

# Learning Topic Models

## – Going Beyond SVD

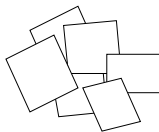
Ankur Moitra, IAS

joint with Sanjeev Arora and Rong Ge

October 21, 2012

# Topic Models

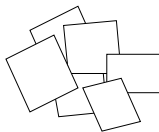
Large collection of articles, say from the New York Times:



newspaper articles

# Topic Models

Large collection of articles, say from the New York Times:



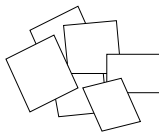
newspaper articles

## Question

*How can we automatically organize them by topic? (unsupervised learning)*

# Topic Models

Large collection of articles, say from the New York Times:

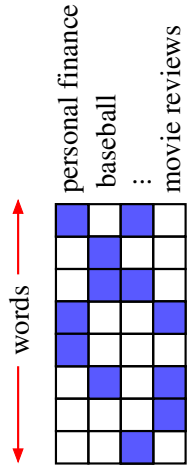


newspaper articles

## Question

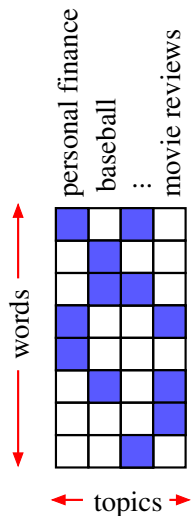
*How can we automatically organize them by topic? (unsupervised learning)*

**Challenge:** Develop tools for automatic comprehension of data - e.g. newspaper articles, webpages, images, genetic sequences, user ratings...



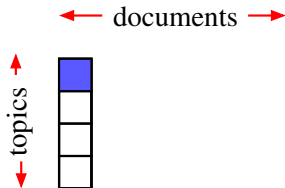
← topics →

A

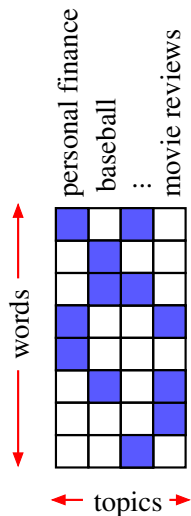


A

document #1: (1.0, personal finance)

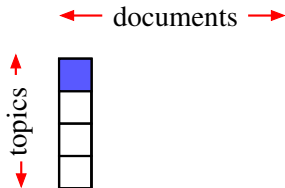


W



A

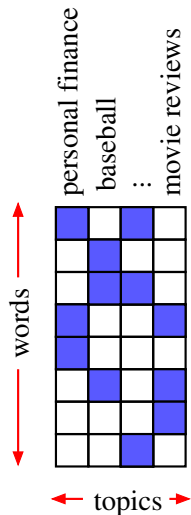
document #1: (1.0, personal finance)



=



W

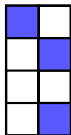


A

document #2: (0.5, baseball); (0.5, movie reviews)

← documents →

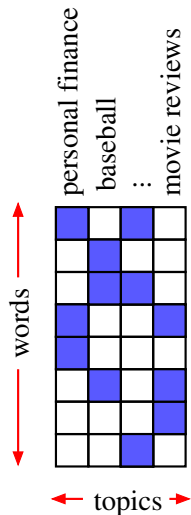
↑ topics ↓



W

=



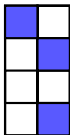


A

document #2: (0.5, baseball); (0.5, movie reviews)

