Learning Topic Models — Provably and Efficiently

Sanjeev Arora
Princeton University
Princeton, NJ
arora@cs.princeton.edu

Rong Ge
Duke University
Durham, NC
rongge@cs.duke.edu

Yoni Halpern
Google
Cambridge, MA
yhalpern@gmail.com

David Mimno
Cornell University
Ithaca, NY
mimno@cornell.edu

Ankur Moitra
MIT
Cambridge, MA
moitra@mit.edu

David Sontag
MIT
Cambridge, MA
dsontag@mit.edu

Yichen Wu
ychwu5@gmail.com

Michael Zhu
Stanford University
Stanford, CA
mhzhu@cs.stanford.edu

1. INTRODUCTION

Today, we have both the blessing and the curse of being overloaded with information. Never before has text been more important to how we communicate, or more easily available. But massive text streams far outstrip anyone's ability to read. We need automated tools that can help make sense of their thematic structure, and find strands of meaning that connect documents, all without human supervision. Such methods can also help us organize and navigate large text corpora. Popular tools for this task range from *Latent Semantic Analysis* (LSA) [8] which uses standard linear algebra, to *deep learning* which relies on non-convex optimization. This paper concerns *topic modeling* which posits a simple probabilistic model of how a document is generated. We give a formal description of the generative model at the end of the section, but next we will outline its important features.

Topic modeling represents each document as a *bag of words* whereby all notions of grammar and syntax are discarded, and each document is associated with its vector of word counts. The central assumption is that there is a fixed set of *topics* — numbering, say, a couple hundred — that are shared and recur in different proportions in each document. For example, a news article about legislation related to retirement accounts might be represented as a mixture of 0.7 of the topic *politics* and 0.3 of the topic *personal finance*. Furthermore, each topic induces a distribution on words in the vocabulary. Note that a word like *account* can occur in several topics: it could refer to a financial product (a bank account) or a story (a fictional account), but the probability that it is assigned would likely vary across topics. Finally, the model specifies that each document is generated by first picking its topic proportions from some distribution, and then sampling each word from the document-specific distribution on words. In the above example, each word would be picked independently from *politics* with probability 0.7 and from *personal finance* with probability 0.3. The goal in topic modeling is, when given a large enough collection of documents, to discover the underlying set of topics used to generate them. Moreover, we want algorithms that are both fast and accurate.

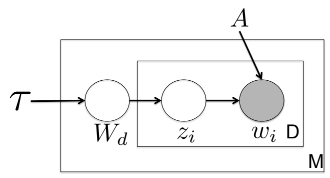
This generative model is a simplistic account of how documents are created. Nevertheless, for a wide range of applications in text analysis, methods based on this model do indeed recover meaningful topics. We give an example of a randomly chosen set of topics recovered by our algorithm, when run on a collection of New York Times articles, in Figure 1. These tools have also found many applications in summarization and exploratory data analysis. In fact, the models described above are not just limited to text analysis and have been used to recover semantic structure in various biological datasets, including fMRI images of brain activity. Variants of this model have also been used in linguistic and humanities applications. See [5] for a thorough survey.

Traditional methods for learning the parameters of a topic model are based on maximizing a *likelihood objective*. Such approaches are popular when learning the parameters of various other probabilistic models, too. However even in the case of topic models with just *two* topics, this optimization

Figure 1: Examples of topics automatically extracted from a collection of New York Times articles. Each row contains words from one topic in descending order by probability.

anthrax, official, mail, letter, worker, attack
president, clinton, white_house, bush, official, bill_clinton
father, family, **elian**, boy, court, miami
oil, prices, percent, million, market, united_states
microsoft, company, computer, system, window, software
government, election, mexico, political, vicente_fox, president
fight, **mike_tyson**, round, right, million, champion
right, law, president, george_bush, senate, **john_ashcroft**

This work appeared as "A Practical Algorithm for Topic Modeling with Provable Guarantees" (Arora, Ge, Halpern, Mimno, Moitra, Sontag, Wu, Zhu, ICML 2013).



For d from $1..M$:
- Sample $W_d \sim \tau$
- For i from $1...D$:
  - Sample $z_i \sim W_d$
  - Sample $w_i \sim A_{z_i}$

Figure 2: Generative model used in topic modeling.

problem is NP-hard [4]. At best, the approaches used in practice are known to converge to the true solution eventually but we know of no good guarantees on the running time needed to fit the parameters up to some desired precision. These gaps in our understanding are not only a theoretical issue but also a practical one: the seemingly large running times of these algorithms means that learning 1000 topics from 20 million news articles requires a distributed algorithm and 100 dedicated computers [1].

Recently, several groups of researchers have designed new algorithms that have provable guarantees. These algorithms run in times that scale as a fixed polynomial in the number of documents and the inverse of the desired precision [4, 2]. Our primary focus is on the algorithm of Arora et al. [4] which is based on a seemingly realistic assumption — termed *separability* — about the structure of topics. The subsequent work of Anandkumar et al. [2] removes this assumption, but requires that the topics are essentially uncorrelated and seems to be quite sensitive to violations of this assumption. The contribution of the present article is to show that some of these new theoretical algorithms can be adapted to yield highly practical tools for topic modeling, that compete with state of the art approximate likelihood approaches in terms of the solution quality and run in a fraction of the time. At the same time, the provable guarantees continue to hold for our simplified algorithms.

The Model

Here we formally state the model we will be interested in. We will rely on these definitions for much of the discussion that follows. Let V denote the number of words in the vocabulary. Let K denote the number of topics. And let M denote the number of documents, and D denote their length. (In general, one can allow documents to be of varying lengths and one could even specify a distribution from which their length is drawn). Each of the K topics is identified with a distribution over words. We will represent these distributions as V-dimensional vectors $A_1, A_2, ..., A_K$ whose entries are nonnegative and sum to one.

