this section, we prove Theorem 2.7, which we restate here:

Theorem 5.1. *Suppose $M = AW$ where W and M are normalized to have rows sum up to 1, A is separable and W is γ-robustly simplicial. When $\epsilon < \gamma/100$ there is a polynomial time algorithm that given $\tilde{M}$ such that for all rows $\|\tilde{M}^i - M^i\|_1 < \epsilon$, finds r row (almost anchor words) in $\tilde{M}$. The i-th almost anchor word corresponds to a row in M that can be represented as $(1 - O(\epsilon/\gamma))W^i + O(\epsilon/\gamma)W^{-i}$. Here W^{-i} is a vector in the convex hull of other rows in W with unit length in ℓ_1 norm.*

The major weakness of the algorithm in [2] is that it only considers the ℓ_1 norm. However in an r dimensional simplex there is another natural measure of distance more suited to our purposes: since each point is a unique convex combination of the vertices, we can view this convex combination as a probability distribution and use statistical distance (on this representation) as a norm for points inside the simplex. We will in fact need a slight modification to this norm, since we would like it to extend to points outside the simplex too:

Definition 5.2 ((δ, ϵ)-close)**.** A point M'^j is (δ, ϵ)-close to M'^i if and only if

$$\min_{c_k \geq 0, \sum_{k=1}^n c_k = 1, c_j \geq 1-\delta} \|M'^i - \sum_{k=1}^n c_k M'^k\|_1 \leq \epsilon.$$

Intuitively think of point M'^j is (δ, ϵ)-close to M'^i if M'^i is ϵ close in ℓ_1 distance to some point Q, where Q is a convex combination of the rows of M' that places at least $1 - \delta$ weight on M'^j. Notice that this definition is not a distance since it is not symmetric, but we will abuse notation and nevertheless call it a distance function. We remark that this distance is easy to compute: To check whether M'^j is (δ, ϵ)-close to M'^i we just need to solve a linear program that minimizes the ℓ_1 distance when the c vector is a probability distribution with at least $1 - \delta$ weight on j (the constraints on c are clearly all linear).

We also consider all points that a row M'^j is close to, this is called the neighborhood of M'^j.

Definition 5.3 ((δ, ϵ)-neighborhood)**.** The (δ, ϵ)-neighborhood of M'^j are the rows M'^i such that M'^j is (δ, ϵ)-close to M'^i.

For each point M'^j, we know its original (unperturbed) point M_j is in a convex combination of W^i's: $M^j = \sum_{i=1}^r A_{j,i} W^i$. Separability implies that for any column index i there is a row $f(i)$ in A whose only nonzero entry is in the ith column. Then $M^{f(i)} = W^i$ and consequently $\|M'^{f(i)} - W^i\|_1 < \epsilon$. Let us call these rows $M'^{f(i)}$ for all i the *canonical rows*. From the above description the following claim is clear.

Claim 5.4. *Every row M'^j has ℓ_1-distance at most 2ϵ to the convex hull of canonical rows.*

Proof: We have:

$$\|M'^j - \sum_{k=1}^r A_{j,k} M'^{f(k)}\|_1 \leq \|M'^j - M^j\|_1 + \|M^j - \sum_{k=1}^r A_{j,k} M^{f(k)}\|_1 + \|\sum_{k=1}^r A_{j,k}(M^{f(k)} - M'^{f(k)})\|_1$$

and we can bound the right hand side by 2ϵ. ■

The algorithm will distinguish rows that are close to vertices and rows that are far by testing whether each row is close (in ℓ_1 norm) to the convex hull of rows outside its neighborhood. In particular, we define a robust loner as:

Definition 5.5 (robust loner)**.** We call a row M'^j a robust-loner if it has ℓ_1 distance at most 2ϵ to the convex hull of rows that are outside its $(6\epsilon/\gamma, 2\epsilon)$ neighborhood.