← documents →

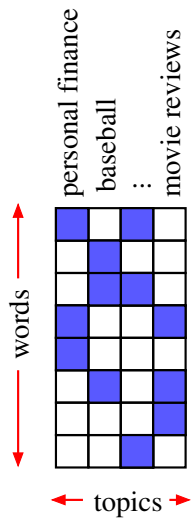
↑ topics ↓



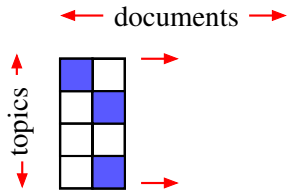
W

=



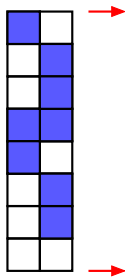


A

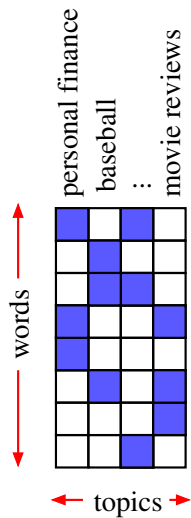


W

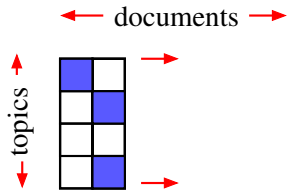
=



M

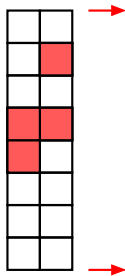


$A$



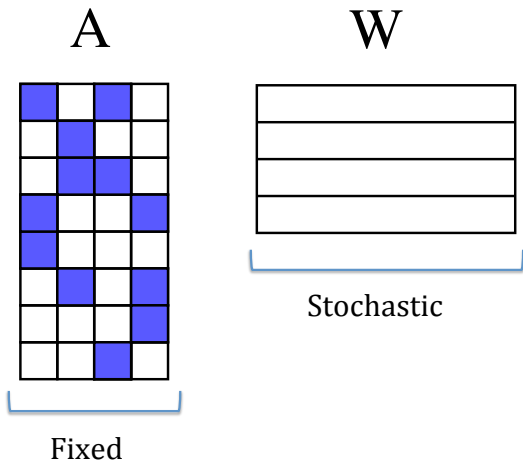
$W$

$\approx$



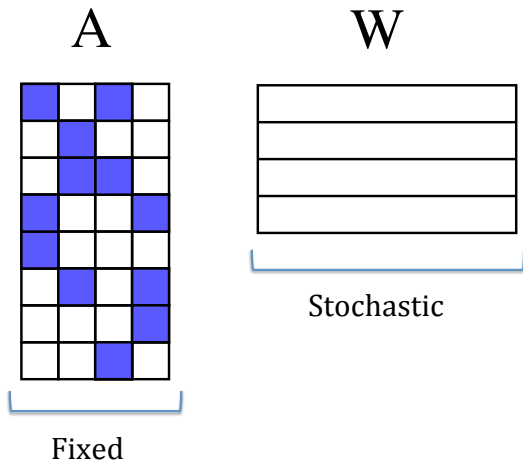
# So Many Models!

**Pure Topics:** one topic per document



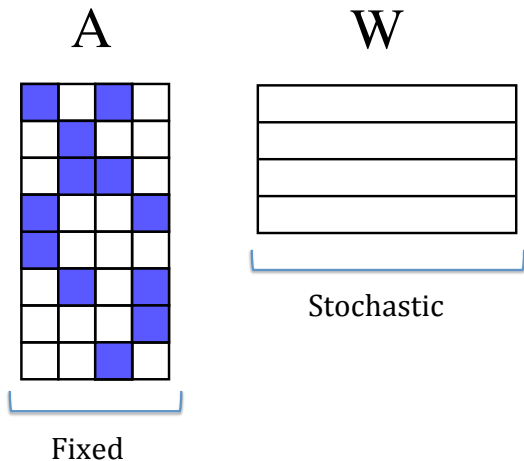
# So Many Models!

**LDA:** [Blei et al] Dirichlet distribution



# So Many Models!

## CTM / Pachinko: structured correlations



# Algorithms

- **Maximum Likelihood:** Find the parameters that maximize the likelihood of generating the observed data.

# Algorithms

- **Maximum Likelihood:** Find the parameters that maximize the likelihood of generating the observed data.  
Hard to compute!

# Algorithms

- **Maximum Likelihood:** Find the parameters that maximize the likelihood of generating the observed data.  
Hard to compute!
- **Spectral:** Compute the singular value decomposition of  $\hat{M}$ .  
[Papadimitriou et al], [Azar et al], ...

# Algorithms

- **Maximum Likelihood:** Find the parameters that maximize the likelihood of generating the observed data.

Hard to compute!

- **Spectral:** Compute the singular value decomposition of  $\hat{M}$ .  
[Papadimitriou et al], [Azar et al], ...

But the singular vectors are orthonormal!