Each document d is generated by picking its topic proportions W_d from a distribution τ. The topic proportions can also be viewed as a vector, but in K-dimensions where the value in coordinate i represents the proportion of topic i present in document d. Finally, each word is independently sampled by choosing its topic $z_j \in \{1, 2, .., K\}$ according to W_d, and then sampling it from that topic's distribution over words $w_j \sim A_{z_j}$. We remark that this formulation is very general and includes most widely used probabilistic topic models, such as the *Latent Dirichlet Allocation Model* (LDA) [7] where τ is a Dirichlet distribution, as well as subsequent extensions that allow topics to be positively or negatively correlated such as the *Correlated Topic Model* (CTM) [6] where τ is a logistic Normal distribution. See also Figure 2.

1.1 Likelihood-based Methods

Here we expand upon some of the computational difficulties of working with likelihood-based methods. The traditional approach to fitting the parameters of topic models is via *maximum likelihood estimation*, whereby we seek a set of K topics, $\{A_1, A_2, ..., A_K\}$, as well as a description of the distribution τ, that maximize the likelihood of the entire collection having been generated by the model. This is a difficult optimization problem because the likelihood objective is non-convex, with many local maxima. Optimizing non-convex functions is notoriously difficult, and standard local-search based techniques like Expectation-Maximization [9] or gradient ascent are only guaranteed to converge to a local maximum, which may be much worse in terms of the objective value than the global optimum. Even worse, evaluating the likelihood function is itself difficult due to the large number of latent variables, namely the topic proportions of each document, W_d, and the topic assignments of each word, z_j. To evaluate the likelihood of even a single document requires integrating over all possible topic-proportions, a high-dimensional integral with no closed form, as well as summing over an exponential number of possible topic assignments for the words in the document.

Other previous works attempt to solve approximate versions of the maximum likelihood problem. For example, the variational-EM approach [7, 14] maximizes an objective that lower bounds the likelihood objective, but cannot guarantee that the solution is close to the optimum of the likelihood objective itself. The Markov Chain Monte Carlo (MCMC) approach [13] uses Markov chains tailored to generate samples from the posterior distribution of the parameters conditioned on the observed collection of documents, but suffers from well-known drawbacks: It is difficult to assess convergence, and no polynomial bounds on its mixing time are known in settings of interest. These approximations to the maximum likelihood objective are, in a sense, necessary, since recent work has shown that even for just two topics finding the maximum likelihood solution is NP-hard [4]. Another reason that these methods tend to be slow in practice, is that they contain an inner loop in which they perform approximate inference, determining which topics are likely present in each document in the collection. This is also known to be NP-hard [25]. Thus, we seek a principled new approach that can circumvent both the hardness of maximum likelihood estimation and inference.

1.2 The Method of Moments

The challenges of working with the maximum likelihood estimator motivate us to investigate other consistent estimators, ones that hopefully can be computed more efficiently. As we mentioned earlier, we build on recent algorithms [4, 2] that provably recover the parameters of a topic model in polynomial time. These approaches are based on the method of moments — which is originally due to Pearson [23] — but has fallen out of favor in the statistics community in large part because it seems to require more samples than the likelihood-based approaches championed by Fisher. However, in the modern age of big data, statistical efficiency is not as pressing an issue as computational efficiency. With this in mind, it seems time to revisit the method of moments.

The key concept behind the method of moments is to set up a system of equations relating quantities that can be easily estimated from data (such as means or averages) and the

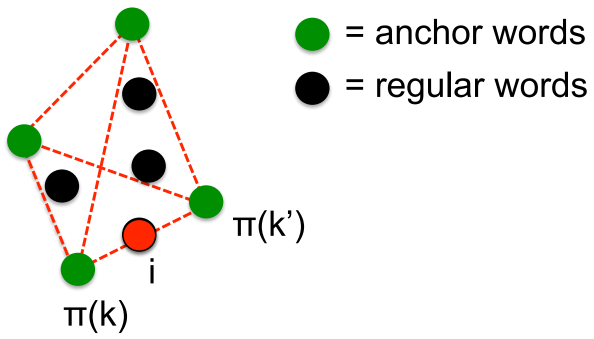


Figure 3: The rows of Q are vectors in V-dimensions, and their convex hull is a K-simplex whose vertices are anchor rows. Here $Q_i = \frac{1}{2}Q_{\pi(k)} + \frac{1}{2}Q_{\pi(k')}$ and this implies that the posterior distribution $P(z_1 = *|w_1 = i)$ assigns $\frac{1}{2}$ to $z_1 = k$ and $\frac{1}{2}$ to $z_1 = k'$ and zero to every other topic.

parameters of model. One has to choose the set of equations carefully so as to guarantee the identifiability of model parameters and to ensure that the system of equations can be solved efficiently. In recent years, the method of moments has been used to give computationally efficient algorithms for a variety of fundamental statistical estimation problems such as learning mixtures of Gaussians [15].

Let us describe the approach in the context of topic modeling, working with second order moments. Let Q be a $V \times V$ matrix where the entry $Q_{j,j'}$ denotes the probability that the first and second words in a randomly generated document are word j and word j' respectively. It turns out that this matrix can be expressed as the product of three entry-wise nonnegative matrices. Let R denote a $K \times K$ matrix where the entry $R_{i,i'}$ represents the probability that the first and second word are sampled from topic i and topic i' respectively. Finally let A denote the $V \times K$ matrix whose columns are $A_1, A_2, ..., A_K$. Then it can be shown that $Q = ARA^T$. Suppose for the moment that we could accurately estimate the entries of Q.

A naive attempt to apply the method of moments runs into its own computational difficulties when one attempts to solve the system of non-linear equations. In particular, we are faced with a matrix decomposition problem where our goal is to express Q as a product of entry-wise nonnegative matrices as above. This is closely related to the *nonnegative matrix factorization* problem and is known to be NP-hard [26, 3]. The approach of Arora et al. [4] is to make use of algorithms that solve nonnegative matrix factorization under a certain assumption that seems natural in the context of topic modeling. We describe this assumption and its rationale next.