Our goal is to show that a row is a robust loner if and only if it is close to some row in W. The following lemma establishes one direction:

Lemma 5.6. *If $A_{j,t}$ is smaller than $1 - 10\epsilon/\gamma$, the point M'^j cannot be $(6\epsilon/\gamma, 2\epsilon)$-close to the canonical row that corresponds to W^t.*

Proof: Assume towards contradiction that M'^j is $(6\epsilon/\gamma, 2\epsilon)$-close to the canonical row M'^i which is a perturbation of W^t. By definition there must be probability distribution $c \in \mathbb{R}^n$ over the rows such that $c_j \geq 1 - 6\epsilon/\gamma$, and $\|M'^i - \sum_{k=1}^n c_k M'^k\|_1 \leq 2\epsilon$. Now we instead consider the unperturbed matrix M, since every row of M' is ϵ close (in ℓ_1 norm) to M we know $\|M^i - \sum_{k=1}^n c_k M^k\|_1 \leq 4\epsilon$. Now we represent M^i and $\sum_{k=1}^n c_k M^k$ as convex combinations of rows of W and consider the coefficient on W^t. Clearly $M^i = W^t$ so the coefficient is 1. But for $\sum_{k=1}^n c_k M^k$, since $c_j \geq 1 - 6\epsilon/\gamma$ and the coefficient $A_{j,t} \leq 1 - 10\epsilon/\gamma$, we know the coefficient of W^t in the sum must be strictly smaller than $1 - 10\epsilon/\gamma + 6\epsilon/\gamma = 1 - 4\epsilon/\gamma$. By the robustly simplicial assumption M^i and $\sum_{k=1}^n c_k M^k$ must be more than $4\epsilon/\gamma \cdot \gamma = 4\epsilon$ apart in ℓ_1 norm, which contradicts our assumption. ■

As a corollary:

Corollary 5.7. *If $A_{j,t}$ is smaller than $1 - 10\epsilon/\gamma$ for all t, the row M'^j cannot be a robust loner.*

Proof: By the above lemma, we know the canonical rows are not in the $(6\epsilon/\gamma, 2\epsilon)$ neighborhood of M'^j. Thus by Claim 5.4 the row is close to the convex hull of canonical rows and cannot be a robust loner. ■

Next we prove the other direction: a canonical row is necessarily a robust loner:

Lemma 5.8. *All canonical rows are robust loners.*

Proof: Suppose M'^i is a canonical row that corresponds to W^t. We first observe that all the rows that are outside the $(6\epsilon/\gamma, 2\epsilon)$ neighborhood of M'^j must have $A_{j,t} < 1 - 6\epsilon/\gamma$. This is because when $A_{j,t} \geq 1 - 6\epsilon/\gamma$ we have $M^j - \sum_{k=1}^r A_{j,t} W^t = \vec{0}$. If we replace M^j by M'^j and W^t by the corresponding canonical row, the distance is still at most 2ϵ and the coefficient on M'^i is at least $1 - 6\epsilon/\gamma$. By definition the corresponding row M'^j must be in the neighborhood of M'^i.

Now we try to represent M'^i with convex combination of rows that has $A_{j,t} < 1 - 6\epsilon/\gamma$. However this is impossible because every point in the convex combination will also have weight smaller than $1 - 6\epsilon/\gamma$ on W^t, while M^i has weight 1 on W^t. The ℓ_1 distance between M_i and the convex combination of the M_j's where $A_{j,t} < 1 - 6\epsilon/\gamma$ is a least 6ϵ by robust simplicial property. Even when the points are perturbed by ϵ (in ℓ_1) the distance can change by at most 2ϵ and is still more than 2ϵ. Therefore M'^i is a robust loner. ■

Now we can prove the main theorem of this section:

Proof: Suppose we know γ and $100\epsilon < \gamma$.When γ is so small we have the following claim:

Claim 5.9. *If $A_{j,t}$ and $A_{i,l}$ is at least $1 - 10\epsilon/\gamma$, and $t \neq l$, then M'^j cannot be $(10\epsilon/\gamma, 2\epsilon)$-close to M'^i and vice versa.*

The proof is almost identical to Lemma 5.6. Also, the canonical row that corresponds to W^t is $(10\epsilon/\gamma, 2\epsilon)$ close to all rows with $A_{j,t} \geq 1 - 10\epsilon/\gamma$. Thus if we connect two robust loners when one is $(10\epsilon/\gamma, 2\epsilon)$ close to the other, the connected component of the graph will exactly be a partition according to the row in W that the robust loner is close to. We pick one robust loner in each connected component to get the almost anchor words.

Now suppose we don't know γ. In this case the problem is we don't know what is the right size of neighborhood to look at. However, since we know $\gamma > 100\epsilon$, we shall first run the algorithm with $\gamma = 100\epsilon$ to get r rows W' that are very close to the true rows in W. It is not hard to show that these rows are at least $\gamma/2$ robustly simplicial and at most $\gamma + 2\epsilon$ robustly simplicial. Therefore we can compute the $\gamma(W')$ parameter for this set of rows and use $\gamma(W') - 2\epsilon$ as the γ parameter. ■

6 Maximum Likelihood Estimation is Hard

Here we prove that computing the Maximum Likelihood Estimate (MLE) of the parameters of a topic model is NP-hard. We call this problem the Topic Model Maximum Likelihoood Estimation (TM-MLE) problem:

Definition 6.1 (TM-MLE)**.** Given m documents and a target of r topics, the TM-MLE problem asks to compute the topic matrix A that has the largest probability of generating the observed documents (when the columns of W are generated by a uniform Dirichlet distribution).

Surprisingly, this appears to be the first proof that computing the MLE estimate in a topic model is indeed computationally hard, although its hardness is certainly to be expected. On a related note, Sontag and Roy [29] recently proved that *given the topic matrix* and a document, computing the Maximum A Posteriori (MAP) estimate for the distribution on topics that generated this document is NP-hard. Here we will establish that TM-MLE is NP-hard via a reduction from the MIN-BISECTION problem: In MIN-BISECTION the input is a graph with n vertices (n is an even integer), and the goal is to partition the vertices into two equal sized sets of $n/2$ vertices each so as to minimize the number of edges crossing the cut.

Theorem 6.2. *There is a polynomial time reduction from MIN-BISECTION to TM-MLE ($r = 2$).*

Proof: Suppose we are given an instance G of the MIN-BISECTION problem with n vertices and m edges. We will now define an instance of the TM-MLE problem. First, we set the number of words to be n. For each word i, we construct $N = \lceil 200m^3 \log n \rceil$ documents each of which contain the word i twice and no other words. For each edge in the graph G, we construct a document whose two words correspond to the endpoints of the edge.

Suppose that $x = (x_1, x_2)^T$ is generated by the Dirichlet distribution $Dir(1, 1)$. Consequently the probability that words i and j appear in a document with only two words is exactly $(A^i x) \cdot (A^j x)$. We can take the expectation of this term over the Dirichlet distribution $Dir(1, 1)$ and hence the probability that a document (with exactly two words) contains the words i and j is

$$\mathbf{E}[(A^i x) \cdot (A^j x)] = \frac{1}{3}(A^i \cdot A^j) + \frac{1}{6}(A_1^i A_2^j + A_1^j A_2^i)$$

In the TM-MLE problem, our goal is to maximize the following objective function (which is the log of the probability of generating the collection of documents):

$$OBJ = \sum_{\text{document} = \{i,j\}} \log \left[\frac{1}{3}(A^i \cdot A^j) + \frac{1}{6}(A_1^i A_2^j + A_1^j A_2^i) \right].$$