# Algorithms

- **Maximum Likelihood:** Find the parameters that maximize the likelihood of generating the observed data.  
Hard to compute!
- **Spectral:** Compute the singular value decomposition of  $\hat{M}$ .  
[Papadimitriou et al], [Azar et al], ...  
But the singular vectors are orthonormal!

## Question

*Can we use tools from nonnegative matrix factorization instead of spectral methods?*

[AGKM]: fixed parameter intractable but there are easy cases

# Our Results

Let  $E[WW^T] = R$  be the topic-topic covariance matrix, let  $\kappa$  be its condition number and let  $a = \max_{i,j} \frac{E[W_i]}{E[W_j]}$  be the topic imbalance.

# Our Results

Let  $E[WW^T] = R$  be the topic-topic covariance matrix, let  $\kappa$  be its condition number and let  $a = \max_{i,j} \frac{E[W_i]}{E[W_j]}$  be the topic imbalance.

If the topic matrix  $A$  satisfies the “anchor word assumption” for  $p > 0$ :

# Our Results

Let  $E[WW^T] = R$  be the topic-topic covariance matrix, let  $\kappa$  be its condition number and let  $a = \max_{i,j} \frac{E[W_i]}{E[W_j]}$  be the topic imbalance.

If the topic matrix  $A$  satisfies the “anchor word assumption” for  $p > 0$ :

## Theorem

*We can learn the topic matrix  $A$  and covariance matrix  $R$  to within accuracy  $\epsilon$  in time and number of docs  $\text{poly}(\log n, r, 1/\epsilon, 1/p, \kappa, a)$  with  $n$  words and  $r$  topics*

# Our Results

Let  $E[WW^T] = R$  be the topic-topic covariance matrix, let  $\kappa$  be its condition number and let  $a = \max_{i,j} \frac{E[W_i]}{E[W_j]}$  be the topic imbalance.

If the topic matrix  $A$  satisfies the “anchor word assumption” for  $p > 0$ :

## Theorem

*We can learn the topic matrix  $A$  and covariance matrix  $R$  to within accuracy  $\epsilon$  in time and number of docs  $\text{poly}(\log n, r, 1/\epsilon, 1/p, \kappa, a)$  with  $n$  words and  $r$  topics*

Suffices to have documents of size two!

If an anchor word (for a topic) occurs, the document is at least partially about the given topic:

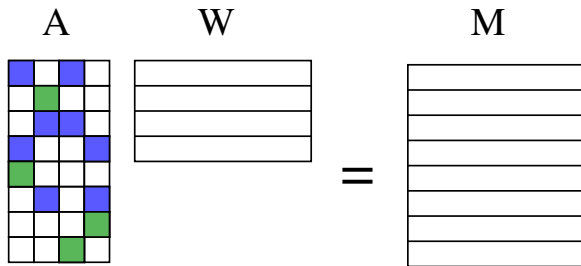
	personal finance	baseball	...	movie reviews	
	blue	white	blue	white	
	white	green	white	white	bunt
	white	blue	blue	white	
	blue	white	white	blue	
	green	white	white	white	401k
	white	blue	white	blue	
	white	white	white	green	oscar-winning
	white	white	green	white	

If an anchor word (for a topic) occurs, the document is at least partially about the given topic:

	personal finance	baseball	...	movie reviews	
	blue	white	blue	white	
	white	green	white	white	bunt
	white	blue	blue	white	
	blue	white	white	blue	
	green	white	white	white	401k
	white	blue	white	blue	
	white	white	white	green	oscar-winning
	white	white	green	white	

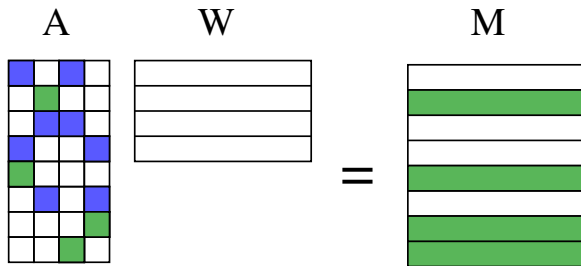
Each topic has an anchor word that occurs with probability  $\geq p$

# Anchor Words as Extreme Points [AGKM]



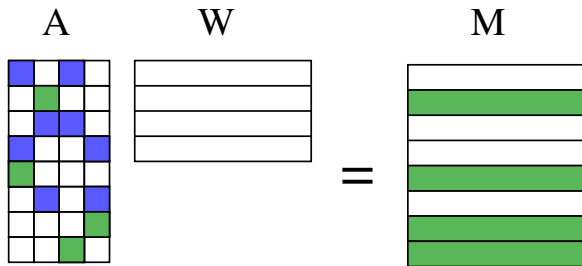
Can we efficiently determine  
if a word is an anchor word?