Separability

The guiding assumption behind the algorithm of Arora et al. [4] is a notion called *separability* [11]. More precisely, this assumption stipulates that topics can be reliably distinguished from one another via *anchor words* — which, in the context of topic models, are specialized words that are specific to a single topic. For example, if the word *401k* occurs in a document then it is a strong indicator that the document is at least partially about *personal finance*. Natural language seems to contain many such unambiguous words. The condition of separability requires that each topic contains at least one (unknown) anchor word. We provide various experiential evidence showing that models fit to real-world data sets contain many anchor words.

Arora et al. [3] gave an algorithm for solving nonnegative matrix factorization under the separability assumption. In a subsequent paper, Arora et al. [4] showed that such an algorithm can be used to provably learn the parameters of a separable topic model. While theoretically important, these algorithms (as stated) were far from practical: the runtime is a large polynomial, and the algorithm itself is sensitive to violations of the modeling assumptions, learning poor quality topics when run on real-world data collections. The current paper addresses these issues, presenting a variant of the above algorithm that achieves state of the art performance and runs orders of magnitude faster than approximate likelihood based approaches. Along the way, we also give a faster algorithm for solving separable nonnegative matrix factorization.

We remark that separability is not the only assumption that allows for polynomial time recovery of topic-models. Anandkumar et al. [2] give a provable algorithm for topic modeling based on third-order moments and tensor decomposition that does not require separability but instead requires that topics are essentially uncorrelated. Although standard topic models like LDA [7] assume this property, there is strong evidence that real-world topics are dependent [6, 19]. For example, the topics *economics* and *politics* are more likely to co-occur than *economics* and *cooking*.

2. THE ANCHOR WORDS ALGORITHM

2.1 From Probability to Geometry

Separable topic models have various important probabilistic and geometric properties. These properties will form the foundation for our algorithm. We will work with simple statistics measuring how often various pairs of words co-occur in a document. Recall that the matrix Q denotes the co-occurrence probabilities of pairs of words. In this section it is more convenient to consider the *conditional probabilities* $\bar{Q}$, where $\bar{Q}_{i,j}$ is the probability of the second word being j conditioned on the first word being i. The matrix $\bar{Q}$ is just a row normalized version of Q whose rows sum up to 1.

It is useful to consider this data geometrically. We can view the rows of $\bar{Q}$ as points in V-dimensional space. Moreover we will call a row of $\bar{Q}$ an *anchor row* if it corresponds to an anchor word. A simplified illustration of anchor and non-anchor rows is given in Figure 3. The key insight behind our algorithm is the following fact. Recall that a vector u is said to be in the convex hull of vectors $v_1, v_2, \ldots v_d$ if it can be written as $u = \sum_i \lambda_i v_i$ where the λ_i's are nonnegative and sum to one.

LEMMA 1. *If the topic matrix is separable, then each row of $\bar{Q}$ is in the convex hull of the anchor rows.*

This geometric property motivates our simple, greedy algorithm for identifying the anchor words. First we sketch a proof of this lemma through elementary manipulations on various conditional probabilities.

Consider a randomly generated document, and let w_1 and w_2 be random variables that denote its first and second words respectively. Furthermore let z_1 and z_2 denote

Algorithm 1. FindAnchors

```
1: **Compute co-occurrences.** Let $N_d$ be the length of document $d$, and $N_d(i)$ be the number of occurrences of word $i$ in document $d$.
     $$\hat{Q}_{i,j} = \frac{1}{M}\sum_d \frac{2}{N_d(N_d-1)} N_d(i)N_d(j)$$
     $$\hat{Q}_{i,i} = \frac{1}{M}\sum_d \frac{2}{N_d(N_d-1)} (N_d^2(i) - N_d(i))$$
2: Let $\hat{\bar{Q}}$ be a row normalization of $\hat{Q}$. Rows of $\hat{\bar{Q}}$ sum up to 1.
3: **for** $k = 1$ TO $K$ **do**
4:     Choose the row in $\hat{\bar{Q}}$ furthest from the affine span of the anchor rows chosen so far.
5: **end for**
6: Return the chosen anchor words
```

their latent topic assignments. We will think of the generative procedure as first picking W_d from τ and then picking $z_1, z_2 \in [K]$ independently according to W_d. Once these topic assignments are fixed, the words w_1 and w_2 are independently sampled from A_{z_1} and A_{z_2} respectively. We will use the notation $\pi(k)$ to denote an anchor word for topic k. Then the definition of an anchor word gives us:

$$P(z_1 = k'|w_1 = \pi(k)) = \begin{cases} 1 & k' = k \\ 0 & else \end{cases}$$

This follows because when an anchor word is observed, there is only one topic that could have generated it! Moreover let $\bar{Q}_i$ denote the ith row of $\bar{Q}$. Then the jth coordinate of $\bar{Q}_i$ is $P(w_2 = j|w_1 = i)$. We will use the shorthand $\bar{Q}_i = P(w_2 = *|w_1 = i)$. It follows that

$$\bar{Q}_{\pi(k)} = P(w_2 = *|w_1 = \pi(k)) = P(w_2 = *|z_1 = k)$$

And finally we can write

$$\begin{aligned} \bar{Q}_i &= \sum_{k'} P(w_2 = *|w_1 = i, z_1 = k')P(z_1 = k'|w_1 = i) \\ &= \sum_{k'} P(w_2 = *|w_1 = \pi(k'))P(z_1 = k'|w_1 = i) \end{aligned}$$

This formula explicitly represents $\bar{Q}_i$ as a convex combination of the anchor rows, but moreover we see that the convex combination is given by the conditional probabilities $P(z_1 = k'|w_1 = i)$ of which topic generated word $w_1 = i$. Thus our strategy is to find the anchor rows, and then solve a low-dimensional convex program to represent every non-anchor row as a convex combination of the anchor rows to find $P(z_1 = k'|w_1 = i)$. From there, we can use Bayes' rule to compute $P(w_1 = i|z_1 = k')$ which are exactly the parameters (except for the hyperparameters) of our topic model.