For any bisection, we define a canonical solution: the first topic is uniform on all words on one side of the bisection and the second topic is uniform on all words on the other side of the bisection.To prove the correctness of our reduction, a key step is to show that any candidate solution to the MLE problem must be close to a canonical solution. In particular, we show the following:

1. The rows A^i have almost the same ℓ_1 norm.

2. In each row A^i, almost all of the weight will be in one of the two topics.

Indeed, canonical solutions have large objective value. Any canonical solution has objective value at least $-Nn \log 3n^2/4 - m \log 3n^2/2$ (this is because documents with same words contribute $-\log 3n^2/4$ and documents with different words contribute at least $-\log 3n^2/2$).

Recall, in our reduction N is large. Roughly, if one of the rows has ℓ_1 norm that is bounded away from $2/n$ by at least $1/20nm$, the contribution (to the objective function) of documents with a repeated word will decrease significantly and the solution cannot be optimal. To prove this we use the fact that the function $\log x^2 = 2 \log x$ is concave. Therefore when one of the rows has ℓ_1 norm more than $2/n + 1/20nm$, the optimal value for documents with a repeated word will be attained when all other rows have the same ℓ_1 norm $2/n - 1/20nm(n-1)$. Using a Taylor expansion, we conclude that the sum of terms for documents with a repeated word will decrease by at least $N/50m^2$ which is much larger than any effect the remaining m documents can recoup. In fact, an identical argument establishes that in each row A^i, the topic with smaller weight will always have weight smaller than $1/20nm$.

Now we claim that among canonical solutions, the one with largest objective value corresponds to a minimum bisection. The proof follows from the observation that the value of the objective function is $-Nn \log 3n^2/4 - k \log 3n^2/2 - (m-k) \log 3n^2/4$ for canonical solutions, where k is the number of edges cut by the bisection. In particular, the objective function of the minimum bisection will be at least an additive $\log 2$ larger than the objective function of a non-minimum bisection.

However, even if the canonical solution is perturbed by $1/20nm$, the objective function will only change by at most $m \cdot 1/10m = 1/10$, which is much smaller than $\frac{\log 2}{2}$. And this completes our reduction. ■

We remark that the canonical solutions in our reduction are all *separable*, and hence this reduction applies even when the topic matrix A is known (and required) to be separable. So, even in the case of a separable topic matrix, it is NP-hard to compute the MLE.

7 Conclusions

We expect that versions of our algorithm may indeed be practical, and are investigating this possibility. Our machine learning colleagues suggest that real-life topic matrices satisfy even stronger *separability* assumptions, e.g., the presence of *many* anchor words per topic instead of a single one. This is a promising suggestion, but leveraging it in our algorithm is an open problem.

Is separability necessary for allowing polynomial-time algorithms for the learning problems considered here? In other words, is the problem difficult if the topic matrix A is not *separable*? Average-case intractability seems more plausible here than NP-completeness.

Acknowledgements

We thank Dave Blei, Ravi Kannan, David Minmo, Sham Kakade, David Sontag for many helpful discussions throughout various stages of this work.

References

[1] A. Anandkumar, D. Hsu and S. Kakade. A method of moments for mixture models and hidden Markov models. Arxiv, 2012. 1

[2] S. Arora, R. Ge, R. Kannan and A. Moitra. Computing a nonnegative matrix factorization – provably. *STOC* 2012, to appear. 1, 2.1, 2.2, 2.6, 2.2, 3, 5

[3] Y. Azar, A. Fiat, A. Karlin, F. McSherry and J. Saia. Spectral analysis of data. *STOC*, pp. 619–626, 2001. 1

[4] D. Blei. Introduction to probabilistic topic models. *Communications of the ACM*, pp. 77–84, 2012. 1, 1.1

[5] D. Blei. Personal communication. 1

[6] D. Blei, A. Ng and M. Jordan. Latent Dirichlet Allocation. *Journal of Machine Learning Research*, pp. 993–1022, 2003. Preliminary version in *NIPS* 2001. 1, 1.1, 4

[7] D. Blei and J. Lafferty. A correlated topic model of Science. *Annals of Applied Statistics*, pp. 17–35, 2007. 1.1

[8] D. Blei and J. Lafferty. Dynamic topic models. *ICML*, pp. 113–120, 2006.

