Learning Topic Models

– Going Beyond SVD

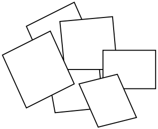
Ankur Moitra, IAS

joint with Sanjeev Arora and Rong Ge

October 21, 2012

Topic Models

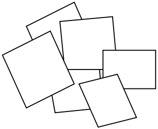
Large collection of articles, say from the New York Times:



newspaper articles

Topic Models

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



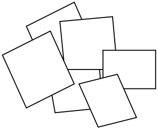
newspaper articles

Question

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

Topic Models

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

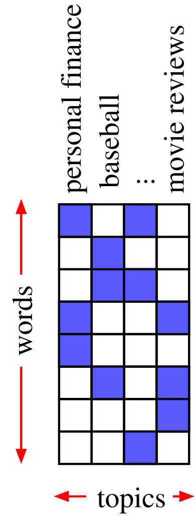


newspaper articles

Question

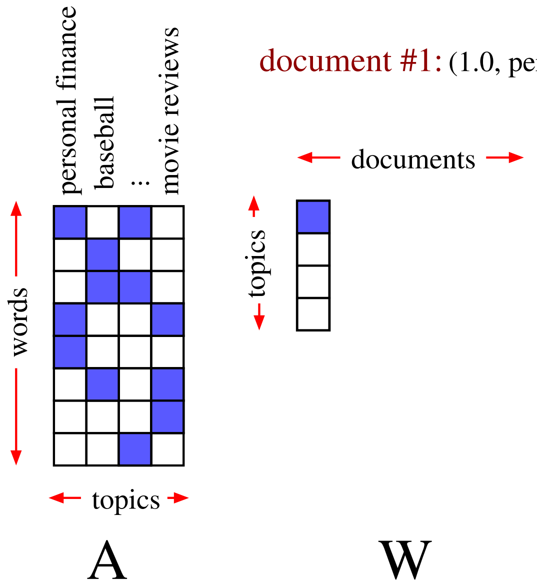
How can we automatically organize them by topic? (unsupervised learning)

