

# TENSOR DECOMPOSITIONS AND THEIR APPLICATIONS

ANKUR MOITRA

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

# SPEARMAN'S HYPOTHESIS

**Charles Spearman (1904):** There are two types of intelligence,  
*eductive and reproductive*

# SPEARMAN'S HYPOTHESIS

**Charles Spearman (1904):** There are two types of intelligence,  
*eductive and reproductive*

eductive (adj): the ability to make sense out of complexity

reproductive (adj): the ability to store and reproduce information

# SPEARMAN'S HYPOTHESIS

**Charles Spearman (1904):** There are two types of intelligence, *eductive* and *reproductive*

To test this theory, he invented **Factor Analysis:**

tests (10)

$$\begin{matrix} \text{students (1000)} \\ M \approx A \\ B^T \end{matrix}$$

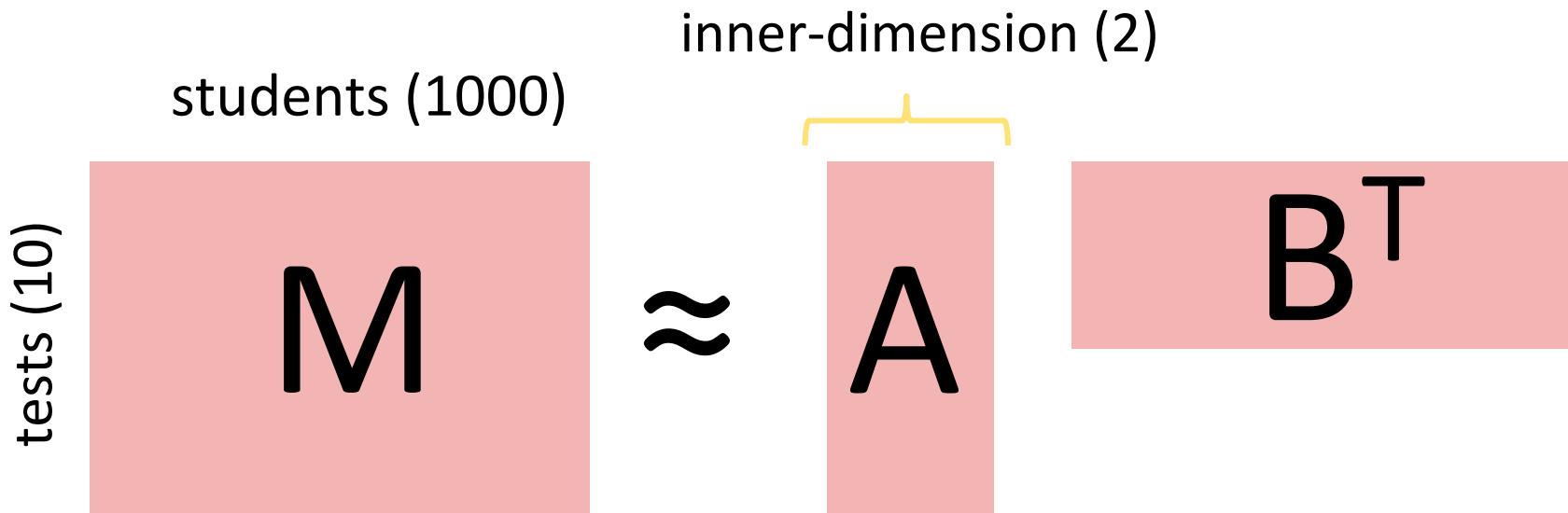
eductive (adj): the ability to make sense out of complexity

reproductive (adj): the ability to store and reproduce information

# SPEARMAN'S HYPOTHESIS

**Charles Spearman (1904):** There are two types of intelligence, *eductive* and *reproductive*

To test this theory, he invented **Factor Analysis:**



eductive (adj): the ability to make sense out of complexity

reproductive (adj): the ability to store and reproduce information

**Given:**  $M = \sum a_i \otimes b_i$

$$= \underbrace{A}_{\text{"correct" factors}} B^T$$

“correct” factors

**Given:**  $M = \sum a_i \otimes b_i$

$$= A \underbrace{B^T}_{\text{"correct" factors}}$$

When can we recover the factors  $a_i$  and  $b_i$  uniquely?

**Given:**  $M = \sum a_i \otimes b_i$

$$= \underbrace{A \quad B^T}_{\text{"correct" factors}} = \underbrace{AR \quad R^{-1}B^T}_{\text{alternative factorization}}$$

When can we recover the factors  $a_i$  and  $b_i$  uniquely?

**Given:**  $M = \sum a_i \otimes b_i$

$$= \underbrace{A \quad B^T}_{\text{"correct" factors}} = \underbrace{AR \quad R^{-1}B^T}_{\text{alternative factorization}}$$

When can we recover the factors  $a_i$  and  $b_i$  uniquely?

**Claim:** The factors  $\{a_i\}$  and  $\{b_i\}$  are not determined uniquely unless we impose additional conditions on them

**Given:**  $M = \sum a_i \otimes b_i$

$$= \underbrace{A \quad B^T}_{\text{"correct" factors}} = \underbrace{AR \quad R^{-1}B^T}_{\text{alternative factorization}}$$

When can we recover the factors  $a_i$  and  $b_i$  uniquely?

**Claim:** The factors  $\{a_i\}$  and  $\{b_i\}$  are not determined uniquely unless we impose additional conditions on them

e.g. if  $\{a_i\}$  and  $\{b_i\}$  are orthogonal, or  $\text{rank}(M)=1$

**Given:**  $M = \sum a_i \otimes b_i$

$$= \underbrace{A \quad B^T}_{\text{"correct" factors}} = \underbrace{AR \quad R^{-1}B^T}_{\text{alternative factorization}}$$

When can we recover the factors  $a_i$  and  $b_i$  uniquely?

**Claim:** The factors  $\{a_i\}$  and  $\{b_i\}$  are not determined uniquely unless we impose additional conditions on them

e.g. if  $\{a_i\}$  and  $\{b_i\}$  are orthogonal, or  $\text{rank}(M)=1$

---

This is called the **rotation problem**, and is a major issue in factor analysis and motivates the study of **tensor methods**...

# OUTLINE

The focus of this tutorial is on Algorithms/Applications/Models for tensor decompositions

## Part I: Algorithms

- The Rotation Problem
- Jennrich's Algorithm

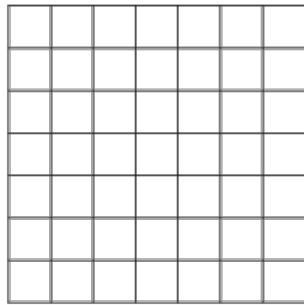
## Part II: Applications

- Phylogenetic Reconstruction
- Pure Topic Models

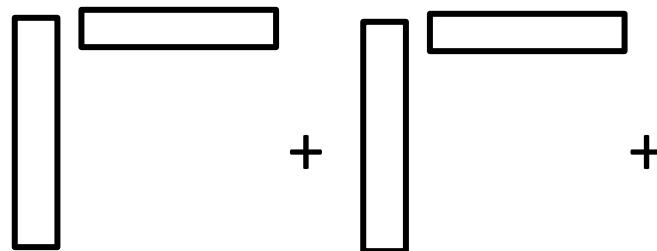
## Part III: Smoothed Analysis

- Overcomplete Problems
- Kruskal Rank and the Khatri-Rao Product

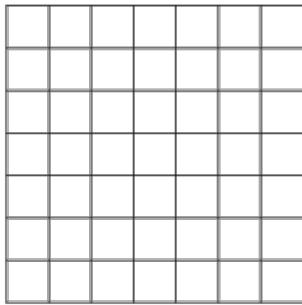
# MATRIX DECOMPOSITIONS



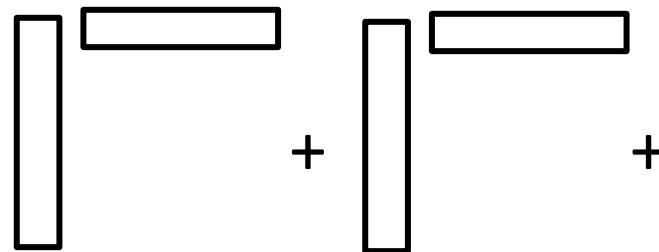
$$M = a_1 \otimes b_1 + a_2 \otimes b_2 + \cdots + a_R \otimes b_R$$



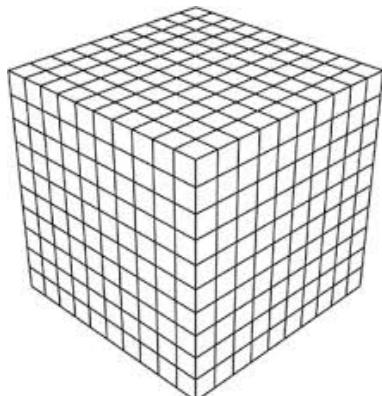
# MATRIX DECOMPOSITIONS



$$M = a_1 \otimes b_1 + a_2 \otimes b_2 + \cdots + a_R \otimes b_R$$



# TENSOR DECOMPOSITIONS



$$T = a_1 \otimes b_1 \otimes c_1 + \cdots + a_R \otimes b_R \otimes c_R$$

$(i, j, k)$  entry of  $x \otimes y \otimes z$  is  $x(i) \times y(j) \times z(k)$

When are tensor decompositions unique?

## When are tensor decompositions unique?

**Theorem [Jennrich 1970]:** Suppose  $\{a_i\}$  and  $\{b_i\}$  are linearly independent and no pair of vectors in  $\{c_i\}$  is a scalar multiple of each other...

## When are tensor decompositions unique?

**Theorem [Jennrich 1970]:** Suppose  $\{a_i\}$  and  $\{b_i\}$  are linearly independent and no pair of vectors in  $\{c_i\}$  is a scalar multiple of each other. Then

$$T = a_1 \otimes b_1 \otimes c_1 + \cdots + a_R \otimes b_R \otimes c_R$$

is unique up to permuting the rank one terms and rescaling the factors.

## When are tensor decompositions unique?

**Theorem [Jennrich 1970]:** Suppose  $\{a_i\}$  and  $\{b_i\}$  are linearly independent and no pair of vectors in  $\{c_i\}$  is a scalar multiple of each other. Then

$$T = a_1 \otimes b_1 \otimes c_1 + \cdots + a_R \otimes b_R \otimes c_R$$

is unique up to permuting the rank one terms and rescaling the factors.

Equivalently, the rank one factors are **unique**

## When are tensor decompositions unique?

**Theorem [Jennrich 1970]:** Suppose  $\{a_i\}$  and  $\{b_i\}$  are linearly independent and no pair of vectors in  $\{c_i\}$  is a scalar multiple of each other. Then

$$T = a_1 \otimes b_1 \otimes c_1 + \cdots + a_R \otimes b_R \otimes c_R$$

is unique up to permuting the rank one terms and rescaling the factors.

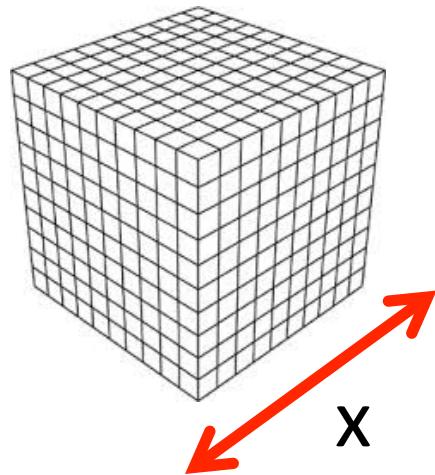
Equivalently, the rank one factors are **unique**

---

There is a simple algorithm to compute the factors too!

# JENNICH'S ALGORITHM

► Compute  $T(\bullet, \bullet, x)$

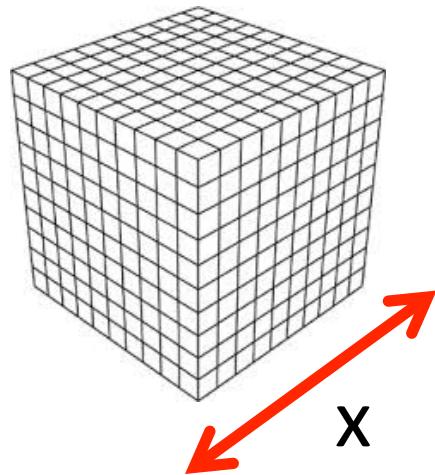


i.e. add up matrix slices

$$\sum x_i T_i$$

# JENNICH'S ALGORITHM

► Compute  $T(\bullet, \bullet, x)$



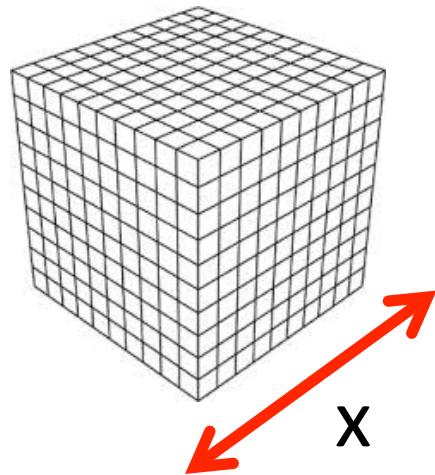
i.e. add up matrix slices

$$\sum x_i T_i$$

If  $T = a \otimes b \otimes c$  then  $T(\bullet, \bullet, x) = \langle c, x \rangle a \otimes b$

# JENNICH'S ALGORITHM

► Compute  $T(\bullet, \bullet, x)$

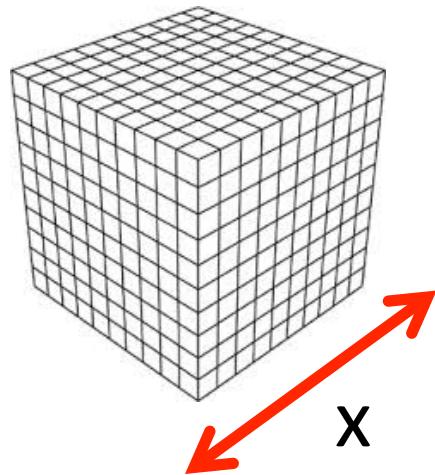


i.e. add up matrix slices

$$\sum x_i T_i$$

## JENNICH'S ALGORITHM

➡ Compute  $T(\bullet, \bullet, x) = \sum \langle c_i, x \rangle a_i \otimes b_i$