2.2 Finding the Anchor Words

We give a simple, greedy algorithm called `FindAnchors` that provably finds the K anchor words (one for each topic) given the empirical estimate $\hat{\bar{Q}}$ of the matrix $\bar{Q}$ defined in the previous subsection. We will analyze this algorithm in the noiseless setting where $\hat{\bar{Q}} = \bar{Q}$, but what is important about this algorithm is its behavior in the presence of noise. In that setting, it can be shown that `FindAnchors` recovers

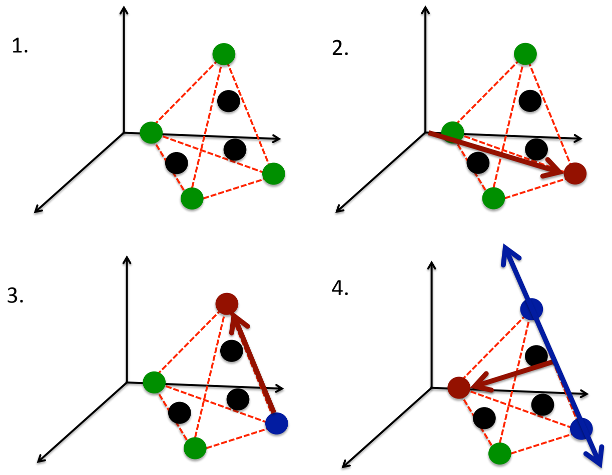


Figure 4: The first three steps of `FindAnchors` consist of finding a starting point furthest from the origin, finding the furthest point from the initial point, and finding the furthest point from the line defined by the first two points.

near anchor words — i.e. words whose row in $\hat{\bar{Q}}$ is close in ℓ_1 distance to some anchor word. We need these latter types of guarantees to quantify how much data we need to get estimates that are provable close to the true parameters of the topic model.

The algorithm builds up a set of anchor words greedily, and starts by choosing the row farthest from the origin. Then it iteratively add points that maximize distance from the affine span of the previously collected points. This procedure can also be seen as iteratively growing the simplex, adding vertices that greedily maximize the enclosed volume. While the general problem of choosing K rows of a matrix $\hat{\bar{Q}}$ to maximize the enclosed volume is NP-hard, it becomes easy when the points are known to lie in a simplex and the vertices of the simplex are themselves among the input points.

For the purpose of improving the noise tolerance, we add a second "clean up" stage that iteratively removes each vertex and adds back the point farthest from the span of the remaining vertices. While this additional round of cleanup has been previously suggested as a heuristic to improve quality, in the full version of our paper we show that it also improves the theoretical guarantees of the algorithm.

Finally, the running time of this algorithm can be further improved by using random projection. Randomly projecting a collection of vectors in high dimensions onto a random low-dimensional subspace is well-known to approximately preserve the pairwise distance between each pair of vectors. And since our algorithm iteratively finding the farthest point from a subspace, its behavior is preserved after a random projection. But this refinement of the algorithm allows it to work with low-dimensional points, and improves its efficiency. The final running time is $\widetilde{O}(V^2 + VK/\epsilon^2)$.

3. TOPIC RECOVERY

Here we give an algorithm called `Recover-Topics` (L2) that provably recovers the parameters of the topic model

Algorithm 2. Recover-Topics (L2)

```
1: for $i = 1$ TO $V$ do
2:     Project row $i$ of $\hat{\bar{Q}}$ into the convex hull of the anchor rows, and interpret the resulting convex combination as $p(z_1 = *|w_1 = i)$
3: end for
4: Solve for $A$ using Bayes' rule, as given in Equation (1)
5: Solve the linear system $\hat{Q} = ARA^T$ for $R$
6: Return $A, R$
```

when given the anchor words. The algorithm exploits the same probabilistic and geometric properties of separable topic models, which we described earlier. Recall that every row of $\hat{\bar{Q}}$ can be (approximately) written as a convex combination of the anchor rows. Moreover, the mixing weights are very close to the probabilities $P(w_1 = i|z_1 = k)$.

For each non-anchor row, our algorithm finds the point in the convex hull of the anchor rows that is closest (in Euclidean distance). This is a minimization problem that can be solved effectively using the Exponentiated Gradient algorithm [16]. The resulting point can be expressed as a convex combination of the anchor rows, which yields the conditional probabilities $P(w_1 = i|z_1 = k')$ as descried earlier. These values differ slightly from what we want. Ultimately, we can recover the entries of A through Bayes' rule

$$P(w_1 = i|z_1 = k) = \frac{P(z_1 = k|w_1 = i)P(w_1 = i)}{\sum_{i'} P(z_1 = k|w_1 = i')P(w_1 = i')} \quad (1)$$

Recall that $Q = ARA^T$, and since A has full column rank (because it is separable), we can solve for R by solving this linear system. Furthermore, it turns out that in the special case of the LDA model, we can additionally recover the Dirichlet hyperparameters directly from R. We defer the details to the full version of our paper. Finally, we remark that in our algorithm, when we find the closest point in Euclidean distance this step can be "kernelized", making the running time of each iteration of exponentiated gradient independent of the vocabulary size, V. We can solve the resulting minimization problem with a tolerance of ϵ^2 requires $K \log K/\epsilon^2$ iterations of the Exponentiated Gradient [16] algorithm. The running time of `Recover-Topics` (L2) is $\tilde{O}(V^2K + VK^3/\epsilon^2)$ and the for-loop which constitutes the main computational bottleneck can be trivially parallelized.

Recall that in implementing Bayes' rule, we compute for $k \in [K]$, the denominator $\sum_{i'} p(z_1 = k|w_1 = i')p(w_1 = i') = p(z_k)$ (this is done implicitly when normalizing the columns of A' in Algorithm 2), which gives us, up to a constant scaling, the Dirichlet hyperparameters. This scaling constant can be recovered from the R matrix as described in [4], but in practice we find it better to choose this single parameter using a grid search to maximize the likelihood of the data.