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

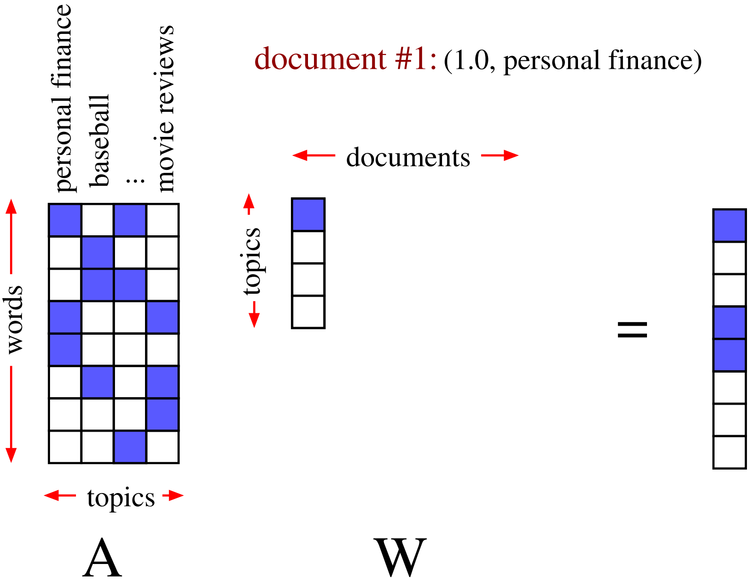


A

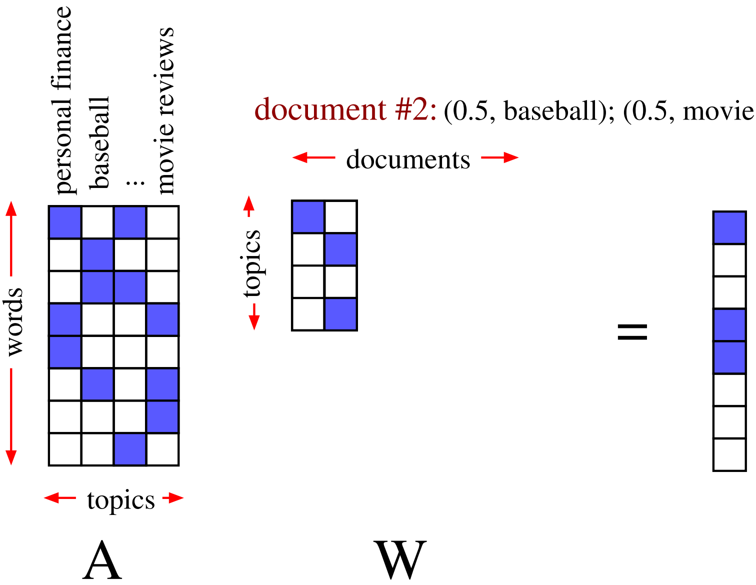
document #1: (1.0, personal finance)



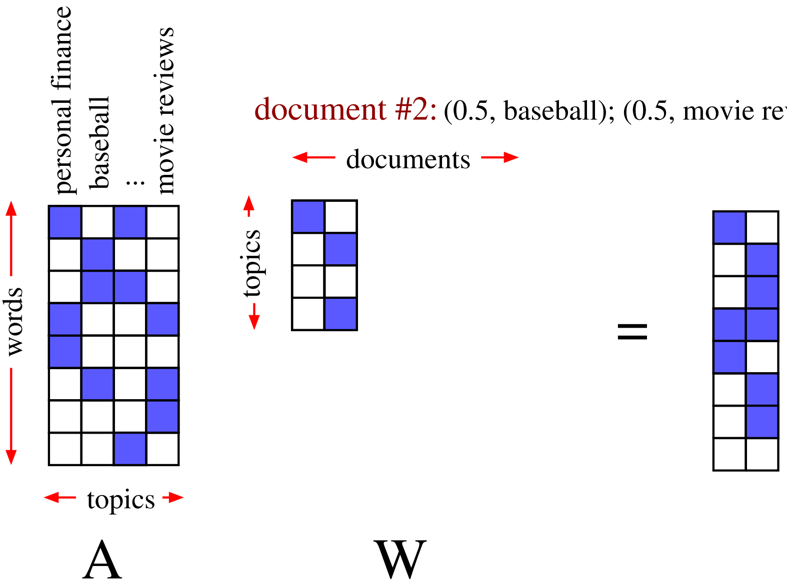
document #1: (1.0, personal finance)

