A Practical Algorithm for Topic Modeling with Provable Guarantees

Sanjeev Arora, Rong Ge, Yoni Halpern, David Mimno, Ankur Moitra, David Sontag, Yichen Wu, Michael Zhu

Reviewed by Zhao Song

December 13, 2013
1 Background

2 The Proposed Algorithm
   Word-Word Co-occurrences Matrix Construction
   Topic Recovery via Bayes’ Rule
   Anchor Words Search

3 Experimental Results
Outline

1 Background

2 The Proposed Algorithm
   Word-Word Co-occurences Matrix Construction
   Topic Recovery via Bayes’ Rule
   Anchor Words Search

3 Experimental Results
Topic Modeling

Nonnegative Matrix Factorization problem [Arora, Ge, and Moitra 2012]:

- **A**: Unknown word-topic matrix with dimension $V \times K$.
  \[ A_{i,k} = p(w_1 = i | z_1 = k) = p(w_2 = i | z_2 = k) \]

- **W**: Unknown topic-document matrix with dimension $K \times M$.
  \[ W_{i,k} = p(z_1 = i | d_1 = k) = p(z_2 = i | d_2 = k) \]

- **R**: Unknown topic-topic covariance matrix with dimension $K \times K$
  \[ R = \mathbb{E}(WW^T) \]

- **Q**: Word-word co-occurrence matrix with dimension $V \times V$.
  \[ Q = \mathbb{E}[AWW^TA^T] = ARA^T \]

**Task**: Given $Q$, recover $A$ and $R$. 
[Arora, Ge, and Moitra 2012] presented a provably polynomial-time algorithm learning parameters of topic model provided that every topic contains at least one anchor word. The word-topic matrix $A$ is $p$–separable: For $p > 0$ and for each topic $k$, there is some word $i$ such that

- $A_{i,k} \geq p$;
- $A_{i,k'} = 0$, for $k' \neq k$.

The word here is called *Anchor Word*. Two steps:

- Selection step: Find anchor words.
- Recovery step: Reconstruct topic distribution.
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 \( DRA^T \) (which is equal to \( DR\bar{1} \))
3) Solve for \( \bar{z} \): \( DRD\bar{z} = DR\bar{1} \).
4) Output \( A^T = ((DRDD\text{Diag}(z))^{-1}DRA^T) \).
5) Output \( R = (\text{Diag}(z)DRDD\text{Diag}(z)) \).
Limitations in [Arora, Ge, and Moitra 2012]

- **Unstable and sensitive to noise:** Use matrix inversion to recover $A$.
- **Not sufficient for real data:** Only use $K$ rows of the word-word co-occurrence matrix $Q$, corresponding to the anchor words.
Outline

1 Background

2 The Proposed Algorithm
   Word-Word Co-occurences Matrix Construction
   Topic Recovery via Bayes’ Rule
   Anchor Words Search

3 Experimental Results
Algorithm 1. High Level Algorithm

Input: Textual corpus $\mathcal{D}$, Number of topics $K$,
Tolerance parameters $\epsilon_a, \epsilon_b > 0$.
Output: Word-topic matrix $A$, topic-topic matrix $R$

$Q \leftarrow \text{Word Co-occurrences}(\mathcal{D})$
Form $\{\tilde{Q}_1, \tilde{Q}_2, \ldots \tilde{Q}_V\}$, the normalized rows of $Q$.
$S \leftarrow \text{FastAnchorWords}(\{\tilde{Q}_1, \tilde{Q}_2, \ldots \tilde{Q}_V\}, K, \epsilon_a)$
(Algorithm 4)
$A, R \leftarrow \text{RecoverKL}(Q, S, \epsilon_b)$ (Algorithm 3)
return $A, R$
Outline

1 Background

2 The Proposed Algorithm
   Word-Word Co-occurences Matrix Construction
   Topic Recovery via Bayes’ Rule
   Anchor Words Search

3 Experimental Results
Construct $Q$ Matrix

Let $H_d \in \mathbb{R}^V$ where the $i$ – th element equals the number of times word $i$ appearing in document $d$ and $n_d$ be the length of document.

- $\tilde{H}_d = \frac{H_d}{\sqrt{n_d(n_d - 1)}}$
- $\hat{H}_d = \frac{\text{Diag}(H_d)}{n_d(n_d - 1)}$

Then, collect all $\tilde{H}_d$ to form $\tilde{H}$ and compute the sum of all $\hat{H}_d$ to get $\hat{H}$. We can prove

$$\mathbb{E}(\tilde{H} \tilde{H}^T - \hat{H}) = Q$$

and use $\tilde{H} \tilde{H}^T - \hat{H}$ to approximate $Q$ and normalize its row to get $\bar{Q}$. 

Outline

1 Background

2 The Proposed Algorithm
   Word-Word Co-occurences Matrix Construction
   Topic Recovery via Bayes’ Rule
   Anchor Words Search

3 Experimental Results
Interpretations for matrix $\bar{Q}$

- For an anchor word $s_k$

$$\bar{Q}_{s_k,j} = \sum_{k'} p(z_1 = k' \mid w_1 = s_k) p(w_2 = j \mid z_1 = k')$$  \hspace{1cm} (1)$$

$$= p(w_2 = j \mid z_1 = k)$$ \hspace{1cm} (2)$$

- For any other word $i$

$$\bar{Q}_{i,j} = \sum_k p(z_1 = k \mid w_1 = i) p(w_2 = j \mid z_1 = k)$$ \hspace{1cm} (3)$$

$$= \sum_k C_{i,k} \bar{Q}_{s_k,j}$$ \hspace{1cm} (4)$$

Every other rows of $\bar{Q}$ lies in the Convex Hull of rows corresponding to the anchor words.
Matrices $A$ and $R$ recovery