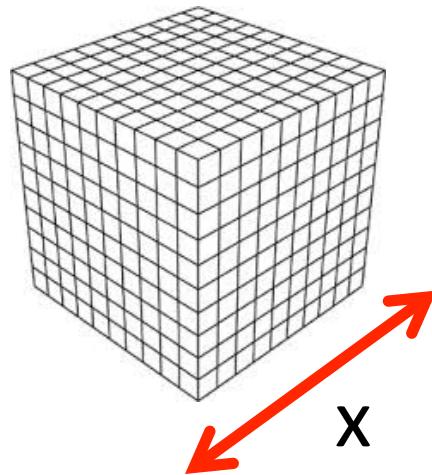


i.e. add up matrix slices

$$\sum x_i T_i$$

## JENNICH'S ALGORITHM

➡ Compute  $T(\bullet, \bullet, x) = \sum \langle c_i, x \rangle a_i \otimes b_i$



i.e. add up matrix slices

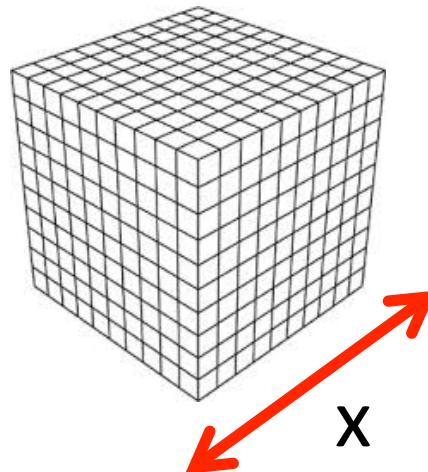
$$\sum x_i T_i$$

( $x$  is chosen uniformly at random from  $S^{n-1}$ )

## JENNICH'S ALGORITHM

Diag( $\langle c_i, x \rangle$ )

Compute  $T(\bullet, \bullet, x) = A D_x B^T$



i.e. add up matrix slices

$$\sum x_i T_i$$

( $x$  is chosen uniformly at random from  $S^{n-1}$ )

## JENNICH'S ALGORITHM

→ Compute  $T(\bullet, \bullet, x) = A D_x B^T$

## JENNICH'S ALGORITHM

- ▶ Compute  $T(\bullet, \bullet, x) = A D_x B^T$
- ▶ Compute  $T(\bullet, \bullet, y) = A D_y B^T$

## JENNICH'S ALGORITHM

- ▶ Compute  $T(\bullet, \bullet, x) = A D_x B^T$
- ▶ Compute  $T(\bullet, \bullet, y) = A D_y B^T$
- ▶ Diagonalize  $T(\bullet, \bullet, x) T(\bullet, \bullet, y)^{-1}$

# JENNICH'S ALGORITHM

- ▶ Compute  $T(\bullet, \bullet, x) = A D_x B^T$
- ▶ Compute  $T(\bullet, \bullet, y) = A D_y B^T$
- ▶ Diagonalize  $T(\bullet, \bullet, x) T(\bullet, \bullet, y)^{-1}$



$$A D_x B^T (B^T)^{-1} D_y^{-1} A^{-1}$$

## JENNICH'S ALGORITHM

- ▶ Compute  $T(\bullet, \bullet, x) = A D_x B^T$
- ▶ Compute  $T(\bullet, \bullet, y) = A D_y B^T$
- ▶ Diagonalize  $T(\bullet, \bullet, x) T(\bullet, \bullet, y)^{-1}$



$$A D_x D_y^{-1} A^{-1}$$

## JENNICH'S ALGORITHM

- ▶ Compute  $T(\bullet, \bullet, x) = A D_x B^T$
- ▶ Compute  $T(\bullet, \bullet, y) = A D_y B^T$
- ▶ Diagonalize  $T(\bullet, \bullet, x) T(\bullet, \bullet, y)^{-1}$



$$A D_x D_y^{-1} A^{-1}$$

---

**Claim:** whp (over  $x, y$ ) the eigenvalues are distinct, so the Eigendecomposition is unique and recovers  $a_i$ 's

## JENNICH'S ALGORITHM

- ▶ Compute  $T(\bullet, \bullet, x) = A D_x B^T$
- ▶ Compute  $T(\bullet, \bullet, y) = A D_y B^T$
- ▶ Diagonalize  $T(\bullet, \bullet, x) T(\bullet, \bullet, y)^{-1}$

## JENNICH'S ALGORITHM

- ▶ Compute  $T(\bullet, \bullet, x) = A D_x B^T$
- ▶ Compute  $T(\bullet, \bullet, y) = A D_y B^T$
- ▶ Diagonalize  $T(\bullet, \bullet, x) T(\bullet, \bullet, y)^{-1}$
- ▶ Diagonalize  $T(\bullet, \bullet, y) T(\bullet, \bullet, x)^{-1}$

## JENNICH'S ALGORITHM

- ▶ Compute  $T(\bullet, \bullet, x) = A D_x B^T$
- ▶ Compute  $T(\bullet, \bullet, y) = A D_y B^T$
- ▶ Diagonalize  $T(\bullet, \bullet, x) T(\bullet, \bullet, y)^{-1}$
- ▶ Diagonalize  $T(\bullet, \bullet, y) T(\bullet, \bullet, x)^{-1}$
- ▶ Match up the factors (their eigenvalues are reciprocals) and find  $\{c_i\}$  by solving a linear syst.

**Given:**  $M = \sum a_i \otimes b_i$

When can we recover the factors  $a_i$  and  $b_i$  uniquely?

This is only possible if  $\{a_i\}$  and  $\{b_i\}$  are orthonormal, or  $\text{rank}(M)=1$

**Given:**  $M = \sum a_i \otimes b_i$

When can we recover the factors  $a_i$  and  $b_i$  uniquely?

This is only possible if  $\{a_i\}$  and  $\{b_i\}$  are orthonormal, or  $\text{rank}(M)=1$

**Given:**  $T = \sum a_i \otimes b_i \otimes c_i$

When can we recover the factors  $a_i$ ,  $b_i$  and  $c_i$  uniquely?

**Given:**  $M = \sum a_i \otimes b_i$

When can we recover the factors  $a_i$  and  $b_i$  uniquely?

This is only possible if  $\{a_i\}$  and  $\{b_i\}$  are orthonormal, or  $\text{rank}(M)=1$

**Given:**  $T = \sum a_i \otimes b_i \otimes c_i$

When can we recover the factors  $a_i$ ,  $b_i$  and  $c_i$  uniquely?

**Jennrich:** If  $\{a_i\}$  and  $\{b_i\}$  are full rank and no pair in  $\{c_i\}$  are scalar multiples of each other

# OUTLINE

The focus of this tutorial is on Algorithms/Applications/Models for tensor decompositions

## Part I: Algorithms

- The Rotation Problem
- Jennrich's Algorithm

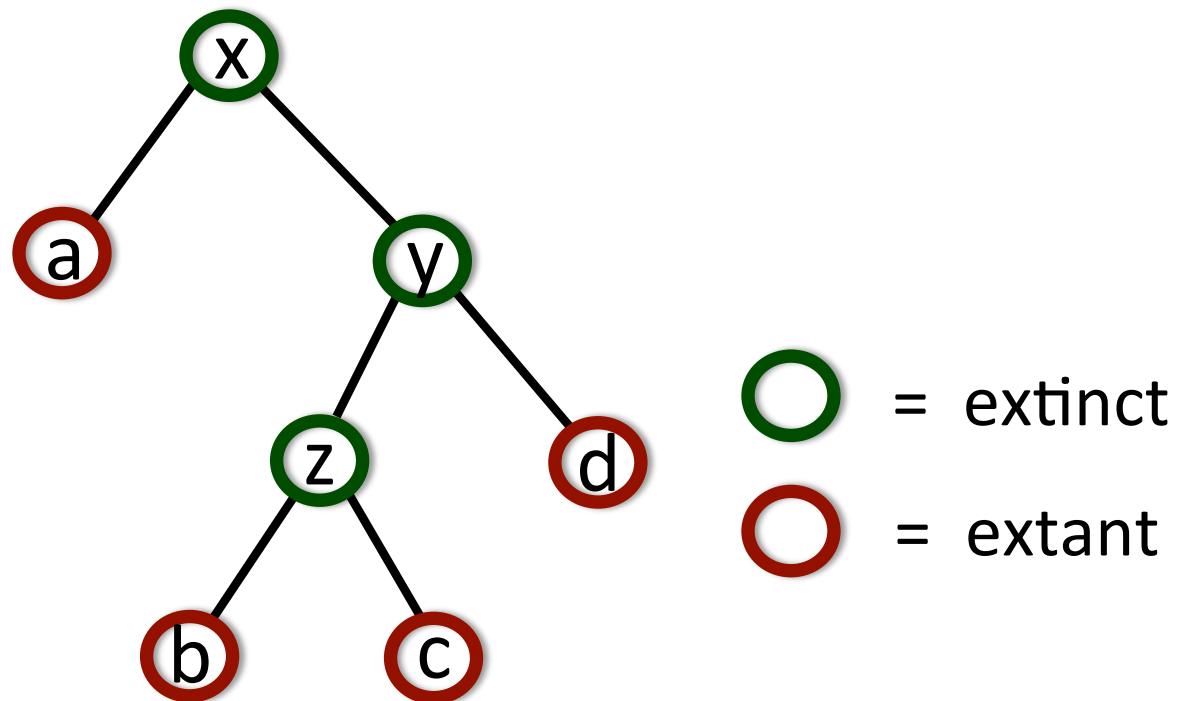
## Part II: Applications

- Phylogenetic Reconstruction
- Pure Topic Models

## Part III: Smoothed Analysis

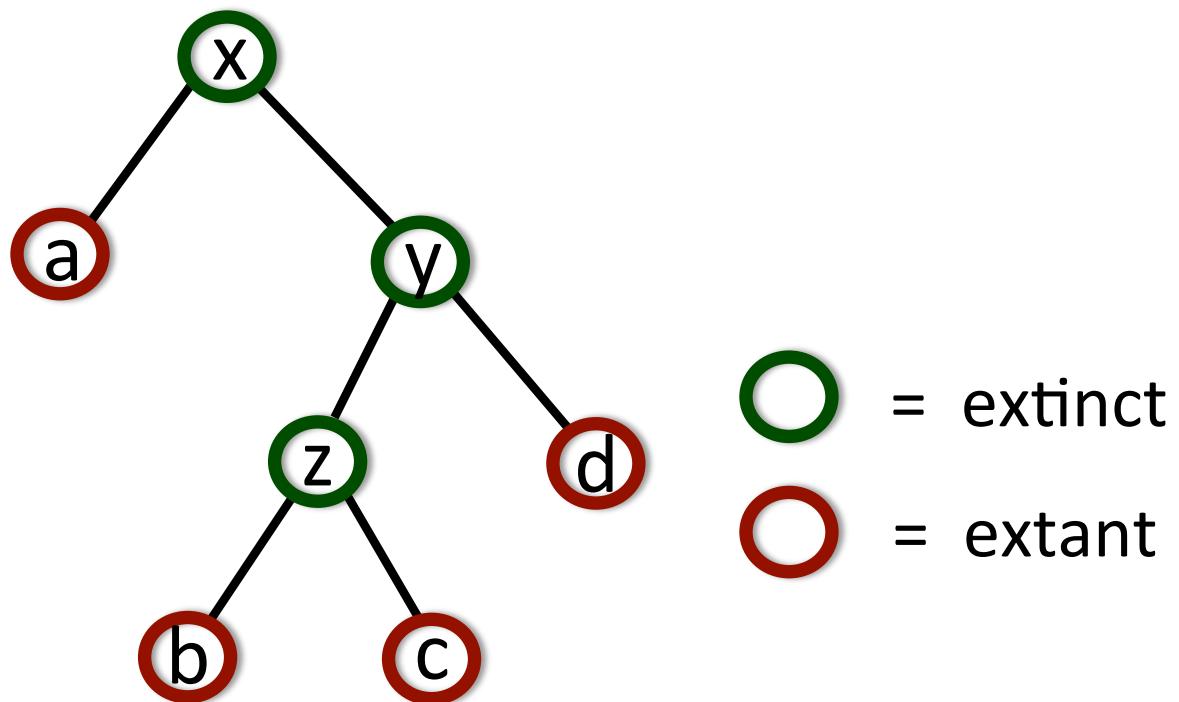
- Overcomplete Problems
- Kruskal Rank and the Khatri-Rao Product

# PHYLOGENETIC RECONSTRUCTION

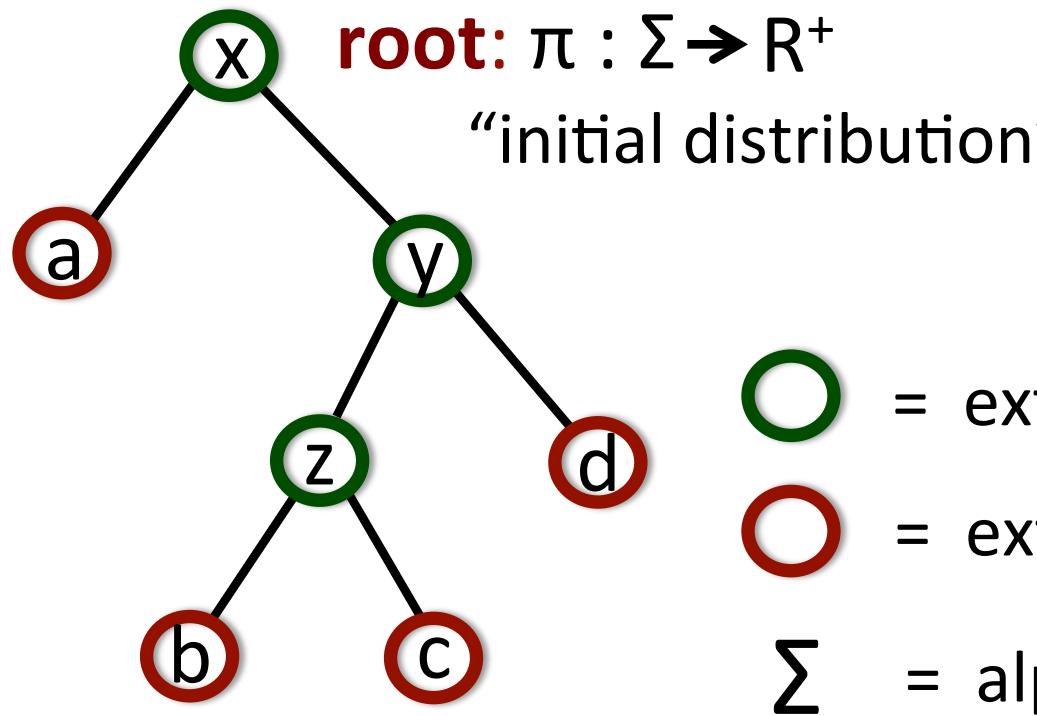


“Tree of Life”