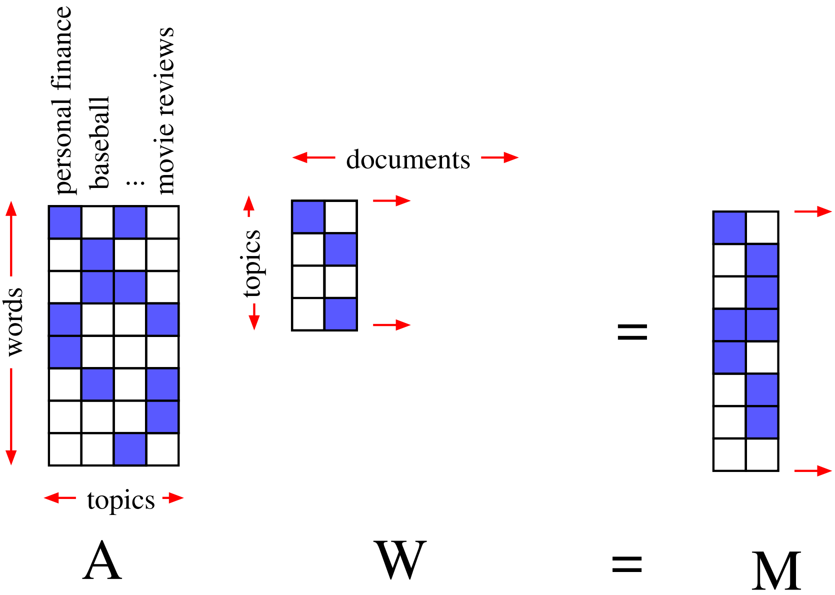


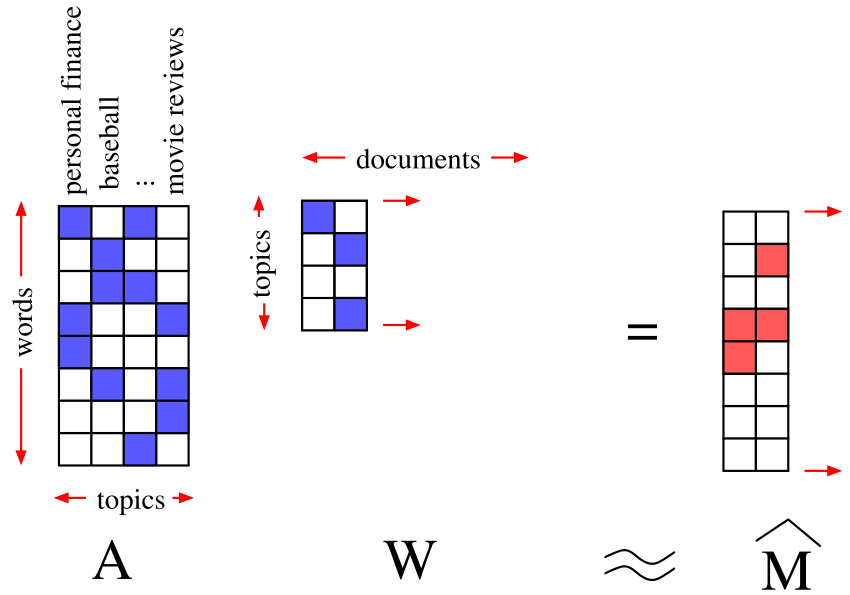
document #2: (0.5, baseball); (0.5, movie reviews)



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



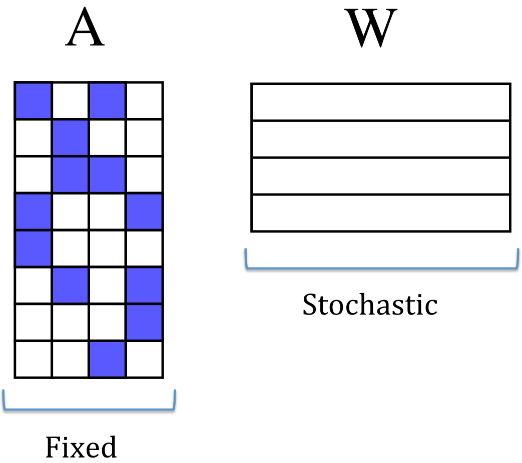

personal finance
baseball
...
movie reviews
words
topics
documents
topics
=
A
W
=
M



personal finance
baseball
...
movie reviews
words
topics
A
documents
topics
W
=
≈
$\widehat{M}$

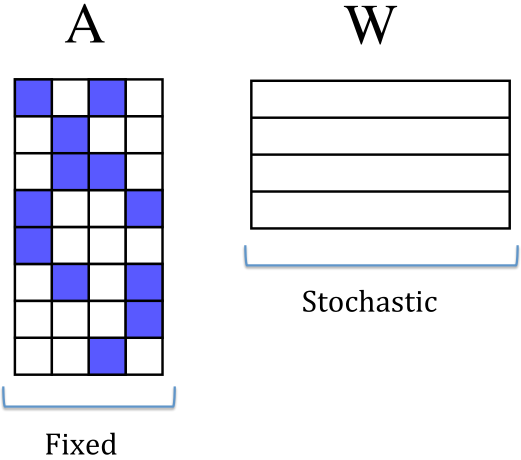
So Many Models!