- Bayes’ rule:

$$p(w_1 = i | z_1 = k) = \frac{p(z_1 = k | w_1 = i) p(w_1 = i)}{\sum_{i'} p(z_1 = k | w_1 = i') p(w_1 = i')}$$  \hspace{1cm} (5)$$

where

$$p(w_1 = i) = \sum_j p(w_1 = i, w_2 = j) = \sum_j Q_{i,j}$$  \hspace{1cm} (6)$$

- How to compute $C_i. = p(z_1 | w_1 = i)$: Defining different loss function

  1. KL Divergence: $\mathbb{D}(\bar{Q}_i \parallel \sum_{k \in S} C_{i,k} \bar{Q}_{s_k})$

  2. Quadratic Loss $\| \bar{Q}_i - \sum_{k \in S} C_{i,k} \bar{Q}_{s_k} \|^2$

- Matrix $R$ recovery

$$R = A^\dagger Q(A^\dagger)^T$$  \hspace{1cm} (7)$$
RecoverKL Algorithm

Algorithm 3. RecoverKL

Input: Matrix $Q$, Set of anchor words $S$, tolerance parameter $\epsilon$.

Output: Matrices $A, R$

Normalize the rows of $Q$ to form $\bar{Q}$.
Store the normalization constants $\bar{p}_w = Q^{1T}$.
$\bar{Q}_{sk}$ is the row of $\bar{Q}$ for the $k^{th}$ anchor word.

for $i = 1, ..., V$ do

Solve $C_i = \arg\min_{\bar{C}_i} D_{KL}(\bar{Q}_i \parallel \sum_{k \in S} C_{i,k} \bar{Q}_{sk})$
Subject to: $\sum_k C_{i,k} = 1$ and $C_{i,k} \geq 0$
With tolerance: $\epsilon$

end for

$A' = \text{diag}(\bar{p}_w) C$

Normalize the columns of $A'$ to form $A$.

$R = A^T Q A^T$

return $A, R$
Outline

1. Background

2. The Proposed Algorithm
   - Word-Word Co-occurences Matrix Construction
   - Topic Recovery via Bayes’ Rule
   - Anchor Words Search

3. Experimental Results
A Geometric Interpretation

• For *infinite* documents, the convex hull $P$ of the rows in $\bar{Q}$ $(a_1, a_2, \ldots, a_V)$ will be a simplex whose vertices correspond to the anchor words.

• For *finite* documents, the rows in $\bar{Q}$ $(d_1, d_2, \ldots, d_V)$ are only approximated to their expectation.

Find an approximation to the vertices of $P$

• Iteratively find the farthest point from the subspace spanned by the anchor words found so far.
Algorithm 4. FastAnchorWords

Input: V points \( \{d_1, d_2, ..., d_V\} \) in V dimensions, almost in a simplex with K vertices and \( \epsilon > 0 \)
Output: K points that are close to the vertices of the simplex.

Project the points \( d_i \) to a randomly chosen 4 log \( V/\epsilon^2 \) dimensional subspace
\( S \leftarrow \{d_i\} \) s.t. \( d_i \) is the farthest point from the origin.
for \( i = 1 \) TO \( K - 1 \) do
    Let \( d_j \) be the point in \( \{d_1, ..., d_V\} \) that has the largest distance to \( \text{span}(S) \).
    \( S \leftarrow S \cup \{d_j\} \).
end for
\( S = \{v'_1, v'_2, ..., v'_K\} \).
for \( i = 1 \) TO \( K \) do
    Let \( d_j \) be the point that has the largest distance to \( \text{span}\{v'_1, v'_2, ..., v'_K\}\backslash\{v'_i\} \)
    Update \( v'_i \) to \( d_j \)
end for
Return \( \{v'_1, v'_2, ..., v'_K\} \).
Outline

1 Background

2 The Proposed Algorithm
   Word-Word Co-occurrences Matrix Construction
   Topic Recovery via Bayes’ Rule
   Anchor Words Search

3 Experimental Results
Experiments Setup

Compare the following four methods:

- Gibbs sampling [McCallum 2002]
- Recover [Arora, Ge, and Moitra 2012]
- RecoverKL
- RecoverL2

Two real-world data sets:

- New York Times articles
  295k documents, vocabulary size 15k, mean document length 298.
- NIPS abstracts
  1100 documents, vocabulary size 2500, mean document length 68.
Experiments Setup (Cont.)

Semi-synthetic corpora

1. Train a model using MCMC for each real corpora
2. Generate new documents using parameter trained.

Performance metrics

- Reconstruction error: $\ell_1$ distance between a learned matrix $\hat{A}$ and the true matrix $A$.
- Held-out probability: probability of previously unseen documents under the learned model.
- Coherence:

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

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.
Figure 1. Training time on synthetic NIPS documents.
Figure 2. $\ell_1$ 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 $\ell_1$ error of $K$ uniform distributions.
Figure 3. When we add artificial anchor words before generating synthetic documents, $\ell_1$ error goes to zero for Recover and close to zero for RecoverKL and RecoverL2.
Correlation

Topics are correlated

Figure 4. $\ell_1$ 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.
Figure 5. Held-out probability (per token) is similar for RecoverKL, RecoverL2, and Gibbs sampling. RecoverKL and RecoverL2 have better coherence, but fewer unique words than Gibbs. (Up is better for all three metrics.)