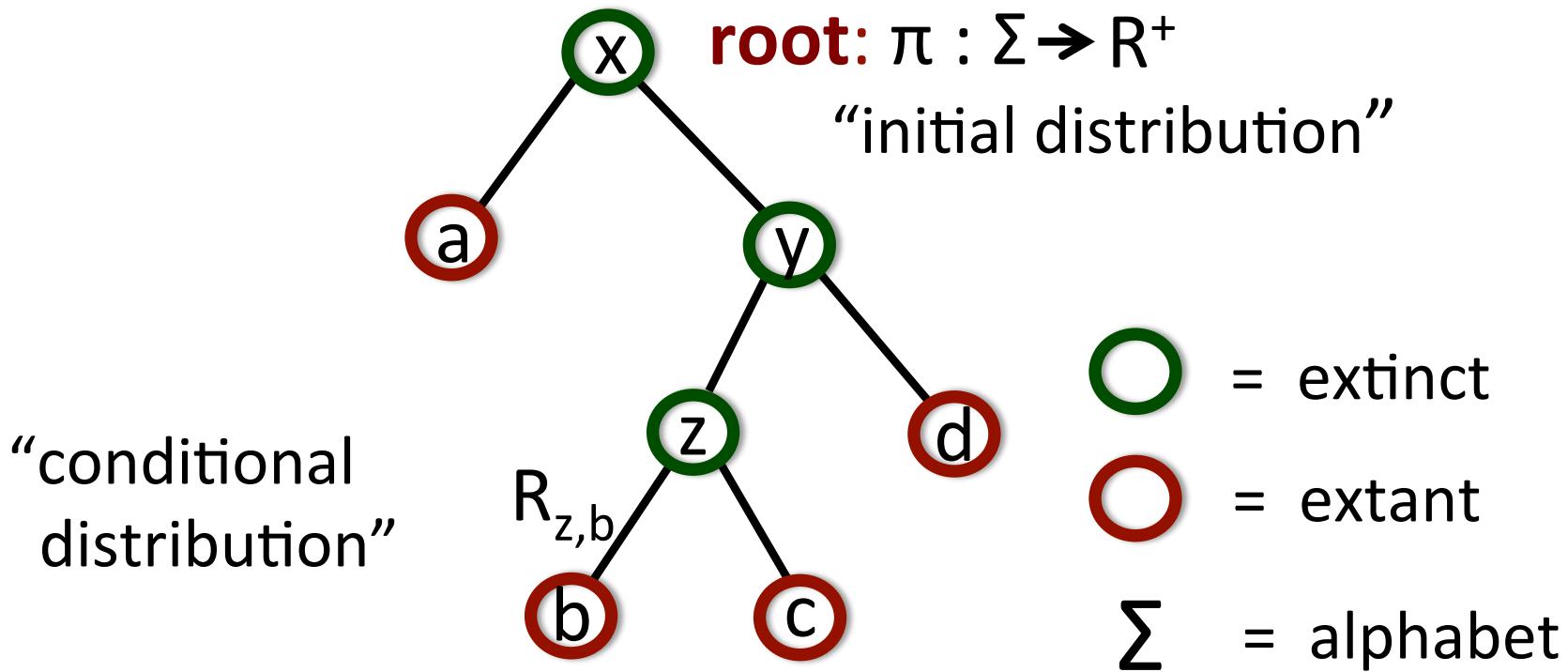
# PHYLOGENETIC RECONSTRUCTION



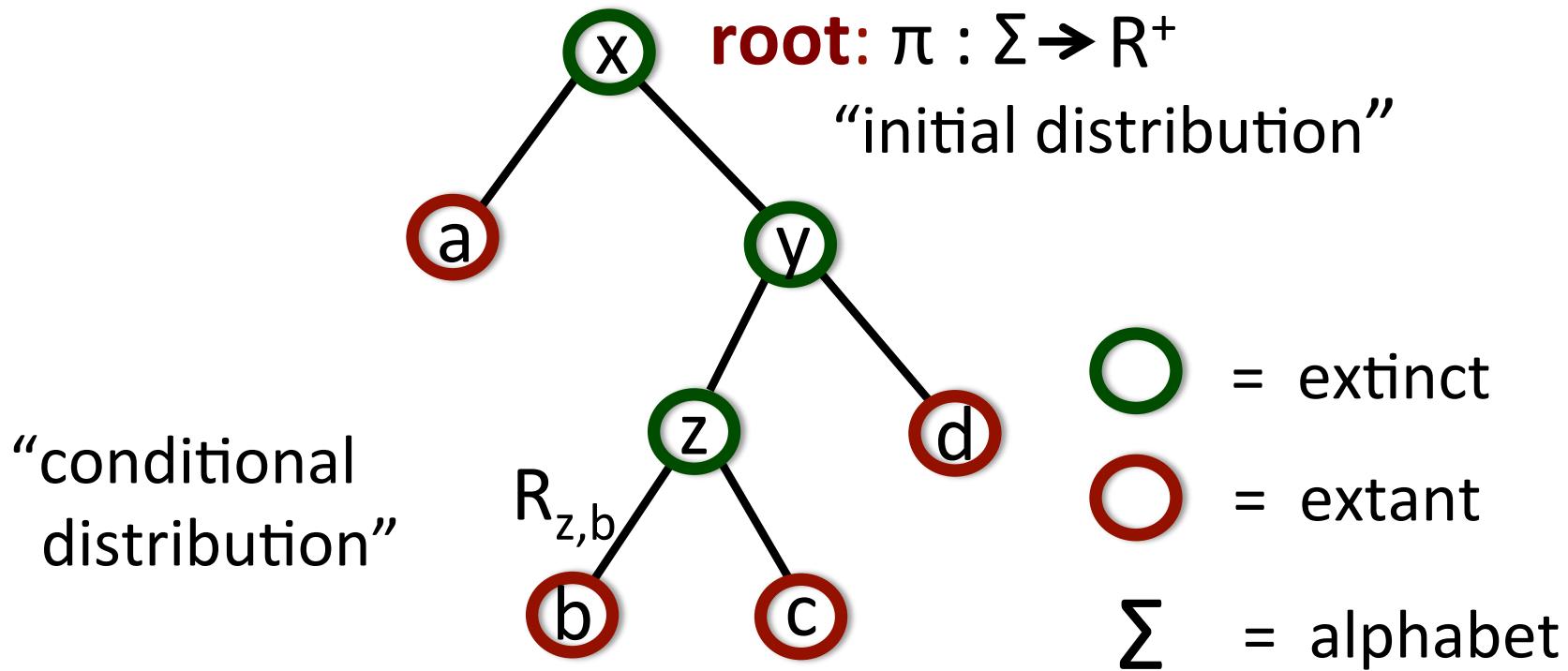
# PHYLOGENETIC RECONSTRUCTION



# PHYLOGENETIC RECONSTRUCTION



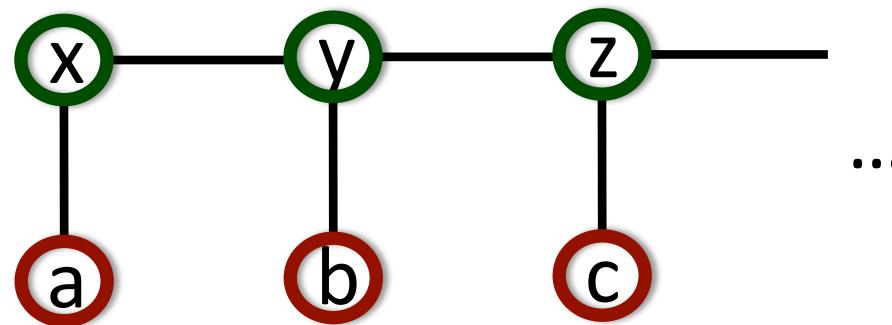
# PHYLOGENETIC RECONSTRUCTION



In each sample, we observe a symbol ( $\Sigma$ ) at each extant (red circle) node where we sample from  $\pi$  for the root, and propagate it using  $R_{x,y}$ , etc

# HIDDEN MARKOV MODELS

○ = hidden  
○ = observed

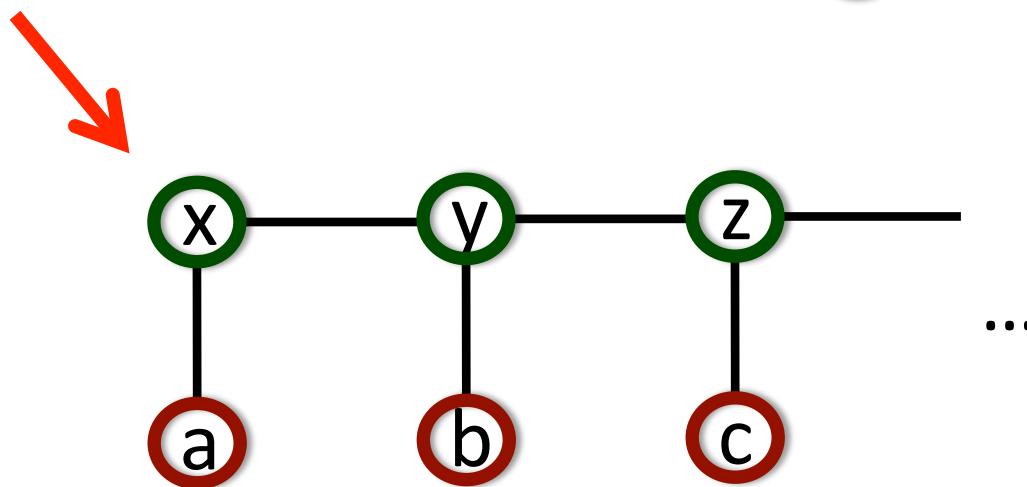


# HIDDEN MARKOV MODELS

$$\pi : \Sigma_s \rightarrow R^+$$

“initial distribution”

 = hidden  
 = observed

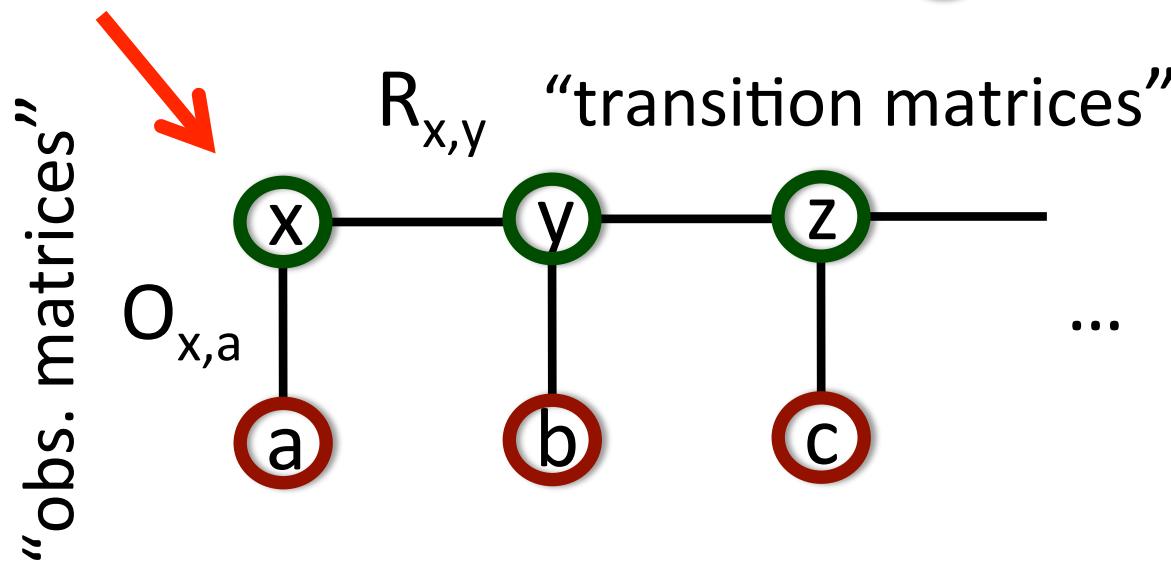


# HIDDEN MARKOV MODELS

$$\pi : \Sigma_s \rightarrow R^+$$

“initial distribution”

 = hidden  
 = observed

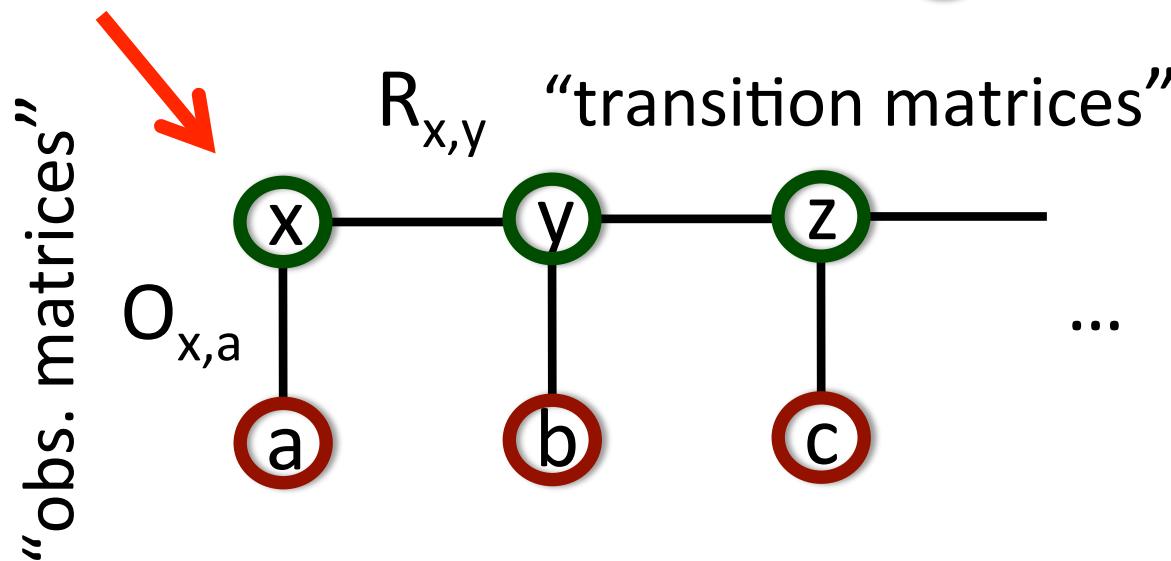


# HIDDEN MARKOV MODELS

$$\pi : \Sigma_s \rightarrow R^+$$

“initial distribution”