Pure Topics: one topic per document



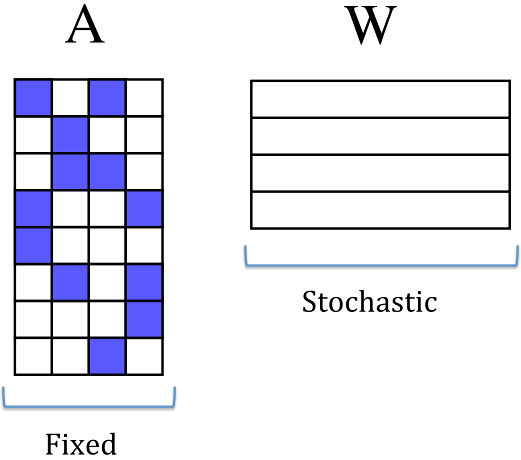
So Many Models!

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 $\widehat{M}$. [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 $\widehat{M}$. [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 $\widehat{M}$. [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 κ 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 κ 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 κ 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 ϵ in time and number of docs 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 κ be its condition number and let $a = \max_{i,j} \frac{E[W_i]}{E[W_j]}$ be the topic imbalance.

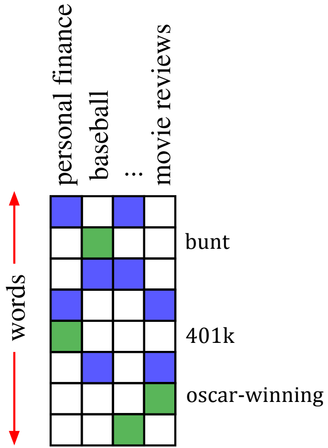
If the topic matrix A satisfies the "anchor word assumption" for $p > 0$:

Theorem

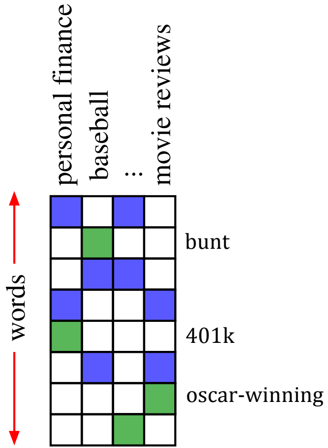
We can learn the topic matrix A and covariance matrix R to within accuracy ϵ in time and number of docs poly$(\log n, r, 1/\epsilon, 1/p, \kappa, a)$ *with n words and r topics*

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:

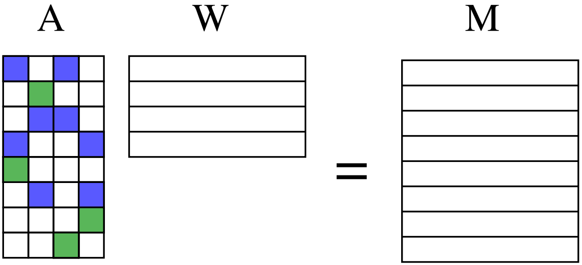


If an anchor word (for a topic) occurs, the document is at least partially about the given topic:



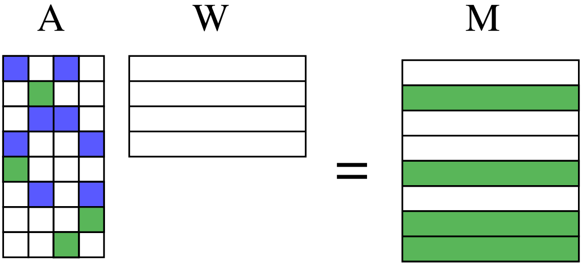
Each topic has an anchor word that occurs with probability $\geq p$

Anchor Words as Extreme Points [AGKM]



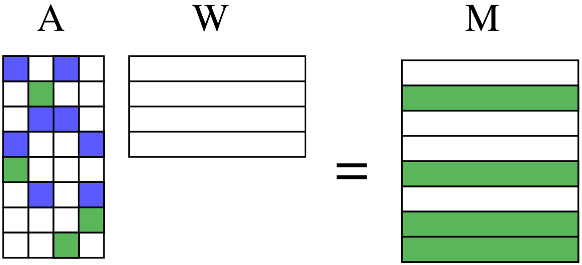
Can we efficiently determine
if a word is an anchor word?