3.1 Theoretical Guarantees

Here we state rigorous guarantees on the sample complexity and running time of our overall algorithm. We defer the guarantees for `FindAnchors` and `Recover-Topics` (L2) themselves to later in this section. When we are given a finite set of samples, our empirical statistics — which we denote by $\hat{\bar{Q}}$ — will be a good, but imperfect approximation to $\bar{Q}$. In order to bound how many samples we need to obtain some target accuracy in recovering the true parameters of the topic model, we need to track the various sources of error through our algorithm.

Moreover, we need that certain parameters are bounded in reasonable ranges to guarantee that the inverse problem we are trying to solve is well-posed. Recall that the existence of anchors implies that we are trying to solve a *separable* non-negative matrix factorization problem. We characterize the separability of the problem as follows:

Definition 1. *The word-topic matrix A is p-separable for $p > 0$ if for each topic k, there is some word i such that $A_{i,k} \geq p$ and $A_{i,k'} = 0$ for $k' \neq k$.*

Thus, not only should each topic have an anchor word but it should also have one that has non-negligible probability. We will require a lower bound on p, and the running time and sample complexity of our algorithm will depend polynomially on $1/p$. We will also need a second measure γ that we will use to denote the smallest singular value of R. When γ is too small, the problem of recovering A and R from $Q = ARA^T$ becomes unstable. Note that this measure also implies that no topic can have very low probability since for any topic i, it can be shown that $\gamma \leq P(z_1 = k)$. When the problem is well-behaved with respect to these two measures, our algorithm achieves the following guarantee:

Theorem 1. *There is a polynomial time algorithm that learns the parameters of a topic model if the number of documents is at least*

$$M = \max\left\{O\left(\frac{\log V \cdot K^6}{\epsilon^2 p^6 \gamma^6 D}\right), O\left(\frac{\log K \cdot K^4}{\gamma^4}\right)\right\},$$

where p and γ are the two non-degeneracy measures defined above and $D \geq 2$ is the length of the shortest document. The algorithm learns the word-topic matrix A and the topic-topic covariance matrix R up to additive error ϵ.

To prove this theorem, we show that the `FindAnchors` algorithm successfully recovers near-anchor words, and the `Recover-Topics` (L2) algorithm accurately estimates the desired parameters given near-anchor words. Before stating the guarantee for `FindAnchors` algorithm, we first introduce the following notion of α-covering. We will say

Let $\{v_1, v_2, ..., v_K\}$ and $\{v'_1, v'_2 ..., v'_K\}$ be two sets of points. We say that these sets of points α-cover each other

Definition 2. *We say that a set of points $\{v'_1, v'_2 ..., v'_K\}$ α-covers another set of points $\{v_1, v_2, ..., v_K\}$, if when representing each v'_i as a convex combination $v = \sum_{k'=1}^{K} c_{k'} v_{k'}$, we have that $c_i \geq 1 - \alpha$.*

Clearly, we would like the anchor points to be α-covered by the set of near-anchors found by `FindAnchors` algorithm. Let δ be the largest perturbation between the rows of $\hat{\bar{Q}}$ and $\bar{Q}$, $\max_i ||\hat{\bar{Q}}_i - \bar{Q}_i|| \leq \delta$. Lemma 2 connects the indices found by `FindAnchors` and the true anchors.

Lemma 2. *If $\delta < (\gamma p)^3/20K$, then* `FindAnchors` *will output a set of rows that $O(\delta/\gamma p)$-covers the true anchor rows.*

Next we show the `Recover-Topics` algorithm is robust to perturbations in the vertices and the internal points, making it possible to bound the error in the reconstruction coefficients in Lemma 3.

Lemma 3. *When* `Recover-Topics` *(L2) is provided with rows which $O(\delta/\gamma p)$-cover the true anchor rows, the element-wise error on the returned matrix A is at most $O(\delta K/\gamma^3 p^2)$.*

Combining these two lemmas, and standard concentration bounds for the empirical correlation matrix $\bar{Q}$, we get the guarantees in the main Theorem 1.

4. EXPERIMENTAL RESULTS

The proposed method in this article, anchor finding followed by convex optimization for topic recovery, is both faster than standard probabilistic approaches and more robust to violations of model assumptions than previous provable approaches. We compare two parameter recovery methods and a standard, probabilistically motivated algorithm. The first method is a simple matrix inversion presented in [4], which we call `Recover`. This inversion method is theoretically optimal, but fails in practice. The second is the constrained recovery method using a squared ℓ_2 loss, which we call `RecoverL2` as shorthand for `Recover-Topics` (L2). As a comparison, we also consider a state-of-the-art Gibbs sampling implementation [20]. We would like an algorithm to be fast, accurate, and robust to noisy data. We find that the anchor-based algorithm is substantially faster than the standard algorithm, especially for large corpora. To evaluate accuracy we test the algorithms on semi-synthetic data (with known topic distributions) and real documents. In addition, we measure the effect of different sources of error and model mismatch.

4.1 Methodology

We train models on two synthetic data sets to evaluate performance when model assumptions are correct, and on real documents to evaluate real-world performance. To ensure that synthetic documents resemble the dimensionality and sparsity characteristics of real data, we generate *semi-synthetic* corpora. For each real corpus, we train a model using Gibbs sampling and then generate new documents using the parameters of that model (these parameters are *not* guaranteed to be separable; we found that about 80% of topics fitted by Gibbs sampling had anchor words).

We use two real-world data sets, a large corpus of **New York Times** articles (295k documents, vocabulary size 15k, mean document length 298) and a small corpus of **NIPS** abstracts (1100 documents, vocabulary size 2500, mean length 68). Vocabularies were pruned with document frequency cutoffs. We generate semi-synthetic corpora of various sizes from models trained with $K = 100$ from NY Times and NIPS, with document lengths set to 300 and 70, respectively, and with document-topic distributions drawn from a Dirichlet with symmetric hyperparameters 0.03.