 = hidden  
 = observed



In each sample, we observe a symbol ( $\Sigma_o$ ) at each obs. (red circle) node where we sample from  $\pi$  for the start, and propagate it using  $R_{x,y}$ , etc ( $\Sigma_s$ )

**Question:** Can we reconstruct just the topology from random samples?

**Question:** Can we reconstruct just the topology from random samples?

Usually, we assume  $T_{x,y}$ , etc are full rank so that we can re-root the tree arbitrarily

**Question:** Can we reconstruct just the topology from random samples?

Usually, we assume  $T_{x,y}$ , etc are full rank so that we can re-root the tree arbitrarily

**[Steel, 1994]:** The following is a distance function on the edges

$$d_{x,y} = -\ln |\det(P_{x,y})| + \frac{1}{2} \ln \prod_{\sigma \text{ in } \Sigma} \pi_{x,\sigma} - \frac{1}{2} \ln \prod_{\sigma \text{ in } \Sigma} \pi_{y,\sigma}$$

where  $P_{x,y}$  is the joint distribution

**Question:** Can we reconstruct just the topology from random samples?

Usually, we assume  $T_{x,y}$ , etc are full rank so that we can re-root the tree arbitrarily

**[Steel, 1994]:** The following is a distance function on the edges

$$d_{x,y} = -\ln |\det(P_{x,y})| + \frac{1}{2} \ln \prod_{\sigma \text{ in } \Sigma} \pi_{x,\sigma} - \frac{1}{2} \ln \prod_{\sigma \text{ in } \Sigma} \pi_{y,\sigma}$$

where  $P_{x,y}$  is the joint distribution, and the distance between leaves is the sum of distances on the path in the tree

**Question:** Can we reconstruct just the topology from random samples?

Usually, we assume  $T_{x,y}$ , etc are full rank so that we can re-root the tree arbitrarily

**[Steel, 1994]:** The following is a distance function on the edges

$$d_{x,y} = -\ln |\det(P_{x,y})| + \frac{1}{2} \ln \prod_{\sigma \text{ in } \Sigma} \pi_{x,\sigma} - \frac{1}{2} \ln \prod_{\sigma \text{ in } \Sigma} \pi_{y,\sigma}$$

where  $P_{x,y}$  is the joint distribution, and the distance between leaves is the sum of distances on the path in the tree

**(It's not even obvious it's nonnegative!)**

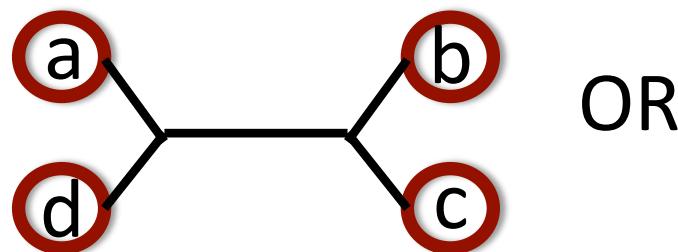
**Question:** Can we reconstruct just the topology from random samples?

Usually, we assume  $T_{x,y}$ , etc are full rank so that we can re-root the tree arbitrarily

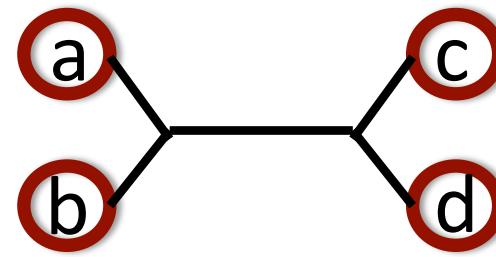
**Question:** Can we reconstruct just the topology from random samples?

Usually, we assume  $T_{x,y}$ , etc are full rank so that we can re-root the tree arbitrarily

**[Erdos, Steel, Szekely, Warnow, 1997]:** Used Steel's distance function and quartet tests



OR



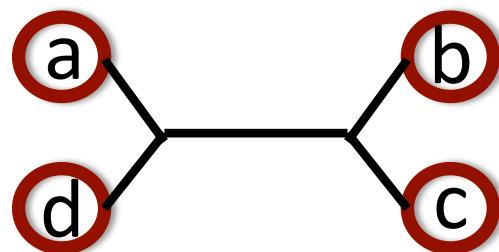
OR ...

to reconstruction the topology

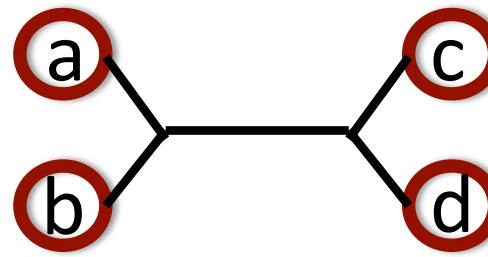
**Question:** Can we reconstruct just the topology from random samples?

Usually, we assume  $T_{x,y}$ , etc are full rank so that we can re-root the tree arbitrarily

**[Erdos, Steel, Szekely, Warnow, 1997]:** Used Steel's distance function and quartet tests



OR



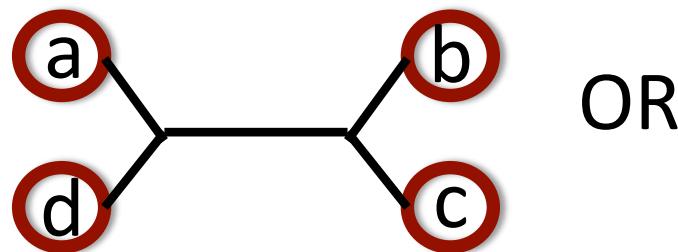
OR ...

to reconstruction the topology, from polynomially many samples

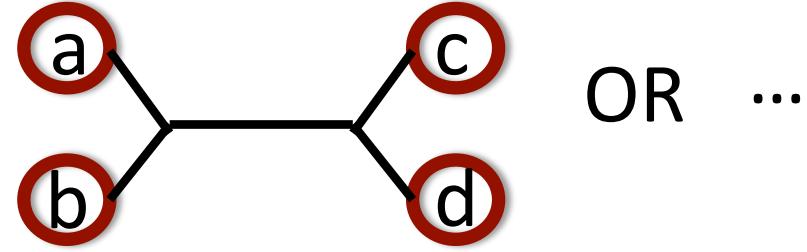
**Question:** Can we reconstruct just the topology from random samples?

Usually, we assume  $T_{x,y}$ , etc are full rank so that we can re-root the tree arbitrarily

**[Erdos, Steel, Szekely, Warnow, 1997]:** Used Steel's distance function and quartet tests



OR



OR ...

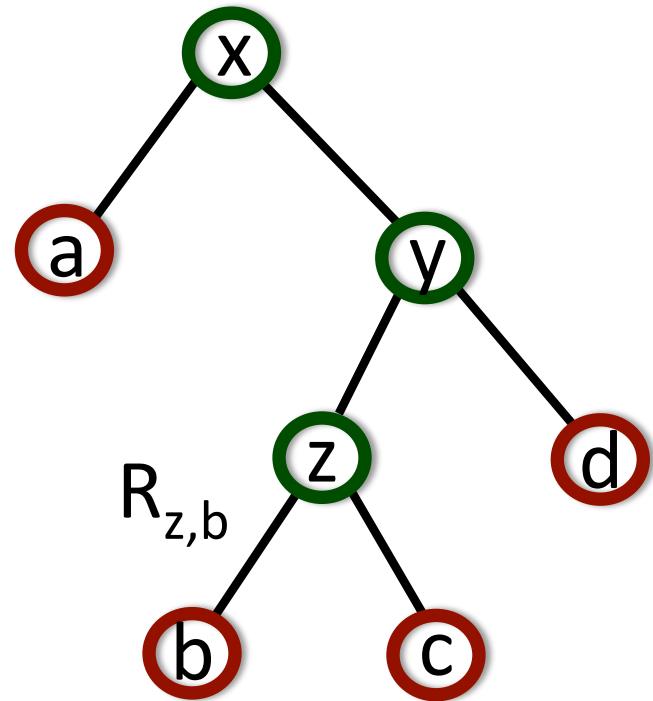
to reconstruction the topology, from polynomially many samples

---

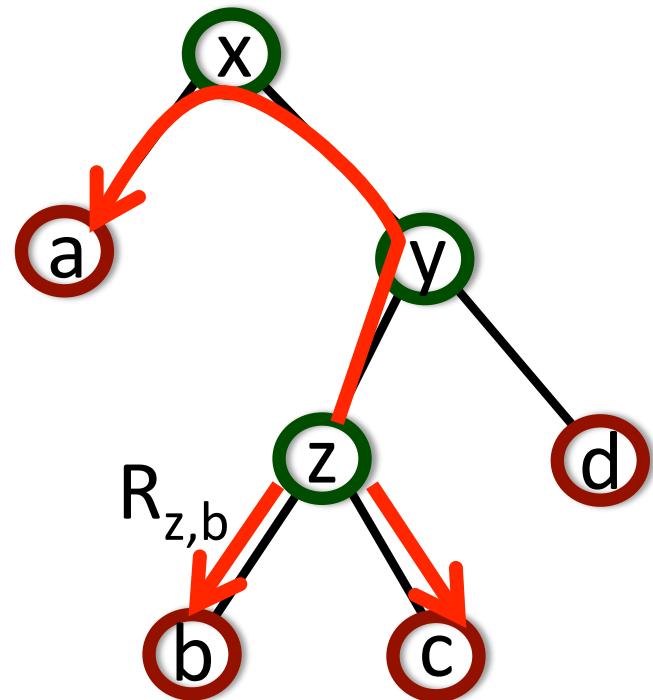
For many problems (e.g. HMMs) finding the transition matrices is the main issue...

**[Chang, 1996]:** The model is identifiable (if R's are full rank)

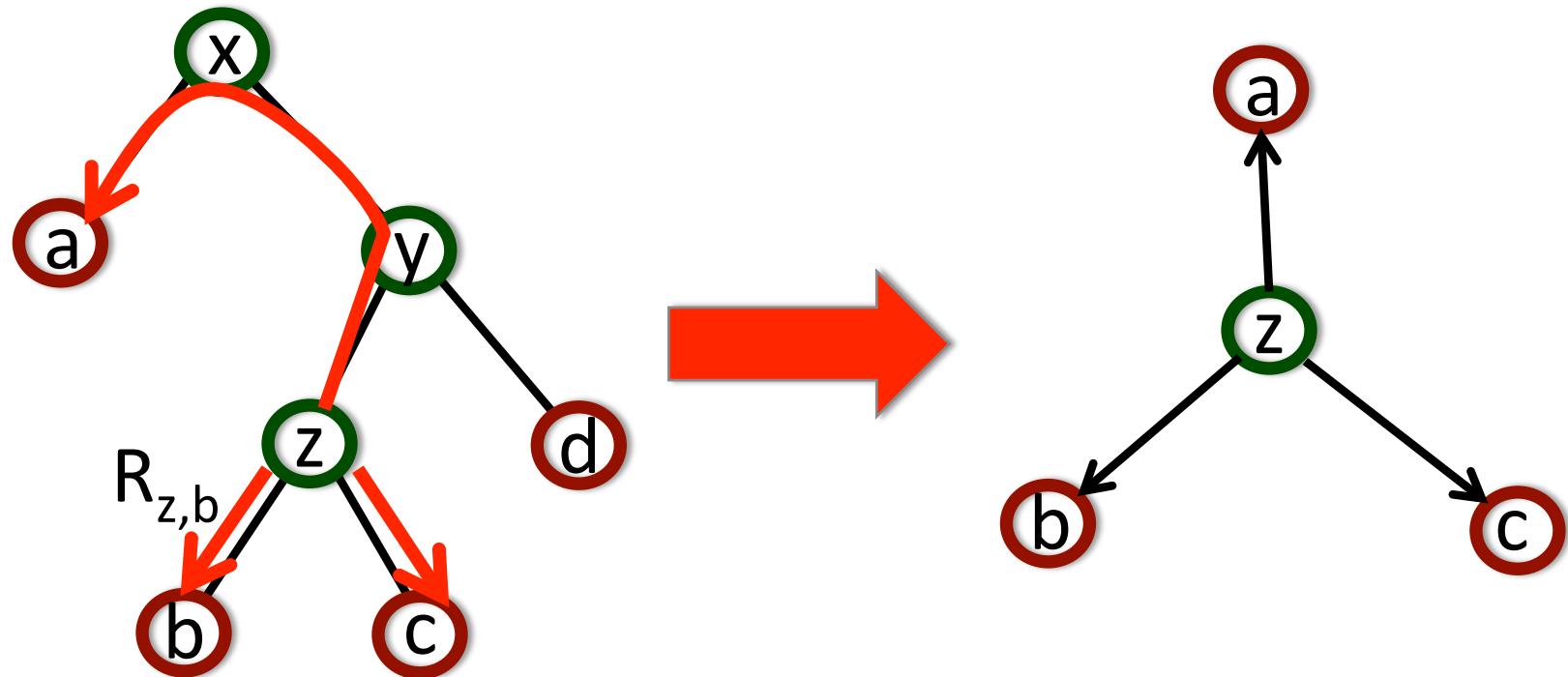
[Chang, 1996]: The model is identifiable (if R's are full rank)