[9] J. Cohen and U. Rothblum. Nonnegative ranks, decompositions and factorizations of nonnegative matices. *Linear Algebra and its Applications*, pp. 149–168, 1993.

[10] S. Deerwester, S. Dumais, T. Landauer, G. Furnas and R. Harshman. Indexing by latent semantic analysis. *JASIS*, pp. 391–407, 1990.

[11] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM Algorithm. *J. Roy. Statist. Soc. Ser. B*, pp. 1–38, 1977. 1, 4

[12] D. Donoho and V. Stodden. When does non-negative matrix factorization give the correct decomposition into parts? *NIPS*, 2003. 1

[13] G. Golub and C. van Loan. *Matrix Computations* The Johns Hopkins University Press, 1996.

[14] N. Gravin, J. Lasserre, D. Pasechnik and S. Robins. The inverse moment problem for convex polytopes. *Discrete and Computation Geometry*, 2012, to appear.

[15] T. Hofmann. Probabilistic latent semantic analysis. *UAI*, pp. 289–296, 1999. (document), 1

[16] P. Hoyer. Non-negative matrix factorization with sparseness constraints. *Journal of Machine Learning Research*, pp. 1457–1469, 2004.

[17] A. Hyvärinen, J. Karhunen and E. Oja. *Independent Component Analysis*. Wiley Interscience, 2001.

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

[19] J. Kleinberg and M. Sandler. Using mixture models for collaborative filtering. *JCSS*, pp. 49–69, 2008. Preliminary version in *STOC* 2004. 1, 2.1

[20] J. Kleinberg and M. Sandler. Convergent algorithms for collaborative filtering. *ACM EC*, pp. 1–10, 2003. 1

[21] R. Kumar, P. Raghavan, S. Rajagopalan and A. Tomkins. Recommendation systems: a probabilistic analysis. *JCSS*, pp. 42–61, 2001. Preliminary version in *FOCS* 1998.

[22] D. Lee and H. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, pp. 788-791, 1999. 1

[23] D. Lee and H. Seung. Algorithms for non-negative matrix factorization. *NIPS*, pp. 556–562, 2000.

[24] W. Li and A. McCallum. Pachinko Allocation: DAG-structured mixture models of topic correlations. *ICML*, pp. 633-640, 2007. 1.1

[25] J. Matousek. *Lectures on Discrete Geometry*. Springer, 2002.

[26] F. McSherry. Spectral partitioning of random graphs. *FOCS*, pp. 529–537, 2001. 1

[27] C. Papadimitriou, P. Raghavan, H. Tamaki and S. Vempala. Latent semantic indexing: a probabilistic analysis. *JCSS*, pp. 217–235, 2000. Preliminary version in *PODS* 1998. (document), 1

[28] R. A. Redner and H. F. Walker. Mixture densities, maximum likelihood and the EM Algorithm. *SIAM Rev.* , pp. 195-239, 1984. 1

[29] D. Sontag and D. Roy. Complexity of inference in Latent Dirichlet Allocation. *NIPS*, pp. 1008–1016, 2011. 6

[30] W. Xu and X. Liu and Y. Gong. Document clustering based on non-negative matrix factorization. *SIGIR*, pp. 267–273, 2003.

[31] S. Vavasis. On the complexity of nonnegative matrix factorization. *SIAM Journal on Optimization*, pp. 1364-1377, 2009. 1

[32] M. Wainwright and M. Jordan. Graphical models, exponential families, and variational inference. *Foundations and Trends in Machine Learning*, pp. 1–305, 2008. 1, 4