Anchor Words as Extreme Points [AGKM]

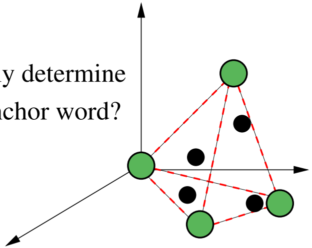


Can we efficiently determine
if a word is an anchor word?

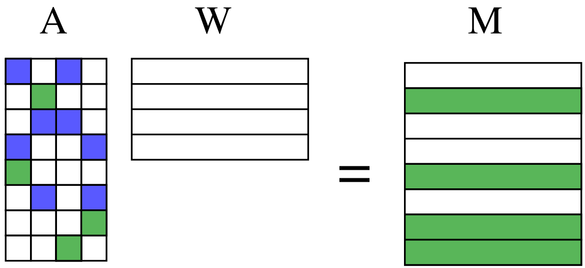
Anchor Words as Extreme Points [AGKM]



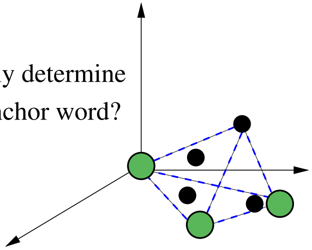
Can we efficiently determine if a word is an anchor word?



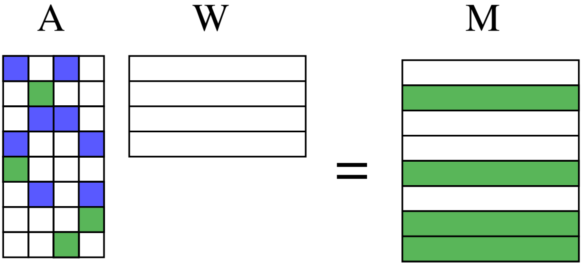
Anchor Words as Extreme Points [AGKM]



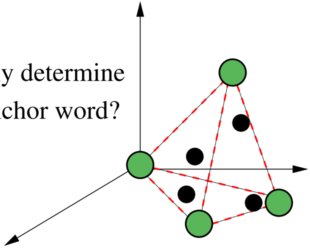
Can we efficiently determine
if a word is an anchor word?



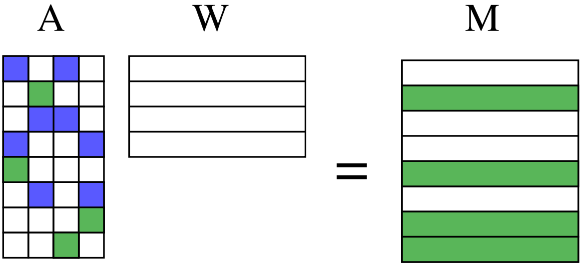
Anchor Words as Extreme Points [AGKM]

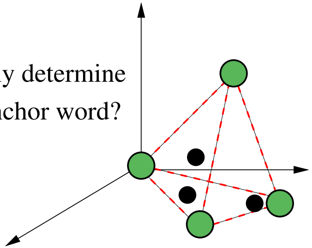


Can we efficiently determine if a word is an anchor word?