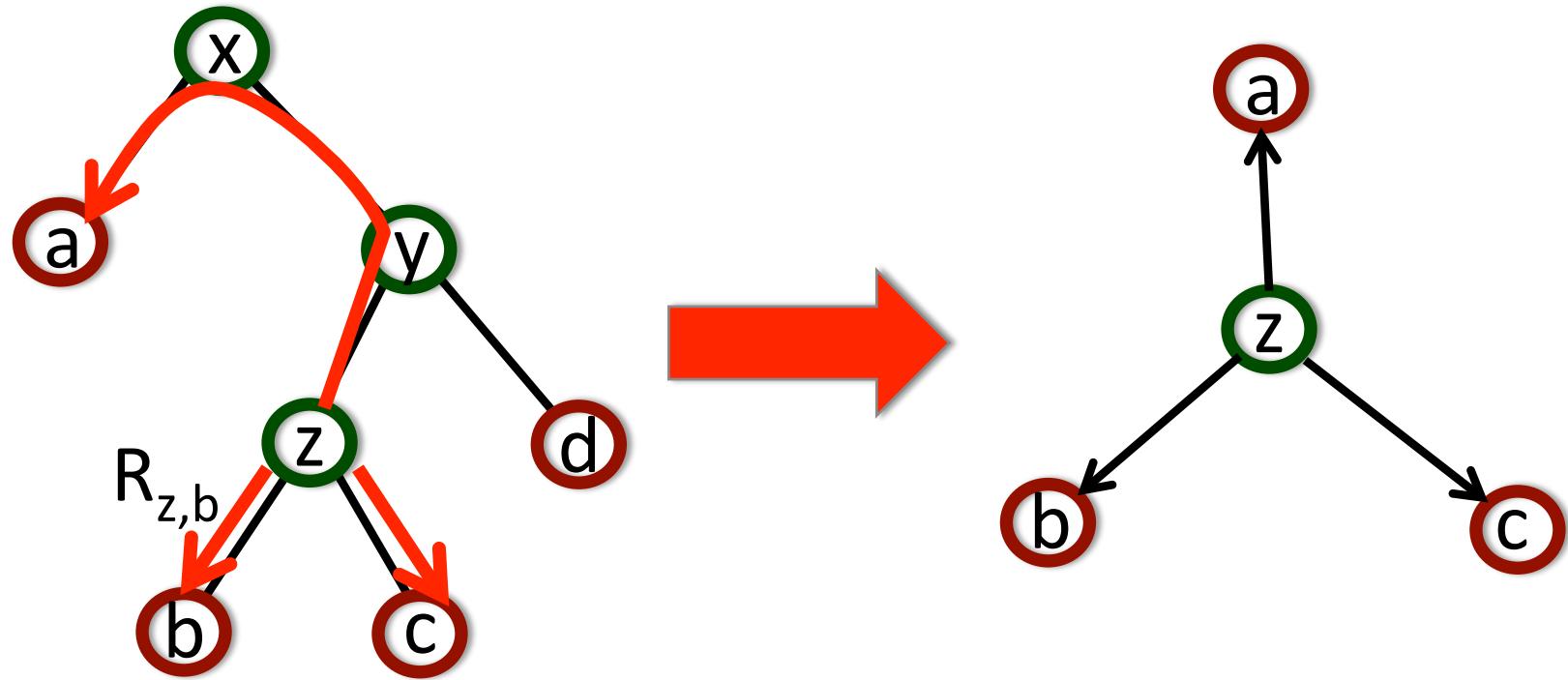
[Chang, 1996]: The model is identifiable (if R's are full rank)



[Chang, 1996]: The model is identifiable (if R's are full rank)



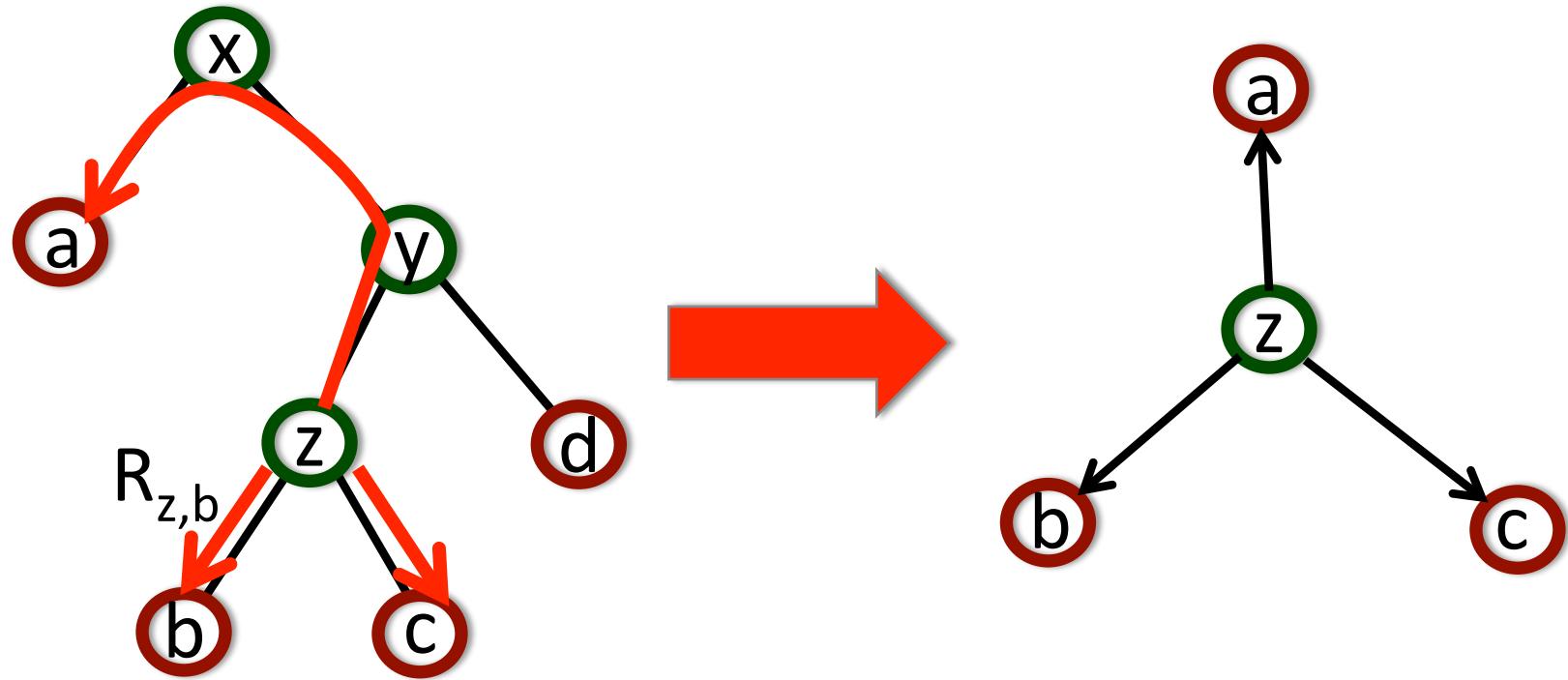
[Chang, 1996]: The model is identifiable (if R's are full rank)



Joint distribution over (a, b, c):

$$\sum_{\sigma} \Pr[z = \sigma] \Pr[a | z = \sigma] \otimes \Pr[b | z = \sigma] \otimes \Pr[c | z = \sigma]$$

[Chang, 1996]: The model is identifiable (if R's are full rank)



Joint distribution over (a, b, c):

$$\sum_{\sigma} \Pr[z = \sigma] \Pr[a | z = \sigma] \otimes \underbrace{\Pr[b | z = \sigma] \otimes \Pr[c | z = \sigma]}_{\text{columns of } R_{z,b}}$$

**[Mossel, Roch, 2006]:** There is an algorithm to PAC learn a phylogenetic tree or an HMM (if its transition/output matrices are full rank) from polynomially many samples

**[Mossel, Roch, 2006]:** There is an algorithm to PAC learn a phylogenetic tree or an HMM (if its transition/output matrices are full rank) from polynomially many samples

**Question:** Is the full-rank assumption necessary?

**[Mossel, Roch, 2006]:** There is an algorithm to PAC learn a phylogenetic tree or an HMM (if its transition/output matrices are full rank) from polynomially many samples

**Question:** Is the full-rank assumption necessary?

**[Mossel, Roch, 2006]:** It is as hard as noisy-parity to learn the parameters of a general HMM

**[Mossel, Roch, 2006]:** There is an algorithm to PAC learn a phylogenetic tree or an HMM (if its transition/output matrices are full rank) from polynomially many samples

**Question:** Is the full-rank assumption necessary?

**[Mossel, Roch, 2006]:** It is as hard as noisy-parity to learn the parameters of a general HMM

Noisy-parity is an infamous problem in learning, where  $O(n)$  samples suffice but the best algorithms run in time  $2^{n/\log(n)}$

Due to **[Blum, Kalai, Wasserman, 2003]**

**[Mossel, Roch, 2006]:** There is an algorithm to PAC learn a phylogenetic tree or an HMM (if its transition/output matrices are full rank) from polynomially many samples

**Question:** Is the full-rank assumption necessary?

**[Mossel, Roch, 2006]:** It is as hard as noisy-parity to learn the parameters of a general HMM

Noisy-parity is an infamous problem in learning, where  $O(n)$  samples suffice but the best algorithms run in time  $2^{n/\log(n)}$

Due to **[Blum, Kalai, Wasserman, 2003]**

---

**(It's now used as a hard problem to build cryptosystems!)**

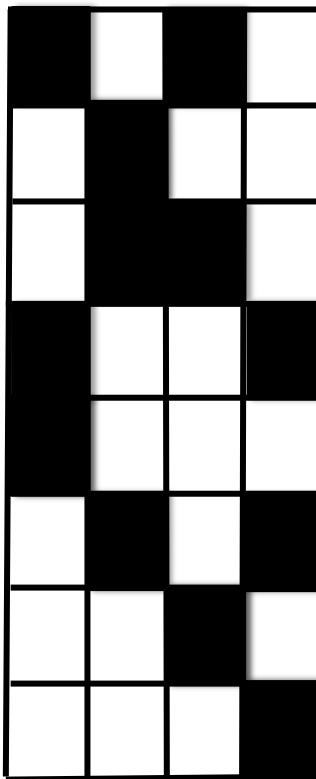
# THE POWER OF CONDITIONAL INDEPENDENCE

[Phylogenetic Trees/HMMS]: (joint distribution on leaves a, b, c)

$$\sum_{\sigma} \Pr[z = \sigma] \Pr[a | z = \sigma] \otimes \Pr[b | z = \sigma] \otimes \Pr[c | z = \sigma]$$

# PURE TOPIC MODELS

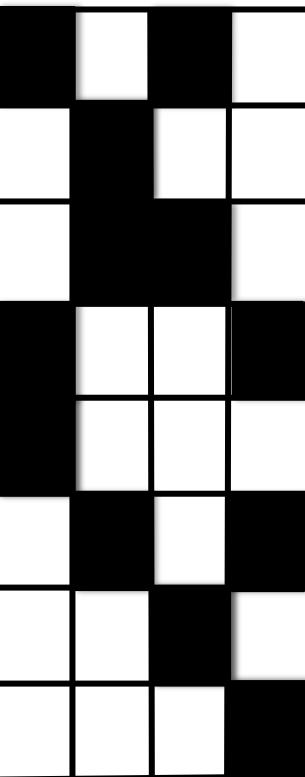
words ( $m$ )



- Each topic is a distribution on words

# PURE TOPIC MODELS

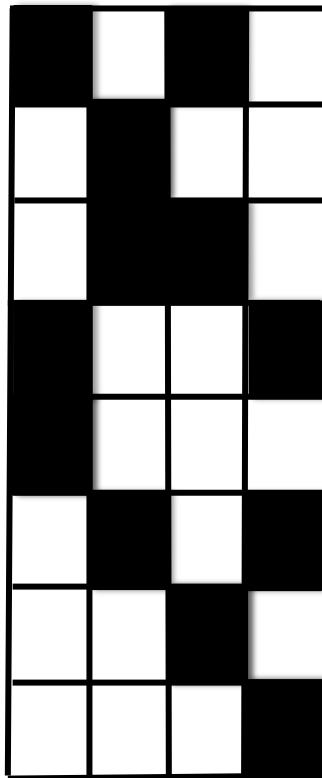
words (m)



- Each topic is a distribution on words
- **Each document is about only one topic**  
(stochastically generated)

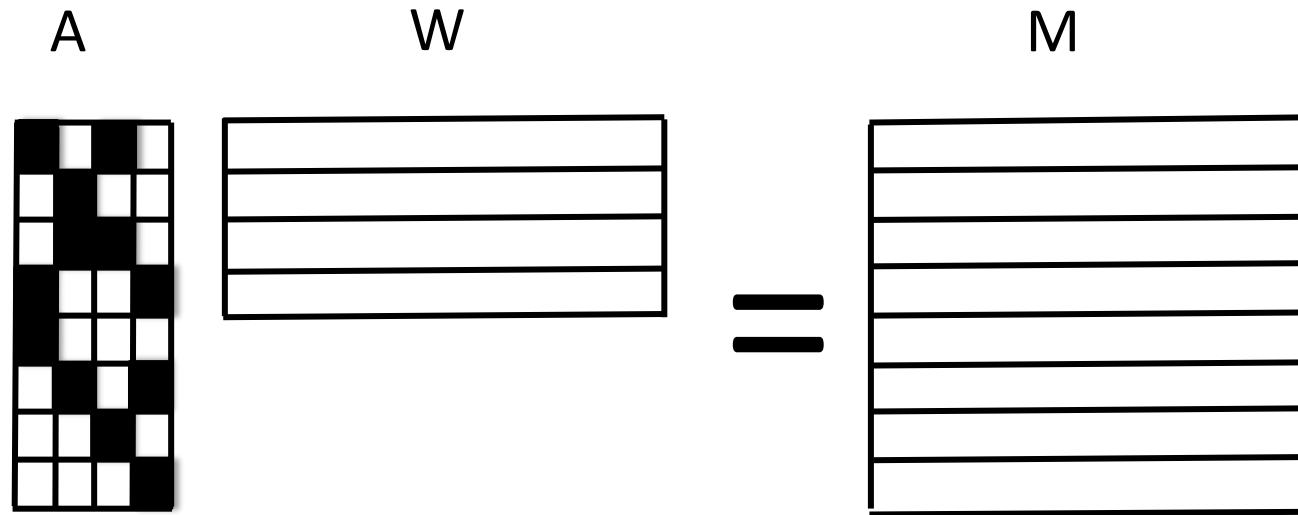
# PURE TOPIC MODELS

topics (r)  
words (m)