# Anchor Words as Extreme Points [AGKM]

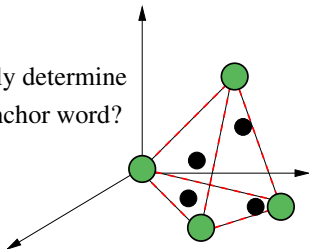


Can we efficiently determine  
if a word is an anchor word?

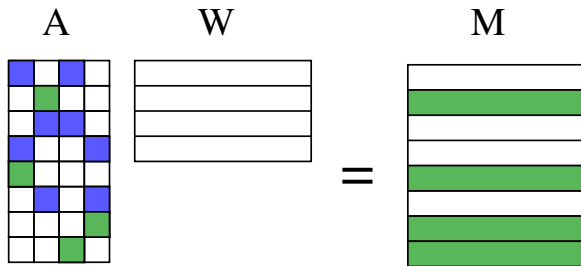
# Anchor Words as Extreme Points [AGKM]



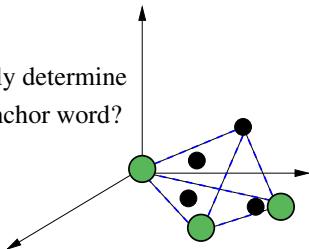
Can we efficiently determine  
if a word is an anchor word?



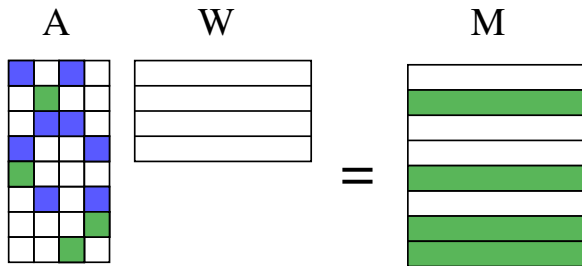
# Anchor Words as Extreme Points [AGKM]



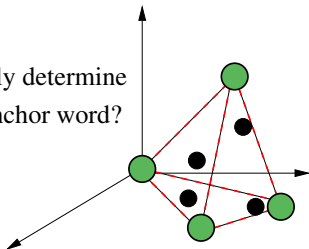
Can we efficiently determine  
if a word is an anchor word?



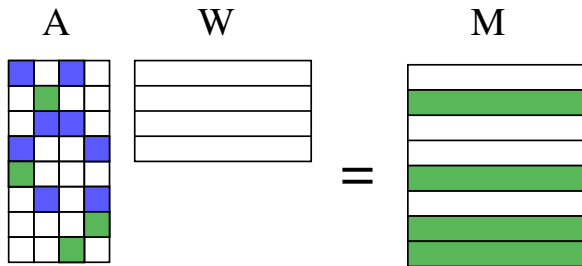
# Anchor Words as Extreme Points [AGKM]



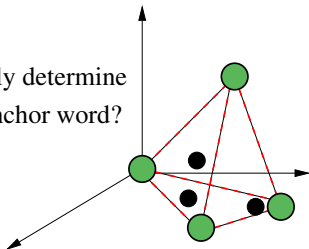
Can we efficiently determine  
if a word is an anchor word?