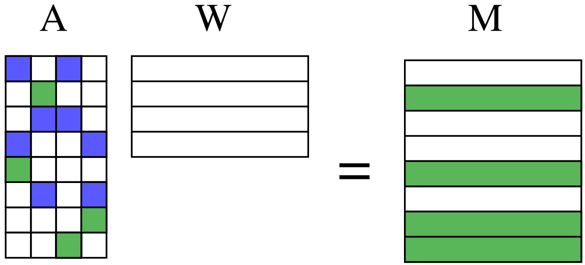
Anchor Words as Extreme Points [AGKM]



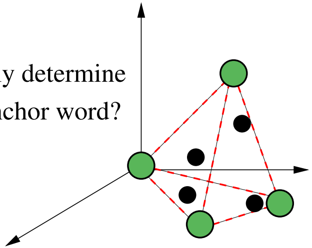


Can we efficiently determine if a word is an anchor word?

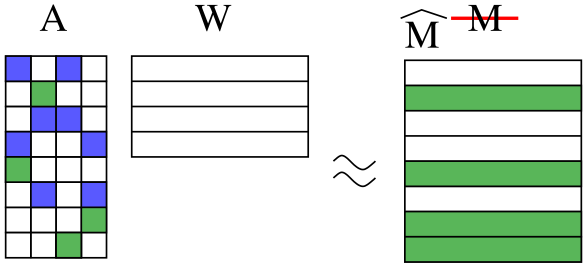
Anchor Words as Extreme Points [AGKM]



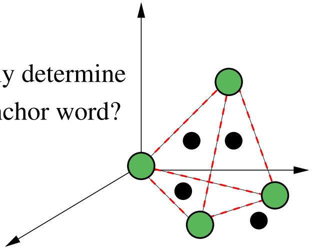
Can we efficiently determine if a word is an anchor word?



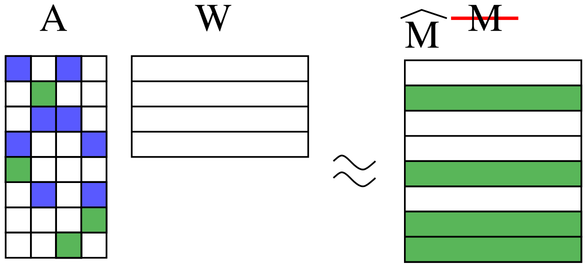
Problem: Sampling "Noise"



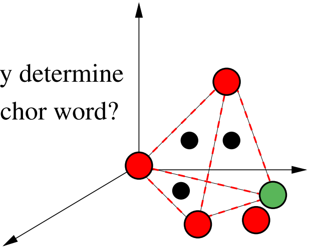
Can we efficiently determine if a word is an anchor word?



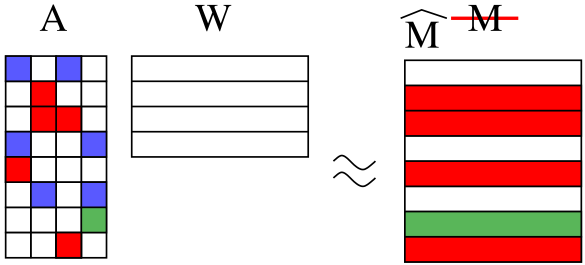
Problem: Sampling "Noise"



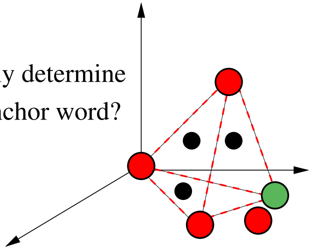
Can we efficiently determine if a word is an anchor word?



Problem: Sampling "Noise"



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



Our Algorithm

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

Our Algorithm

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

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

Our Algorithm

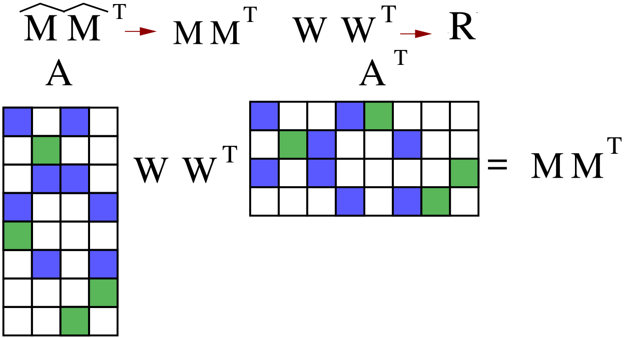
$\widehat{M}$ is far from M, but let's use $\widehat{M}\widehat{M}^T$ instead!