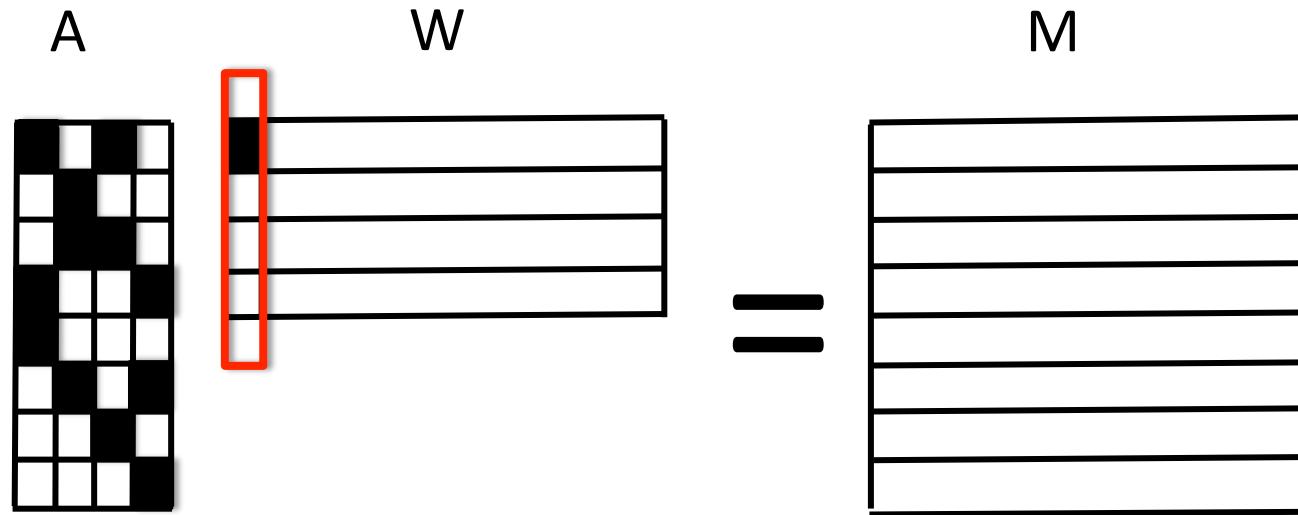


- Each topic is a distribution on words
- **Each document is about only one topic**  
(stochastically generated)
- Each document, we sample L words from its distribution