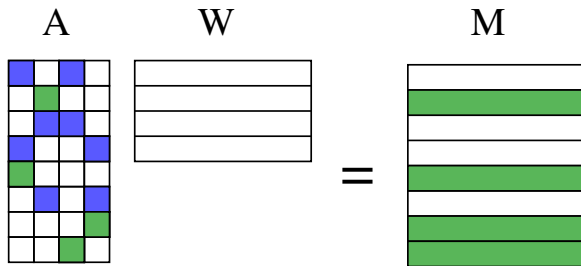
# Anchor Words as Extreme Points [AGKM]



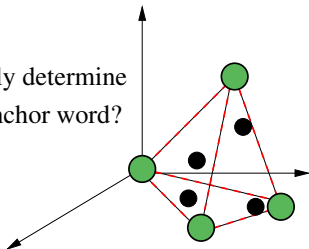
Can we efficiently determine  
if a word is an anchor word?



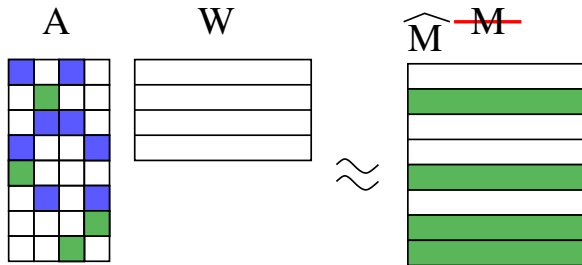
# Anchor Words as Extreme Points [AGKM]



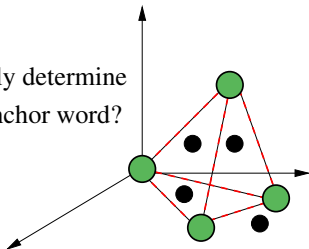
Can we efficiently determine  
if a word is an anchor word?



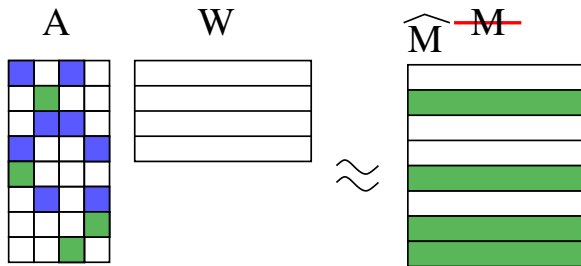
# Problem: Sampling “Noise”



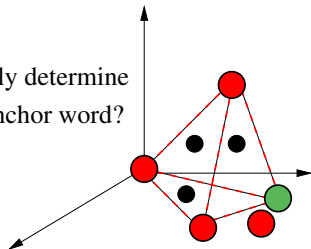
Can we efficiently determine  
if a word is an anchor word?



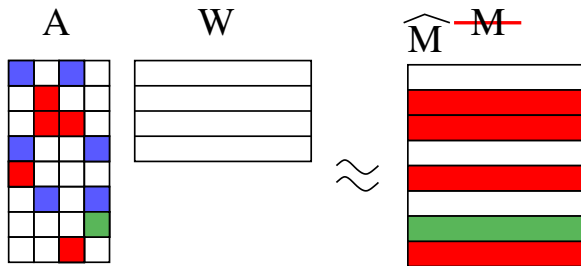
# Problem: Sampling “Noise”



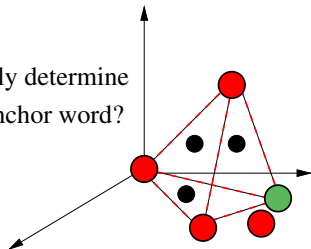
Can we efficiently determine  
if a word is an anchor word?



# Problem: Sampling “Noise”



Can we efficiently determine  
if a word is an anchor word?



# Our Algorithm

$\hat{M}$  is far from  $M$ , but let's use  $\hat{M}\hat{M}^T$  instead!

# Our Algorithm

$\hat{M}$  is far from  $M$ , but let's use  $\hat{M}\hat{M}^T$  instead!

$\hat{M}\hat{M}^T \rightarrow MM^T$  and  $WW^T \rightarrow R$  as number of documents increase

# Our Algorithm

$\hat{M}$  is far from  $M$ , but let's use  $\hat{M}\hat{M}^T$  instead!