For the first stage of the algorithm, anchor word recovery, we use the `FindAnchors` algorithm in all cases. The original linear programming-based anchor word finding method presented with `Recover` in [4] is too slow to be comparable. For Gibbs sampling we obtain the word-topic distributions by averaging over 10 saved states, each separated by 100 iterations, after 1000 burn-in iterations.

We use a variety of metrics to evaluate the learned models. For the semi-synthetic corpora, we compute the **reconstruction error** between the true word-topic distributions and the learned distributions. In particular, given a learned matrix $\hat{A}$ and the true matrix A, we use bipartite matching to align topics, and then evaluate the ℓ_1 distance between each pair of topics. When true parameters are not available, a standard evaluation for topic models is to compute **held-out probability**, the probability of previously unseen documents under the learned model.

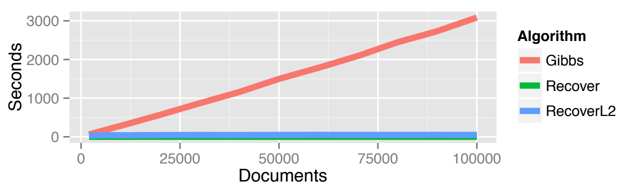


Figure 5: Training time on synthetic NIPS documents.

Topic models are useful because they provide interpretable latent dimensions. We can evaluate the **semantic quality** of individual topics using a metric called *Coherence* [21]. This metric has been shown to correlate well with human judgments of topic quality. If we perfectly reconstruct topics, all the high-probability words in a topic should co-occur frequently, otherwise, the model may be mixing unrelated concepts. Given a set of words $\mathcal{W}$, coherence is

$$Coherence(\mathcal{W}) = \sum_{w_1, w_2 \in \mathcal{W}} \log \frac{D(w_1, w_2) + \epsilon}{D(w_2)}, \qquad (2)$$

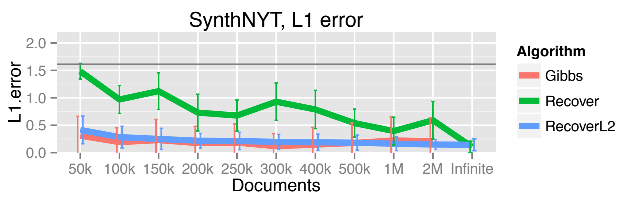
where $D(w)$ and $D(w_1, w_2)$ are the number of documents with at least one instance of w, and of w_1 and w_2, respectively. We set $\epsilon = 0.01$ to avoid taking the log of zero for words that never co-occur. Coherence measures the quality of individual topics, but does not measure redundancy, so we measure **inter-topic similarity**. For each topic, we gather the set of the N most probable words. We then count how many of those words do not appear in any other topic's set of N most probable words. For these experiments we use $N = 20$. Some overlap is expected due to semantic ambiguity, but lower numbers of unique words indicate less useful models.

4.2 Efficiency

Both the `Recover` and `RecoverL2` algorithms, in Python, are faster than a heavily optimized Gibbs sampling implementation in Java. Fig. 5 shows the time to train models on synthetic corpora on a single machine. Gibbs sampling is linear in the corpus size. `RecoverL2` is also linear ($\rho = 0.79$), but only varies from 33 to 50 seconds. Estimating Q is linear, but takes only 7 seconds for the largest corpus. `FindAnchors` takes less than 6 seconds for all corpora.

4.3 Semi-synthetic documents

The new algorithms have good ℓ_1 reconstruction error on semi-synthetic documents, especially for larger corpora. Results for semi-synthetic corpora drawn from topics trained on NY Times articles are shown in Fig. 6 (top) for corpus sizes ranging from 50k to 2M synthetic documents. In addition, we report results for the `Recover` and `RecoverL2` algorithms on "infinite data," that is, the true Q matrix from the model used to generate the documents. Error bars show variation between topics. `Recover` performs poorly in all but the noiseless, infinite data setting. Gibbs sampling has the lowest ℓ_1 on smaller corpora. However, for the larger corpora the new `RecoverL2` algorithm have the lowest ℓ_1 error



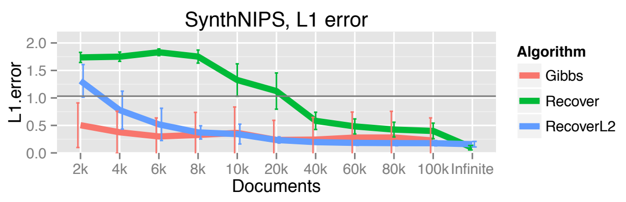
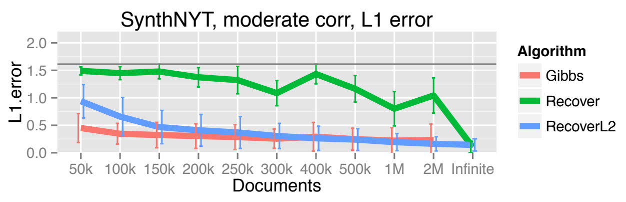


Figure 6: ℓ_1 error for learning semi-synthetic LDA models with $K = 100$ topics (top: based on NY Times, bottom: based on NIPS abstracts). The horizontal lines indicate the ℓ_1 error of K uniform distributions.



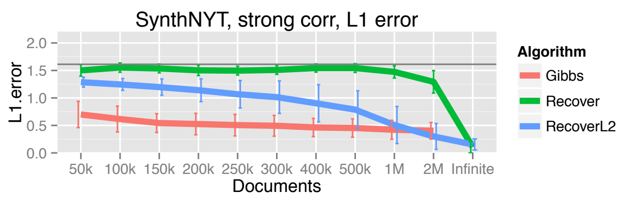


Figure 8: ℓ_1 error increases as we increase topic correlation (top: $\rho = 0.05$, bottom: $\rho = 0.1$). Based on the NY Times semi-synthetic model with 100 topics.

and smaller variance (running sampling longer may reduce MCMC error further). Results for semi-synthetic corpora drawn from NIPS topics are shown in Fig. 6 (bottom), and are similar.