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

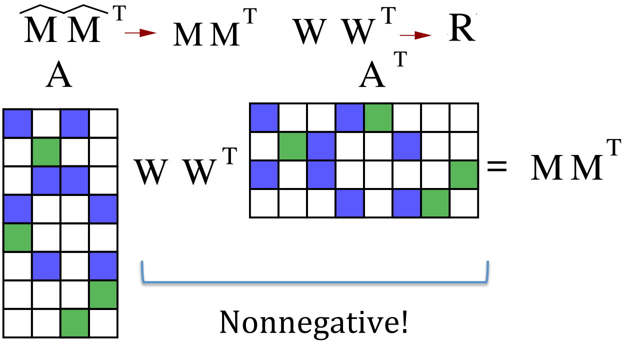
Step

We can recover the anchor words from MM^T.

Finding the Anchor Words

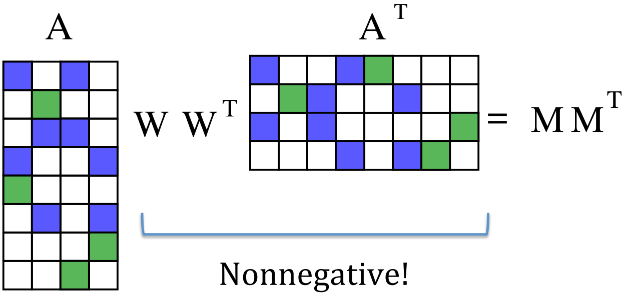


Finding the Anchor Words



Finding the Anchor Words

$\widehat{M}\widehat{M}^T \rightarrow MM^T \quad WW^T \rightarrow R$



Anchor words from: MM^T

Our Algorithm

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

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

Step

We can recover the anchor words from MM^T.

Our Algorithm

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

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

Step

We can recover the anchor words from MM^T.