$\hat{M}\hat{M}^T \rightarrow MM^T$  and  $WW^T \rightarrow R$  as number of documents increase

## Step

*We can recover the anchor words from  $MM^T$ .*

# Finding the Anchor Words

$$\begin{array}{c}
 \widehat{M M^T} \xrightarrow{A} M M^T \quad W W^T \xrightarrow{A^T} R \\
 \begin{array}{|c|c|c|c|} \hline \text{blue} & & \text{blue} & \\ \hline & \text{green} & & \\ \hline & \text{blue} & \text{blue} & \\ \hline \text{blue} & & & \text{blue} \\ \hline \text{green} & & & \\ \hline & \text{blue} & & \text{blue} \\ \hline & & & \text{green} \\ \hline & & \text{green} & \\ \hline \end{array}
 \end{array}
 \quad
 \begin{array}{c}
 W W^T \\
 \begin{array}{|c|c|c|c|c|c|} \hline \text{blue} & & \text{blue} & \text{green} & & \\ \hline & \text{green} & \text{blue} & & \text{blue} & \\ \hline \text{blue} & & \text{blue} & & & \text{green} \\ \hline & & & \text{blue} & \text{blue} & \text{green} \\ \hline \end{array}
 \end{array}
 = M M^T$$

## Finding the Anchor Words

$$\begin{array}{ccc} \widehat{\mathbf{M}} \widehat{\mathbf{M}}^T & \rightarrow & \mathbf{M} \mathbf{M}^T \\ \mathbf{A} & & \mathbf{A}^T \end{array} \quad \begin{array}{ccc} \mathbf{W} \mathbf{W}^T & \rightarrow & \mathbf{R} \\ & & \mathbf{A}^T \end{array}$$

Diagram illustrating the nonnegativity of the product of two nonnegative matrices. It shows a 5x5 matrix  $W$  with blue and green squares, followed by the expression  $W W^T$ , then another 5x5 matrix with blue and green squares, followed by an equals sign and the expression  $M M^T$ . A blue bracket underneath the  $W W^T$  and the second matrix points to the text "Nonnegative!".

## Finding the Anchor Words

$$\widehat{MM^T} \xrightarrow{A} MM^T \quad W W^T \xrightarrow{A^T} R$$

Nonnegative!

Anchor words from:  $\mathbf{M}\mathbf{M}^T$

# Our Algorithm

$\hat{M}$  is far from  $M$ , but let's use  $\hat{M}\hat{M}^T$  instead!

$\hat{M}\hat{M}^T \rightarrow MM^T$  and  $WW^T \rightarrow R$  as number of documents increase

## Step

*We can recover the anchor words from  $MM^T$ .*

# Our Algorithm

$\hat{M}$  is far from  $M$ , but let's use  $\hat{M}\hat{M}^T$  instead!

$\hat{M}\hat{M}^T \rightarrow MM^T$  and  $WW^T \rightarrow R$  as number of documents increase

## Step

*We can recover the anchor words from  $MM^T$ .*

## Step

*We can recover  $A$  and  $R$  given  $MM^T$  and the anchor words.*

# Using the Anchor Words

$$= \mathbf{M} \mathbf{M}^T$$

# Using the Anchor Words

$$\begin{matrix} & \mathbf{A} \\ \text{(D)} & \begin{bmatrix} \text{green} & & & \\ & \text{green} & & \\ & & \text{green} & \\ & & & \text{green} \\ \text{blue} & & & \\ \text{(U)} & & \text{blue} & \text{blue} & \\ & \text{blue} & & \text{blue} \\ & & \text{blue} & \text{blue} \\ & & & \end{bmatrix} \\ & \mathbf{W} \mathbf{W}^T \end{matrix} \quad \begin{matrix} & \mathbf{A}^T \\ \begin{bmatrix} \text{green} & & & \text{blue} & & \text{blue} \\ & \text{green} & & & \text{blue} & \\ & & \text{green} & & & \text{blue} \\ & & & \text{green} & & \text{blue} \\ & & & & \text{blue} & \text{blue} \\ & & & & & \end{bmatrix} \\ \text{(R)} & \end{matrix} = \mathbf{M} \mathbf{M}^T$$

# Using the Anchor Words

$$\begin{array}{c}
 \text{(D)} \\
 \text{(U)}
 \end{array}
 \begin{array}{c}
 \mathbf{A} \\
 \mathbf{W} \mathbf{W}^T
 \end{array}
 \begin{array}{c}
 \mathbf{A}^T \\
 \mathbf{R}
 \end{array}
 = \mathbf{M} \mathbf{M}^T$$