Effect of separability. Notice that in Fig. 6, `Recover` does not achieve zero ℓ_1 error even with noiseless "infinite" data. Here we show that this is due to lack of separability, and that the new recovery algorithms are more robust to violations of the separability assumption. In our semi-synthetic corpora, documents are generated from an LDA model, but the topic-word distributions are learned from data and may not satisfy the anchor words assumption. We now add a synthetic anchor word to each topic that is, by construction, unique to that topic. We assign the synthetic anchor word a probability equal to the most probable word in the original topic. This causes the distribution to sum to greater than 1.0, so we renormalize. Results are shown in Fig. 7. The ℓ_1 error goes to zero for `Recover`, and close to zero for `RecoverL2` (not zero because we do not solve to perfect optimality).

Effect of correlation. The theoretical guarantees of the new algorithms apply even if topics are correlated. To test the empirical performance in the presence of correlation, we generated new synthetic corpora from the same $K = 100$

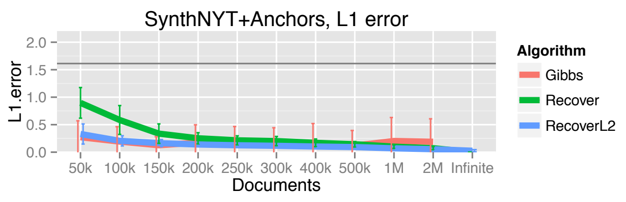


Figure 7: When we add artificial anchor words before generating synthetic documents, ℓ_1 error goes to zero for `Recover` and close to zero for `RecoverL2`.

model trained on NY Times articles. Instead of a symmetric Dirichlet distribution, we use a logistic Normal distribution with a block-structured covariance matrix. We partition topics into 10 groups. For each pair of topics in a group, we add a non-zero off-diagonal element (ρ) to the covariance matrix. This block structure is not necessarily realistic, but shows the effect of correlation. Results for $\rho = 0.05$ and 0.1 are shown in Fig. 8. `Recover` performs much worse with correlated topics than with LDA-generated corpora (c.f. Fig. 6). The other three algorithms, especially Gibbs sampling, are more robust to correlation. Performance consistently degrades as correlation increases. For the recovery algorithms this is due to a decrease in γ, the condition number of the R matrix. With infinite data, ℓ_1 error is equal to the ℓ_1 error in the uncorrelated synthetic corpus (non-zero because of violations of the separability assumption).

4.4 Real documents

The new algorithms produce comparable quantitative and qualitative results on real data. Fig. 9 shows three metrics for both corpora. Error bars show the distribution of log probabilities across held-out *documents* (top panel) and coherence and unique words across *topics* (center and bottom panels). Held-out sets are 230 documents for NIPS and 59k for NY Times. For the small NIPS corpus we average over 5 non-overlapping train/test splits. The matrix inversion step in `Recover` fails for the NIPS corpus so we modify the procedure to use pseudoinverse. This modification is described in the supplementary materials. In both corpora, `Recover` produces noticeably worse held-out log probability per token than the other algorithms. Gibbs sampling produces the best average held-out probability ($p < 0.0001$ under a paired t-test), but the difference is within the range of variability between documents. We tried several methods for estimating hyperparameters, but the observed differences did not change the relative performance of algorithms. Gibbs sampling has worse coherence than the other algorithms, but produces more unique words per topic. These patterns are consistent with semi-synthetic results for similarly sized corpora (details are in supplementary material).

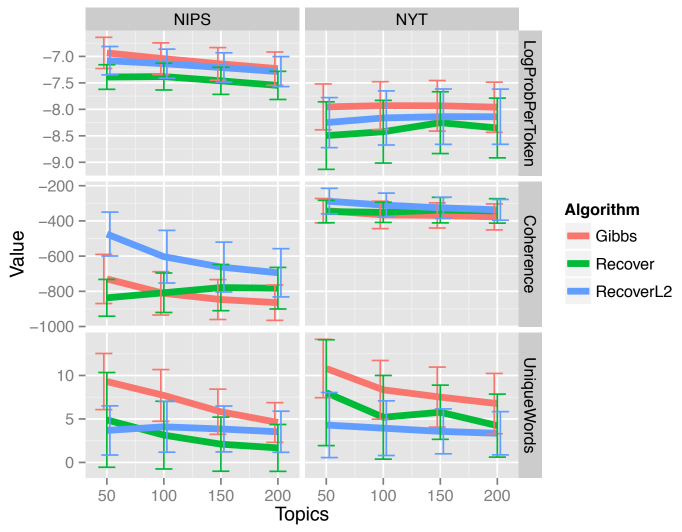


Figure 9: Held-out probability (per token) is similar for `RecoverL2` and Gibbs sampling. `RecoverL2` has better coherence, but fewer unique terms in the top $N = 20$ words than Gibbs. (Up is better for all three metrics.)

For each NY Times topic learned by `RecoverL2` we find the closest Gibbs topic by ℓ_1 distance. The closest, median, and farthest topic pairs are shown in Table 1. We observe that when there is a difference, recover-based topics tend to have more specific words (*Anaheim Angels* vs. *pitch*).

Table 1: Example topic pairs from NY Times (closest ℓ_1), anchor words in bold. The UCI NY Times corpus includes named-entity annotations, indicated by the *zzz* prefix. All 100 topics are shown in the supplementary material.