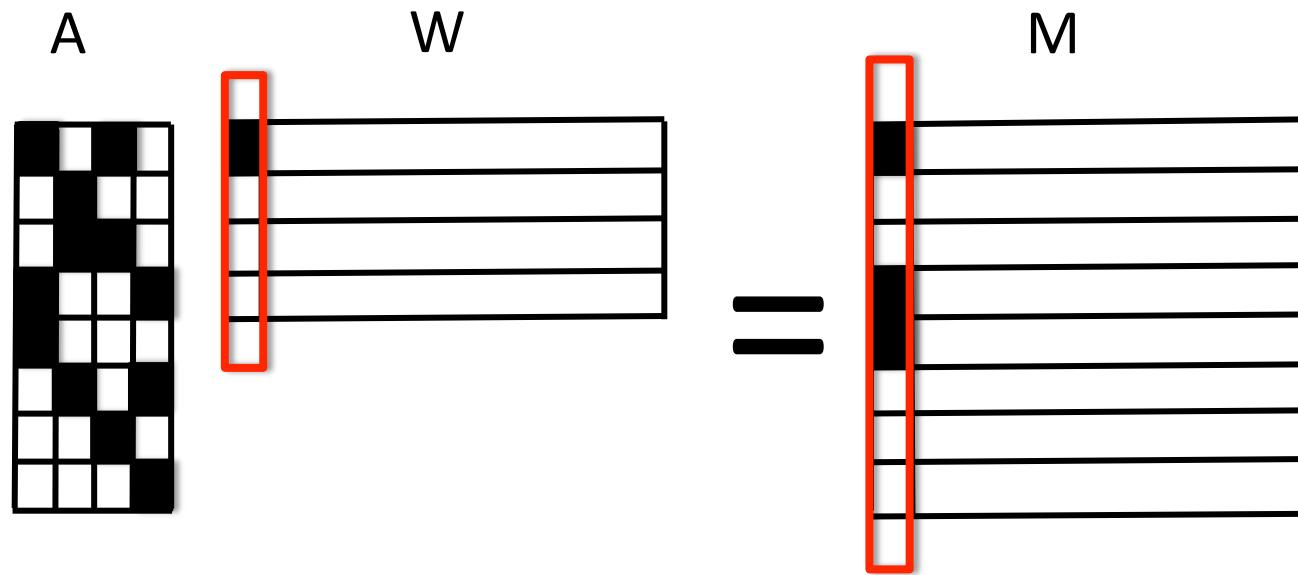
# PURE TOPIC MODELS



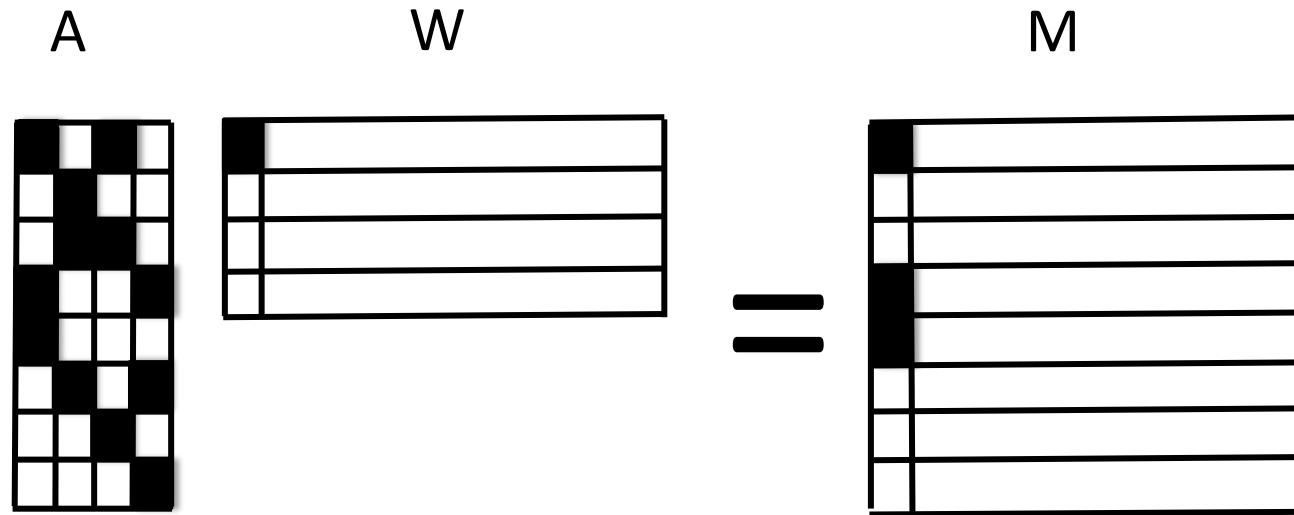
# PURE TOPIC MODELS



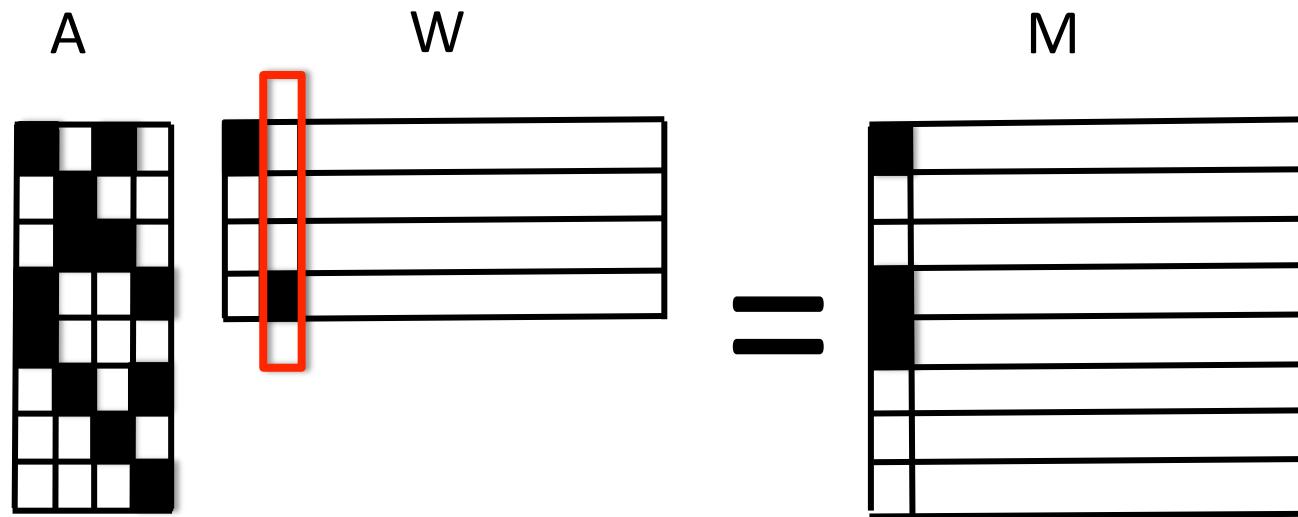
# PURE TOPIC MODELS



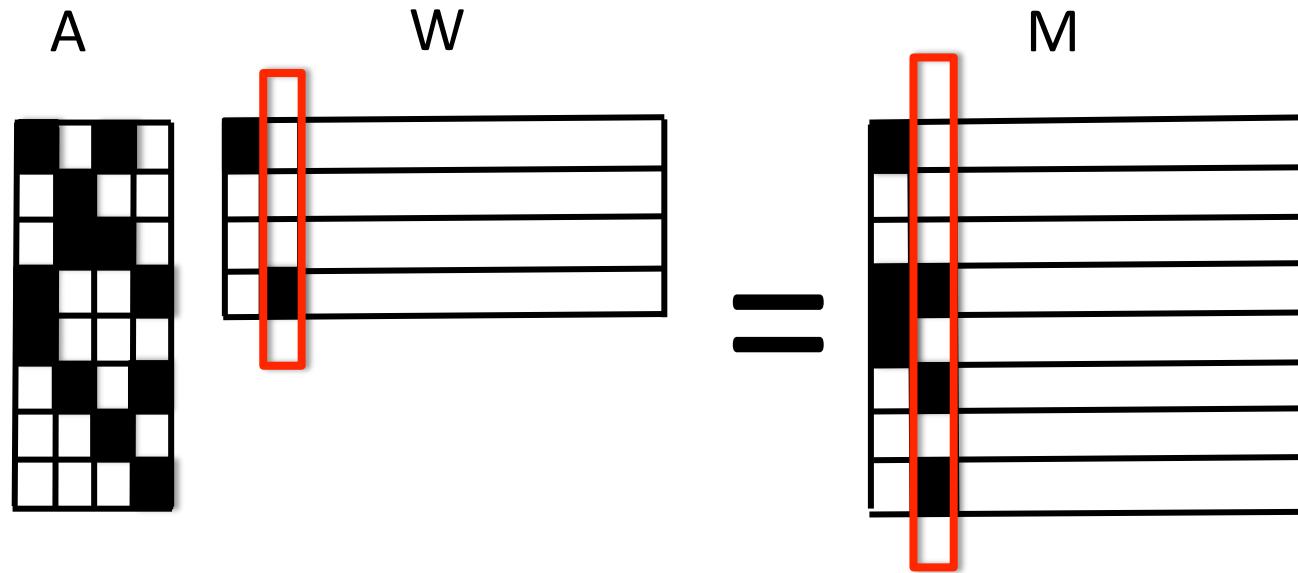
# PURE TOPIC MODELS



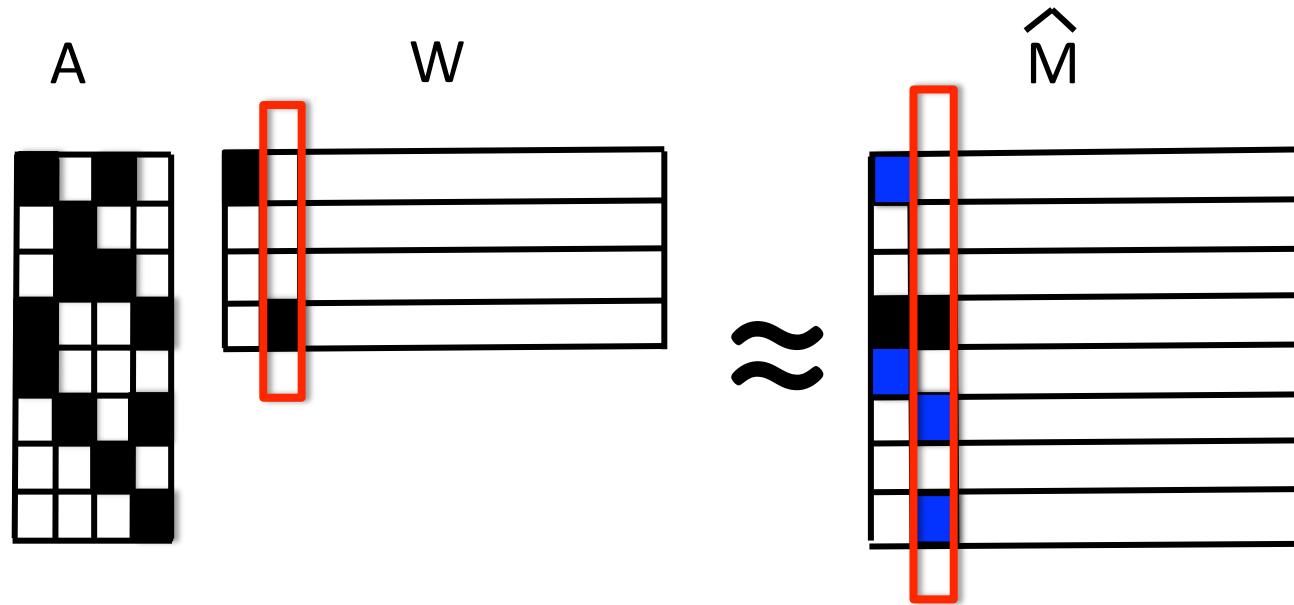
# PURE TOPIC MODELS



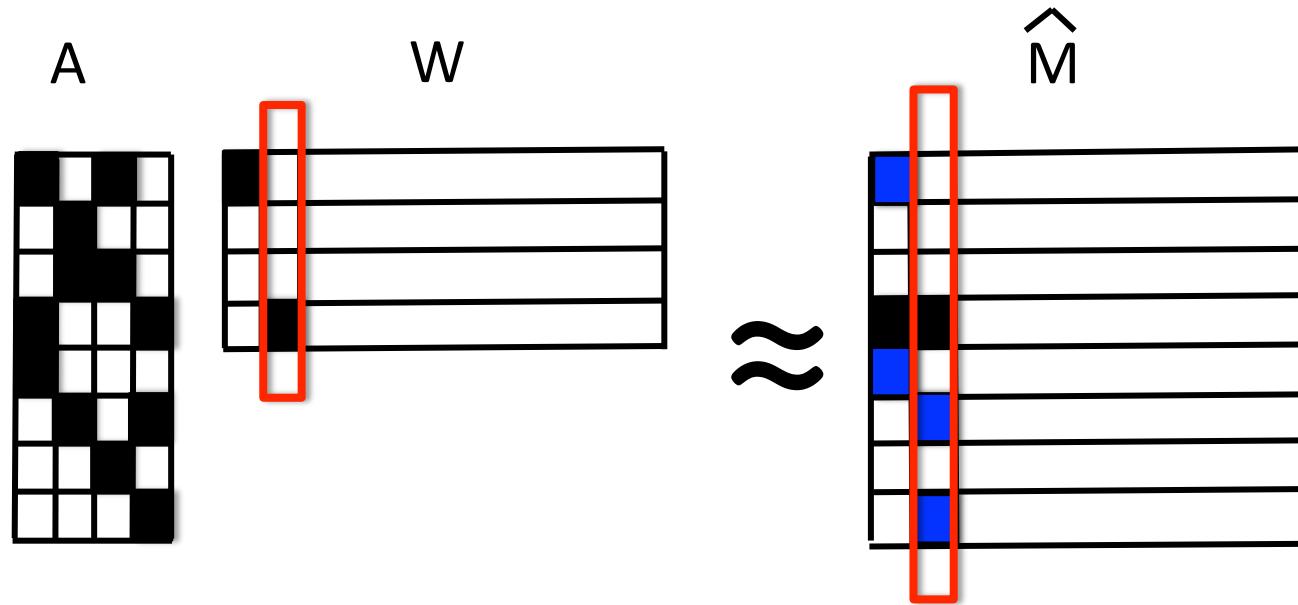
# PURE TOPIC MODELS



# PURE TOPIC MODELS

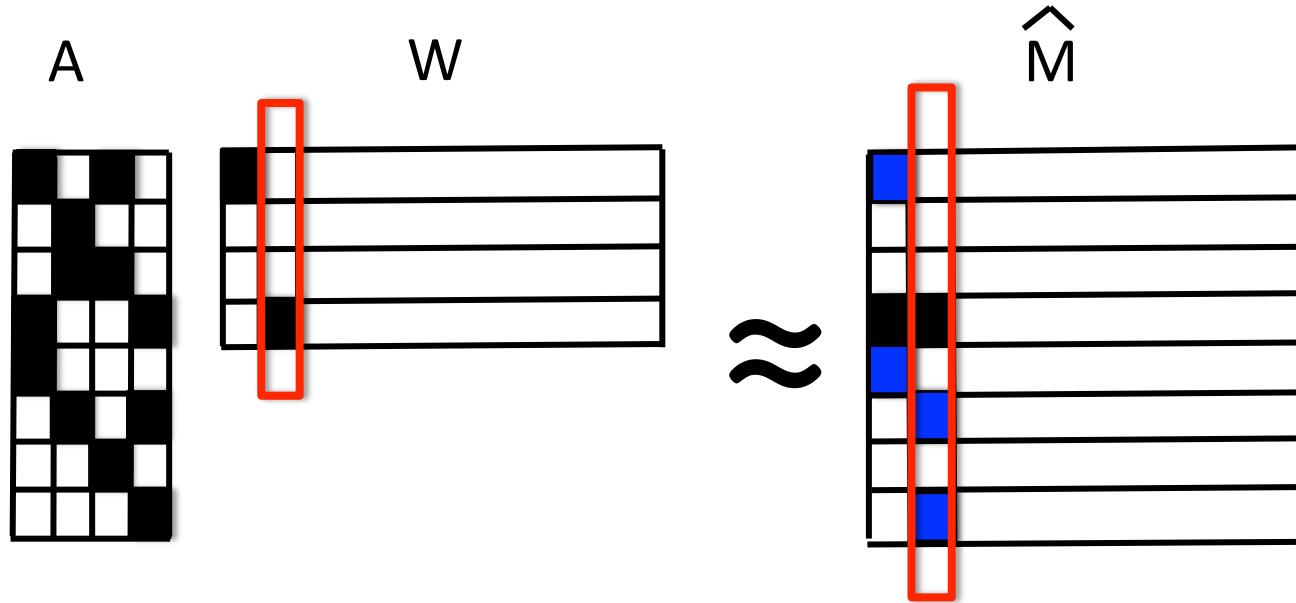


# PURE TOPIC MODELS



**[Anandkumar, Hsu, Kakade, 2012]:** Algorithm for learning pure topic models from polynomially many samples ( $A$  is full rank)

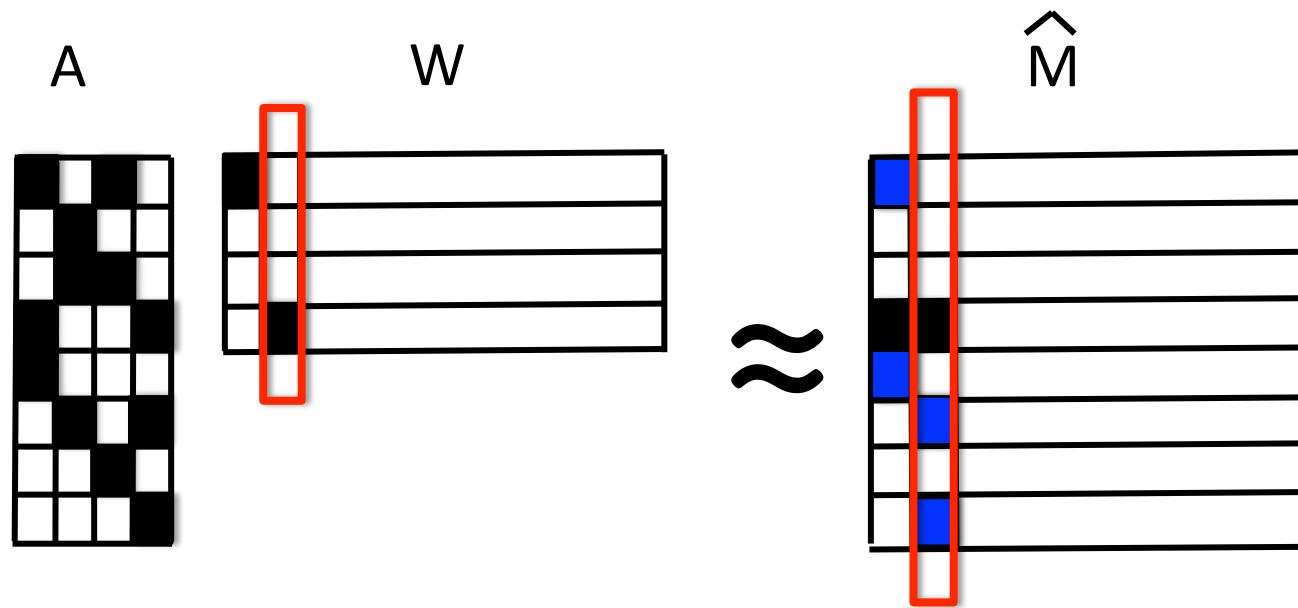
# PURE TOPIC MODELS



**[Anandkumar, Hsu, Kakade, 2012]:** Algorithm for learning pure topic models from polynomially many samples ( $A$  is full rank)

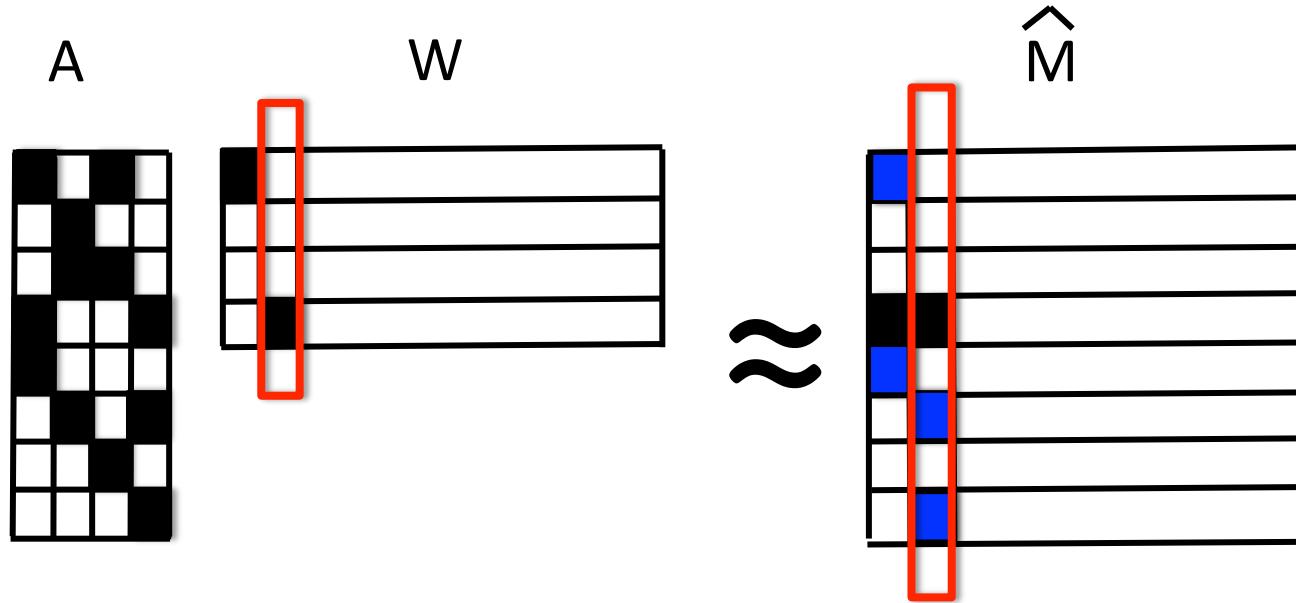
**Question:** Where can we find three conditionally independent random variables?

# PURE TOPIC MODELS



**[Anandkumar, Hsu, Kakade, 2012]:** Algorithm for learning pure topic models from polynomially many samples ( $A$  is full rank)

# PURE TOPIC MODELS



**[Anandkumar, Hsu, Kakade, 2012]:** Algorithm for learning pure topic models from polynomially many samples ( $A$  is full rank)

The first, second and third words are independent conditioned on the topic  $t$  (and are random samples from  $A_t$ )

# THE POWER OF CONDITIONAL INDEPENDENCE

[Phylogenetic Trees/HMMS]: (joint distribution on leaves a, b, c)

$$\sum_{\sigma} \Pr[z = \sigma] \Pr[a | z = \sigma] \otimes \Pr[b | z = \sigma] \otimes \Pr[c | z = \sigma]$$

# THE POWER OF CONDITIONAL INDEPENDENCE

[Phylogenetic Trees/HMMS]: (joint distribution on leaves a, b, c)

$$\sum_{\sigma} \Pr[z = \sigma] \Pr[a | z = \sigma] \otimes \Pr[b | z = \sigma] \otimes \Pr[c | z = \sigma]$$

[Pure Topic Models/LDA]: (joint distribution on first three words)

$$\sum_j \Pr[\text{topic} = j] A_j \otimes A_j \otimes A_j$$

# THE POWER OF CONDITIONAL INDEPENDENCE

[Phylogenetic Trees/HMMS]: (joint distribution on leaves a, b, c)

$$\sum_{\sigma} \Pr[z = \sigma] \Pr[a | z = \sigma] \otimes \Pr[b | z = \sigma] \otimes \Pr[c | z = \sigma]$$

[Pure Topic Models/LDA]: (joint distribution on first three words)

$$\sum_j \Pr[\text{topic} = j] A_j \otimes A_j \otimes A_j$$

[Community Detection]: (counting stars)

$$\sum_j \Pr[C_x = j] (C_A \Pi)_j \otimes (C_B \Pi)_j \otimes (C_C \Pi)_j$$

# OUTLINE

The focus of this tutorial is on Algorithms/Applications/Models for tensor decompositions

## Part I: Algorithms

- The Rotation Problem
- Jennrich's Algorithm

## Part II: Applications

- Phylogenetic Reconstruction
- Pure Topic Models

## Part III: Smoothed Analysis

- Overcomplete Problems
- Kruskal Rank and the Khatri-Rao Product

So far, Jennrich's algorithm has been the key but it has a crucial limitation.

So far, Jennrich's algorithm has been the key but it has a crucial limitation. Let

$$T = \sum_{i=1}^R a_i \otimes a_i \otimes a_i$$

where  $\{a_i\}$  are n-dimensional vectors

So far, Jennrich's algorithm has been the key but it has a crucial limitation. Let

$$T = \sum_{i=1}^R a_i \otimes a_i \otimes a_i$$

where  $\{a_i\}$  are n-dimensional vectors

**Question:** What if R is much larger than n?

So far, Jennrich's algorithm has been the key but it has a crucial limitation. Let

$$T = \sum_{i=1}^R a_i \otimes a_i \otimes a_i$$

where  $\{a_i\}$  are n-dimensional vectors

**Question:** What if R is much larger than n?

This is called the **overcomplete** case — e.g. the number of factors is much larger than the number of observations...

So far, Jennrich's algorithm has been the key but it has a crucial limitation. Let

$$T = \sum_{i=1}^R a_i \otimes a_i \otimes a_i$$

where  $\{a_i\}$  are n-dimensional vectors

**Question:** What if R is much larger than n?

This is called the **overcomplete** case — e.g. the number of factors is much larger than the number of observations...

**In such cases, why stop at third-order tensors?**

Consider a **sixth**-order tensor T:

$$T = \sum_{i=1}^R a_i \otimes a_i \otimes a_i \otimes a_i \otimes a_i \otimes a_i$$

Consider a **sixth**-order tensor T:

$$T = \sum_{i=1}^R a_i \otimes a_i \otimes a_i \otimes a_i \otimes a_i \otimes a_i$$

**Question:** Can we find its factors, even if R is much larger than n?

Consider a **sixth**-order tensor  $T$ :

$$T = \sum_{i=1}^R a_i \otimes a_i \otimes a_i \otimes a_i \otimes a_i \otimes a_i$$

**Question:** Can we find its factors, even if  $R$  is much larger than  $n$ ?

Let's flatten it:

$$\text{flat}(T) = \sum_{i=1}^R b_i \otimes b_i \otimes b_i \quad (\text{where } b_i = a_i \otimes_{KR} a_i)$$



$n^2$ -dimensional vector whose  $(j,k)^{\text{th}}$  entry is the product of the  $j^{\text{th}}$  and  $k^{\text{th}}$  entries of  $a_i$  — **Khatri-Rao product**

Consider a **sixth**-order tensor  $T$ :

$$T = \sum_{i=1}^R a_i \otimes a_i \otimes a_i \otimes a_i \otimes a_i \otimes a_i$$

**Question:** Can we find its factors, even if  $R$  is much larger than  $n$ ?

Let's flatten it by rearranging its entries into a **third**-order tensor:

$$\text{flat}(T) = \sum_{i=1}^R b_i \otimes b_i \otimes b_i \quad (\text{where } b_i = a_i \otimes_{KR} a_i)$$

$n^2$ -dimensional vector whose  $(j,k)^{\text{th}}$  entry is the product of the  $j^{\text{th}}$  and  $k^{\text{th}}$  entries of  $a_i$  — **Khatri-Rao product**

**Question:** Can we apply Jennrich's Algorithm to  $\text{flat}(T)$ ?

**Question:** Can we apply Jennrich's Algorithm to flat( $T$ )?

When are the new factors  $b_i = a_i \bigotimes_{KR} a_i$  linearly independent?