The diagram illustrates the matrix factorization  $\mathbf{A} \mathbf{A}^T = \mathbf{M} \mathbf{M}^T$  using anchor words. Matrix  $\mathbf{A}$  is a 5x4 grid with green cells at (1,1), (2,2), (3,3), and (4,4), and blue cells at (5,1), (5,3), (5,4), (5,5), (6,1), (6,3), (6,4), and (6,5). Matrix  $\mathbf{A}^T$  is a 4x5 grid with green cells at (1,1), (2,2), (3,3), and (4,4), and blue cells at (1,5), (1,6), (1,7), (1,8), (2,5), (2,6), (2,7), (2,8), (3,5), (3,6), (3,7), (3,8), and (4,5), (4,6), (4,7), (4,8). The matrices  $\mathbf{W}$  and  $\mathbf{W}^T$  are 5x4 and 4x5 respectively, and  $\mathbf{R}$  is a 4x4 grid. The matrices  $\mathbf{D}$  and  $\mathbf{U}$  are 5x4 and 4x5 respectively, and  $\mathbf{M}$  is a 5x4 grid.

DRD	DRU <sup>T</sup>

# Using the Anchor Words

$$\begin{array}{c}
 \text{(D)} \\
 \text{(U)}
 \end{array}
 \begin{array}{c}
 \mathbf{A} \\
 \mathbf{W} \mathbf{W}^T
 \end{array}
 \begin{array}{c}
 \mathbf{A}^T \\
 \mathbf{R}
 \end{array}
 = \mathbf{M} \mathbf{M}^T$$

The diagram illustrates the matrix factorization  $\mathbf{A} \mathbf{A}^T = \mathbf{M} \mathbf{M}^T$  using anchor words. Matrix  $\mathbf{A}$  (4x4) has green cells at (1,1), (2,2), (3,3), and (4,4), and blue cells at (5,1), (5,2), (5,3), and (5,4). Matrix  $\mathbf{A}^T$  (4x4) has blue cells at (1,5), (2,5), (3,5), and (4,5), and green cells at (1,1), (2,2), (3,3), and (4,4). Matrix  $\mathbf{W}$  (4x4) is the identity matrix, and  $\mathbf{W}^T$  is its transpose. Matrix  $\mathbf{R}$  (4x4) is the identity matrix. The product  $\mathbf{W} \mathbf{W}^T$  is the identity matrix, and  $\mathbf{A} \mathbf{A}^T$  is the matrix  $\mathbf{M} \mathbf{M}^T$ .

$\mathbf{D} \mathbf{R} \mathbf{D}$	$\mathbf{D} \mathbf{R} \mathbf{U}^T$

$$\mathbf{D} \mathbf{R} \vec{1} = \mathbf{D} \mathbf{R} \mathbf{A}^T \vec{1}$$

# Using the Anchor Words

$$\begin{array}{c}
 \text{(D)} \\
 \text{(U)}
 \end{array}
 \begin{array}{c}
 \mathbf{A} \\
 \mathbf{W} \mathbf{W}^T
 \end{array}
 \begin{array}{c}
 \mathbf{A}^T \\
 \mathbf{R}
 \end{array}
 = \mathbf{M} \mathbf{M}^T$$

The diagram illustrates the matrix multiplication  $\mathbf{W} \mathbf{W}^T = \mathbf{M} \mathbf{M}^T$ . The matrix  $\mathbf{W}$  is a 5x4 grid with green squares at (1,1), (2,2), (3,3), and (4,4), and blue squares at (5,1), (5,3), (5,4), (5,5), (5,6), and (5,7). The matrix  $\mathbf{W}^T$  is a 4x7 grid with green squares at (1,1), (2,2), (3,3), and (4,4), and blue squares at (1,5), (1,6), (1,7), (2,5), (2,6), (2,7), (3,5), (3,6), (3,7), and (4,5), (4,6), (4,7). The matrix  $\mathbf{M}$  is a 5x7 grid with blue squares at (5,1), (5,3), (5,4), (5,5), (5,6), and (5,7). The matrix  $\mathbf{M}^T$  is a 7x5 grid with blue squares at (5,1), (5,3), (5,4), (5,5), (5,6), and (5,7).