`RecoverL2`	run inning game hit season zzz_anaheim_angel
Gibbs	run inning hit game ball pitch
`RecoverL2`	father family **zzz_elian** boy court zzz_miami
Gibbs	zzz_cuba zzz_miami cuban zzz_elian boy protest
`RecoverL2`	**file** sport read internet email zzz_los_angeles
Gibbs	web site com www mail zzz_internet

5. CONCLUDING REMARKS

Here we have shown that algorithms based on the separability assumption are highly practical and produce topic models of quality comparable to likelihood-based methods that use Gibbs sampling, while running in a fraction of the time. Moreover, these algorithms are particularly well-suited to parallel implementations, since each of the major steps — with the exception of finding the anchor words — can be trivially parallelized. Our algorithms inherit provable guarantees from earlier approaches [4] in the sense that given samples from a topic model, the estimates provably converge to the true parameters at an inverse polynomial rate. However an important question going forward is to theoretically explain why these algorithms appear to be (somewhat) robust to model misspecification. In our experiments, we fit a topic model to real data and the resulting topic model is *not* separable, but merely close to being separable. Nevertheless, our algorithms recover high quality topics in this setting too. Likelihood based methods are known to be well-behaved when the model is misspecified, and ideally one should be able to design provable algorithms that not only have good running time and sample complexity, but can also tolerate a realistic amount of noise.

Since its publication, our algorithms have been extended in a number of directions. Roberts et al. [24] consider applications in the social sciences and find that using an anchor-based model as an initialization for a likelihood-based algorithm reduces variability and improves model fit. Nguyen et al. [22] improve the topic recovery step by adding regularization to smooth the estimated topic-word distributions, resulting in improved interpretability. A number of authors have suggested new approaches to find anchor words. Ding et al. [10] present a distributed algorithm that can be parallelized across multiple servers. Zhou et al. [27] find anchor words by projecting rows of $\bar{Q}$ into the plane, and selecting words that often appear as extreme points. Lee and Mimno [18] replace random projections with a single heavy-tailed t-SNE projection that does not preserve pairwise ℓ_2 distances, but preserves local distances, allowing points to be more spread out in the projected space. Ge and Zou [12] relaxed the anchor word assumption to a subset separable assumption that can hold even when anchor words are not in a single topic, but a combination of a few topics. Other recent work [17] established criteria necessary for the anchor factorization. Enforcing these criteria on the input matrix through an initial rectification step substantially improved model robustness, especially for small numbers of topics.

More broadly, the anchor words themselves have also proven to be a useful tool in summarizing the meaning of a topic and distinguishing a topic from related topics. When coupled with the right visualization and analytic tools, it may be possible to design semi-supervised learning algorithms where a domain expert helps choose the final set of anchors. It is also possible that anchor words will find applications beyond text analysis, and will enable efficient algorithms in other domains much the same way this assumption has in topic modeling.

6. REFERENCES

[1] A. Ahmed, M. Aly, J. Gonzalez, S. Narayanamurthy, and A. J. Smola. Scalable inference in latent variable models. In *WSDM '12: Proceedings of the fifth ACM international conference on Web search and data mining*, pages 123–132, New York, NY, USA, 2012. ACM.

[2] A. Anandkumar, D. Foster, D. Hsu, S. Kakade, and Y. Liu. Two svds suffice: Spectral decompositions for probabilistic topic modeling and latent dirichlet allocation. In *NIPS*, 2012.

[3] S. Arora, R. Ge, R. Kannan, and A. Moitra. Computing a nonnegative matrix factorization – provably. In *STOC*, pages 145–162, 2012.

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

[5] D. Blei. Introduction to probabilistic topic models. *Communications of the ACM*, pages 77–84, 2012.

[6] D. Blei and J. Lafferty. A correlated topic model of science. *Annals of Applied Statistics*, pages 17–35, 2007.

[7] D. Blei, A. Ng, and M. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, pages

993–1022, 2003. Preliminary version in *NIPS* 2001.

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

[9] 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*, pages 1–38, 1977.

[10] W. Ding, M. H. Rohban, P. Ishwar, and V. Saligrama. Efficient distributed topic modeling with provable guarantees. *JMLR*, pages 167–175, 2014.

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

[12] R. Ge and J. Zou. Intersecting faces: Non-negative matrix factorization with new guarantees. In *Proceedings of The 32nd International Conference on Machine Learning*, pages 2295–2303, 2015.

[13] T. L. Griffiths and M. Steyvers. Finding scientific topics. *Proceedings of the National Academy of Sciences*, 101:5228–5235, 2004.

[14] M. D. Hoffman and D. M. Blei. Structured stochastic variational inference. In *18th International Conference on Artificial Intelligence and Statistics*, 2015.

[15] A. T. Kalai, A. Moitra, and G. Valiant. Disentangling gaussians. *Commun. ACM*, 55(2):113–120, Feb. 2012.

[16] J. Kivinen and M. K. Warmuth. Exponentiated gradient versus gradient descent for linear predictors. *Inform. and Comput.*, 132, 1995.

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

[18] M. Lee and D. Mimno. Low-dimensional embeddings for interpretable anchor-based topic inference. In *EMNLP*, 2014.

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

[20] A. McCallum. Mallet: A machine learning for language toolkit, 2002. http://mallet.cs.umass.edu.

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

[22] T. Nguyen, Y. Hu, and J. Boyd-Graber. Anchors regularized: Adding robustness and extensibility to scalable topic-modeling algorithms. In *ACL*, 2014.

[23] K. Pearson. Contributions to the mathematical theory of evolution. *Philosophical Transactions of the Royal Society of London. A*, 185:71–110, 1894.

[24] M. E. Roberts, B. M. Stewart, and D. Tingley. Navigating the local modes of big data: The case of topic models. In *Data Science for Politics, Policy and Government*. Cambridge University Press, 2014.

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

[26] S. Vavasis. On the complexity of nonnegative matrix factorization. *SIAM Journal on Optimization*, pages 1364–1377, 2009.

[27] T. Zhou, J. A. Bilmes, and C. Guestrin. Divide-and-conquer learning by anchoring a conical hull. In *NIPS*, pages 1242–1250, 2014.