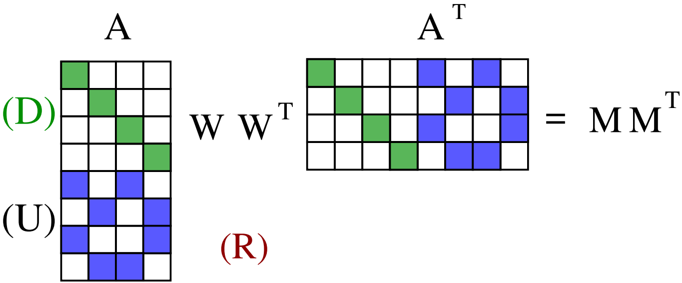
Step

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

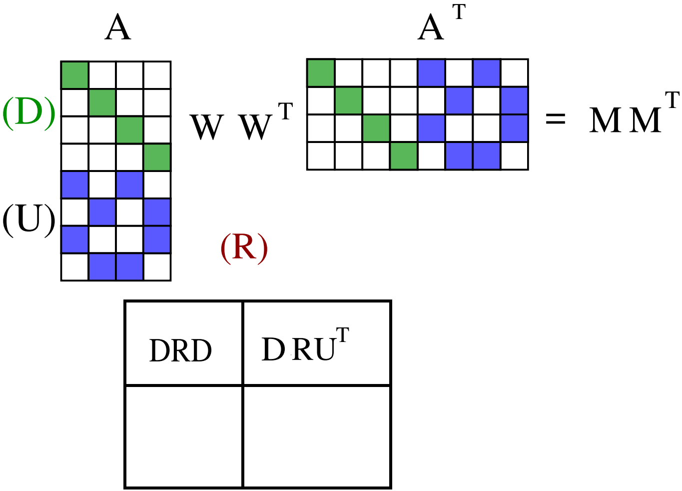
Using the Anchor Words

$$= M M^T$$

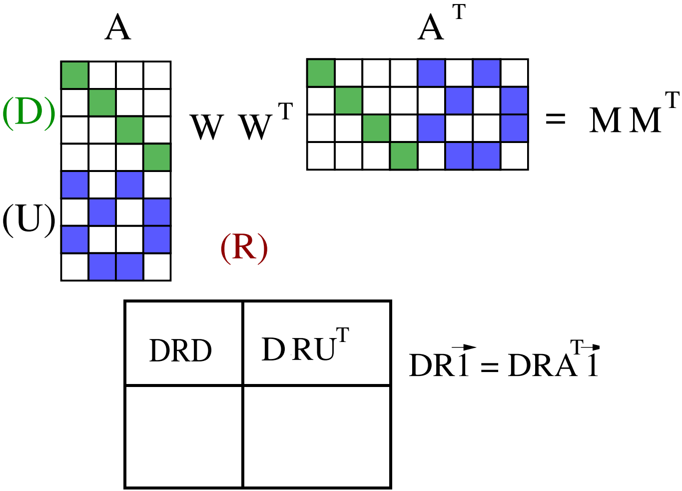
Using the Anchor Words



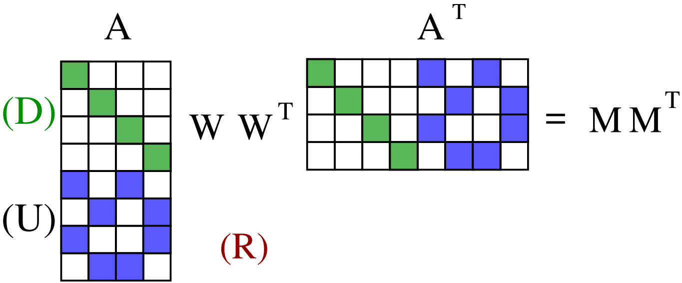
Using the Anchor Words



Using the Anchor Words



Using the Anchor Words



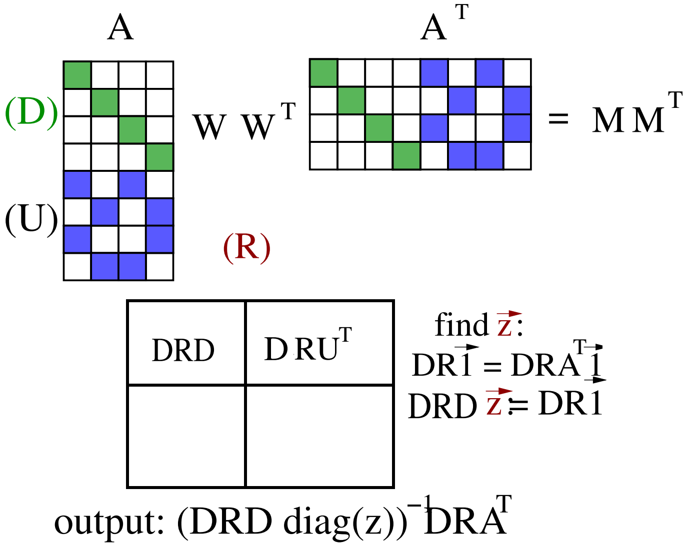
DRD	DRU^T

find $\vec{z}$:

$DR\vec{1} = DRA^T\vec{1}$

$DRD\,\vec{z} = DR\vec{1}$

Using the Anchor Words



Our Algorithm

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

$\widehat{M}\widehat{M}^T \to MM^T$ and $WW^T \to R$ 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

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

$\widehat{M}\widehat{M}^T \to MM^T$ and $WW^T \to R$ 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!