DRD	DRU <sup>T</sup>

find  $\vec{z}$ :

$$\begin{aligned}
 \text{DR} \vec{1} &= \text{DR} \mathbf{A}^T \vec{1} \\
 \text{DRD} \vec{z} &= \text{DR} \vec{1}
 \end{aligned}$$

# Using the Anchor Words

$$\begin{array}{c}
 \text{(D)} \\
 \text{(U)}
 \end{array}
 \begin{array}{c}
 \mathbf{A} \\
 \mathbf{W} \mathbf{W}^T
 \end{array}
 \begin{array}{c}
 \mathbf{A}^T \\
 \mathbf{R}
 \end{array}
 = \mathbf{M} \mathbf{M}^T$$

DRD	DRU <sup>T</sup>

find  $\vec{z}$ :

$$\begin{aligned}
 \text{DR} \vec{1} &= \text{DR} \mathbf{A}^T \vec{1} \\
 \text{DRD} \vec{z} &= \text{DR} \vec{1}
 \end{aligned}$$

output:  $(\text{DRD} \text{diag}(\mathbf{z}))^{-1} \text{DR} \mathbf{A}^T$

# Our Algorithm

$\hat{M}$  is far from  $M$ , but let's use  $\hat{M}\hat{M}^T$  instead!

$\hat{M}\hat{M}^T \rightarrow MM^T$  and  $WW^T \rightarrow R$  as number of documents increase

## Step

*We can recover the anchor words from  $MM^T$ .*

## Step

*We can recover  $A$  and  $R$  given  $MM^T$  and the anchor words.*

# Our Algorithm

$\hat{M}$  is far from  $M$ , but let's use  $\hat{M}\hat{M}^T$  instead!

$\hat{M}\hat{M}^T \rightarrow MM^T$  and  $WW^T \rightarrow R$  as number of documents increase

## Step

*We can recover the anchor words from  $MM^T$ .*

## Step

*We can recover  $A$  and  $R$  given  $MM^T$  and the anchor words.*

And we can use matrix perturbation bounds to quantify how error accumulates

# Concluding Remarks

joint work with Arora, Ge, Halpern, Mimno, Sontag, Wu and Zhu

We ran our algorithm on a database of 300,000 New York Times articles (from the UCI database) with 30,000 distinct words

# Concluding Remarks

joint work with Arora, Ge, Halpern, Mimno, Sontag, Wu and Zhu

We ran our algorithm on a database of 300,000 New York Times articles (from the UCI database) with 30,000 distinct words

- Run time: 12 minutes (compared to 10 hours for MALLET and other state-of-the-art topic modeling tools)

# Concluding Remarks

joint work with Arora, Ge, Halpern, Mimno, Sontag, Wu and Zhu

We ran our algorithm on a database of 300,000 New York Times articles (from the UCI database) with 30,000 distinct words

- Run time: 12 minutes (compared to 10 hours for MALLET and other state-of-the-art topic modeling tools)
- Topics are **high quality** (Ask me if you want to see the results!)

# Concluding Remarks

joint work with Arora, Ge, Halpern, Mimno, Sontag, Wu and Zhu

We ran our algorithm on a database of 300,000 New York Times articles (from the UCI database) with 30,000 distinct words

- Run time: 12 minutes (compared to 10 hours for MALLET and other state-of-the-art topic modeling tools)
- Topics are **high quality** (Ask me if you want to see the results!)

Independently, [Anandkumar et al] gave an algorithm for LDA without any assumptions!

# Concluding Remarks

joint work with Arora, Ge, Halpern, Mimno, Sontag, Wu and Zhu

We ran our algorithm on a database of 300,000 New York Times articles (from the UCI database) with 30,000 distinct words

- Run time: 12 minutes (compared to 10 hours for MALLET and other state-of-the-art topic modeling tools)
- Topics are **high quality** (Ask me if you want to see the results!)

Independently, [Anandkumar et al] gave an algorithm for LDA without any assumptions!

*Are there other trapdoors – like anchor words – that make machine learning much easier?*

# Questions?

Thanks!