**Question:** Can we apply Jennrich's Algorithm to flat( $T$ )?

When are the new factors  $b_i = a_i \bigotimes_{KR} a_i$  linearly independent?

### Example #1:

Let  $\{a_i\}$  be all  $\binom{n}{2}$  vectors with exactly two ones

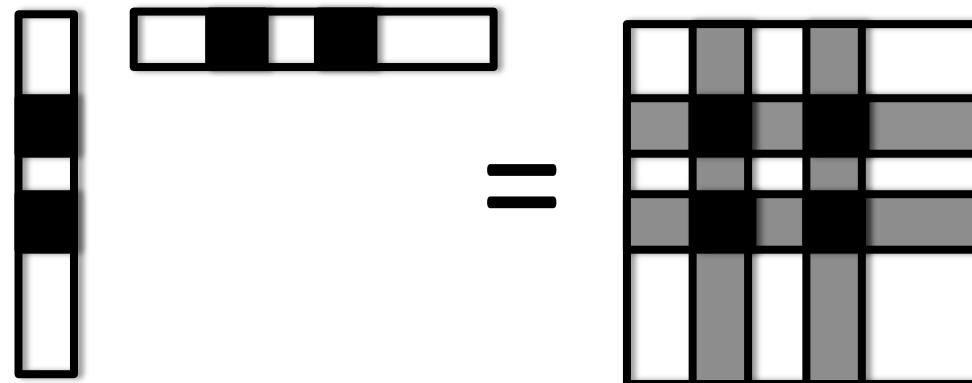
**Question:** Can we apply Jennrich's Algorithm to  $\text{flat}(T)$ ?

When are the new factors  $b_i = a_i \otimes_{KR} a_i$  linearly independent?

## Example #1:

Let  $\{a_i\}$  be all  $\binom{n}{2}$  vectors with exactly two ones

Then  $\{b_i\}$  are vectorizations of:



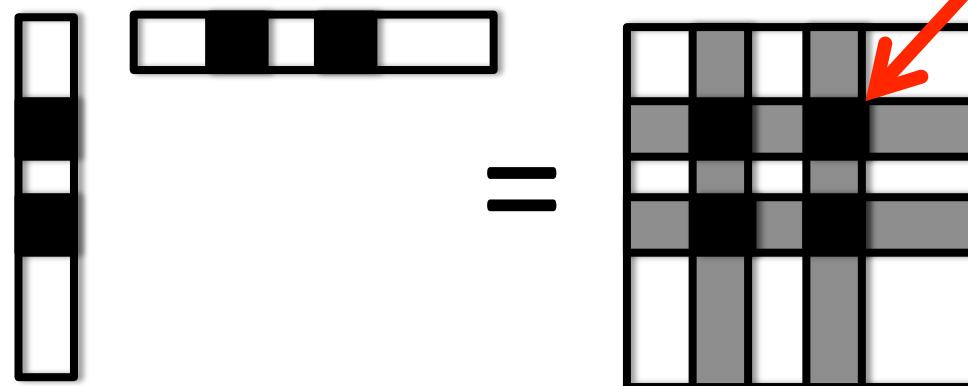
**Question:** Can we apply Jennrich's Algorithm to  $\text{flat}(T)$ ?

When are the new factors  $b_i = a_i \otimes_{\text{KR}} a_i$  linearly independent?

### Example #1:

Let  $\{a_i\}$  be all  $\binom{n}{2}$  vectors with exactly two ones

Then  $\{b_i\}$  are vectorizations of:



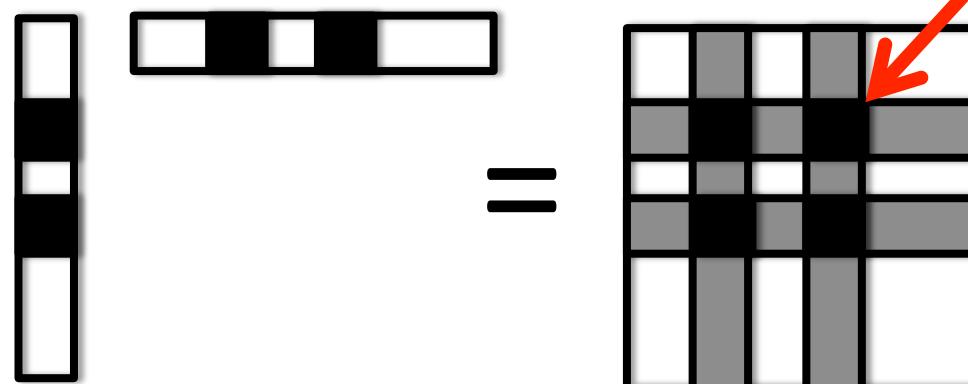
**Question:** Can we apply Jennrich's Algorithm to  $\text{flat}(T)$ ?

When are the new factors  $b_i = a_i \otimes_{\text{KR}} a_i$  linearly independent?

### Example #1:

Let  $\{a_i\}$  be all  $\binom{n}{2}$  vectors with exactly two ones

Then  $\{b_i\}$  are vectorizations of:



Non-zero  
only in  $b_i$

and are linearly independent

**Question:** Can we apply Jennrich's Algorithm to flat( $T$ )?

When are the new factors  $b_i = a_i \bigotimes_{KR} a_i$  linearly independent?

**Question:** Can we apply Jennrich's Algorithm to  $\text{flat}(T)$ ?

When are the new factors  $b_i = a_i \bigotimes_{KR} a_i$  linearly independent?

## Example #2:

Let  $\{a_{1\dots n}\}$  and  $\{a_{n+1..2n}\}$  be two random orthonormal bases

**Question:** Can we apply Jennrich's Algorithm to  $\text{flat}(T)$ ?

When are the new factors  $b_i = a_i \bigotimes_{KR} a_i$  linearly independent?

## Example #2:

Let  $\{a_{1\dots n}\}$  and  $\{a_{n+1..2n}\}$  be two random orthonormal bases

Then there is a linear dependence with  $2n$  terms:

**Question:** Can we apply Jennrich's Algorithm to  $\text{flat}(T)$ ?

When are the new factors  $b_i = a_i \otimes_{KR} a_i$  linearly independent?

## Example #2:

Let  $\{a_{1\dots n}\}$  and  $\{a_{n+1\dots 2n}\}$  be two random orthonormal bases

Then there is a linear dependence with  $2n$  terms:

$$\sum_{i=1}^n a_i \otimes_{KR} a_i - \sum_{i=n+1}^{2n} a_i \otimes_{KR} a_i = 0$$

**Question:** Can we apply Jennrich's Algorithm to  $\text{flat}(T)$ ?

When are the new factors  $b_i = a_i \otimes_{KR} a_i$  linearly independent?

## Example #2:

Let  $\{a_{1\dots n}\}$  and  $\{a_{n+1\dots 2n}\}$  be two random orthonormal bases

Then there is a linear dependence with  $2n$  terms:

$$\sum_{i=1}^n a_i \otimes_{KR} a_i - \sum_{i=n+1}^{2n} a_i \otimes_{KR} a_i = 0$$

(as matrices, both sum to the identity)

# THE KRUSKAL RANK

# THE KRUSKAL RANK

**Definition:** The **Kruskal rank** ( $k$ -rank) of  $\{b_i\}$  is the largest  $k$  s.t. every set of  $k$  vectors is linearly independent

# THE KRUSKAL RANK

**Definition:** The **Kruskal rank** (k-rank) of  $\{b_i\}$  is the largest k s.t. every set of k vectors is linearly independent

$$b_i = a_i \bigotimes_{KR} a_i \quad k\text{-rank}(\{a_i\}) = n$$

# THE KRUSKAL RANK

**Definition:** The **Kruskal rank** ( $k$ -rank) of  $\{b_i\}$  is the largest  $k$  s.t. every set of  $k$  vectors is linearly independent

$$b_i = a_i \bigotimes_{KR} a_i \quad k\text{-rank}(\{a_i\}) = n$$

**Example #1:**  $k\text{-rank}(\{b_i\}) = R = \binom{n}{2}$

# THE KRUSKAL RANK

**Definition:** The **Kruskal rank** ( $k$ -rank) of  $\{b_i\}$  is the largest  $k$  s.t. every set of  $k$  vectors is linearly independent

$$b_i = a_i \bigotimes_{KR} a_i \quad k\text{-rank}(\{a_i\}) = n$$

**Example #1:**  $k\text{-rank}(\{b_i\}) = R = \binom{n}{2}$

**Example #2:**  $k\text{-rank}(\{b_i\}) = 2n-1$

# THE KRUSKAL RANK

**Definition:** The **Kruskal rank** ( $k$ -rank) of  $\{b_i\}$  is the largest  $k$  s.t. every set of  $k$  vectors is linearly independent

$$b_i = a_i \bigotimes_{KR} a_i \quad k\text{-rank}(\{a_i\}) = n$$

**Example #1:**  $k\text{-rank}(\{b_i\}) = R = \binom{n}{2}$

**Example #2:**  $k\text{-rank}(\{b_i\}) = 2n-1$

---

The Kruskal rank always **adds** under the Khatri-Rao product, but sometimes it **multiples** and that can allow us to handle  $R \gg n$

**[Allman, Matias, Rhodes, 2009]:** Almost surely, the Kruskal rank multiplies under the Khatri-Rao product

**[Allman, Matias, Rhodes, 2009]:** Almost surely, the Kruskal rank multiplies under the Khatri-Rao product

**Proof:** The set of  $\{a_i\}$  where

$$b_i = a_i \bigotimes_{KR} a_i \quad \text{and} \quad \det(\{b_i\}) = 0$$

is measure zero ■

**[Allman, Matias, Rhodes, 2009]:** Almost surely, the Kruskal rank multiplies under the Khatri-Rao product

**Proof:** The set of  $\{a_i\}$  where

$$b_i = a_i \bigotimes_{KR} a_i \quad \text{and} \quad \det(\{b_i\}) = 0$$

is measure zero ■

---

But this yields a very weak bound on the **condition number** of  $\{b_i\}$ ...

**[Allman, Matias, Rhodes, 2009]:** Almost surely, the Kruskal rank multiplies under the Khatri-Rao product

**Proof:** The set of  $\{a_i\}$  where

$$b_i = a_i \bigotimes_{KR} a_i \quad \text{and} \quad \det(\{b_i\}) = 0$$

is measure zero ■

---

But this yields a very weak bound on the **condition number** of  $\{b_i\}$ ...

... which is what we need to apply it to learning/statistics, where we have an estimate to T

**[Allman, Matias, Rhodes, 2009]:** Almost surely, the Kruskal rank multiplies under the Khatri-Rao product

**[Allman, Matias, Rhodes, 2009]:** Almost surely, the Kruskal rank multiplies under the Khatri-Rao product

**Definition:** The **robust Kruskal rank** ( $k\text{-rank}_\gamma$ ) of  $\{b_i\}$  is the largest  $k$  s.t. every set of  $k$  vectors has condition number at most  $O(\gamma)$

**[Allman, Matias, Rhodes, 2009]:** Almost surely, the Kruskal rank multiplies under the Khatri-Rao product

**Definition:** The **robust Kruskal rank** ( $k\text{-rank}_\gamma$ ) of  $\{b_i\}$  is the largest  $k$  s.t. every set of  $k$  vectors has condition number at most  $O(\gamma)$

**[Bhaskara, Charikar, Vijayaraghavan, 2013]:** The robust Kruskal rank always under the Khatri-Rao product

**[Allman, Matias, Rhodes, 2009]:** Almost surely, the Kruskal rank multiplies under the Khatri-Rao product

**Definition:** The **robust Kruskal rank** ( $k\text{-rank}_\gamma$ ) of  $\{b_i\}$  is the largest  $k$  s.t. every set of  $k$  vectors has condition number at most  $O(\gamma)$

**[Bhaskara, Charikar, Vijayaraghavan, 2013]:** The robust Kruskal rank always under the Khatri-Rao product

**[Bhaskara, Charikar, Moitra, Vijayaraghavan, 2014]:** Suppose the vectors  $\{a_i\}$  are  $\varepsilon$ -perturbed...

**[Allman, Matias, Rhodes, 2009]:** Almost surely, the Kruskal rank multiplies under the Khatri-Rao product

**Definition:** The **robust Kruskal rank** ( $k\text{-rank}_\gamma$ ) of  $\{b_i\}$  is the largest  $k$  s.t. every set of  $k$  vectors has condition number at most  $O(\gamma)$

**[Bhaskara, Charikar, Vijayaraghavan, 2013]:** The robust Kruskal rank always under the Khatri-Rao product

**[Bhaskara, Charikar, Moitra, Vijayaraghavan, 2014]:** Suppose the vectors  $\{a_i\}$  are  $\varepsilon$ -perturbed. Then

$$k\text{-rank}_\gamma(\{b_i\}) = R$$

for  $R = n^2/2$  and  $\gamma = \text{poly}(1/n, \varepsilon)$  with **exponentially** small failure probability ( $\delta$ )

**[Bhaskara, Charikar, Moitra, Vijayaraghavan, 2014]:** Suppose the vectors  $\{a_i\}$  are  $\varepsilon$ -perturbed. Then

$$k\text{-rank}_\gamma(\{b_i\}) = R$$

for  $R = n^2/2$  and  $\gamma = \text{poly}(1/n, \varepsilon)$  with **exponentially** small failure probability ( $\delta$ )

**[Bhaskara, Charikar, Moitra, Vijayaraghavan, 2014]:** Suppose the vectors  $\{a_i\}$  are  $\varepsilon$ -perturbed. Then

$$k\text{-rank}_\gamma(\{b_i\}) = R$$

for  $R = n^2/2$  and  $\gamma = \text{poly}(1/n, \varepsilon)$  with **exponentially** small failure probability ( $\delta$ )

Hence we can apply Jennrich's Algorithm to  $\text{flat}(T)$  with  $R \gg n$

**[Bhaskara, Charikar, Moitra, Vijayaraghavan, 2014]:** Suppose the vectors  $\{a_i\}$  are  $\varepsilon$ -perturbed. Then

$$k\text{-rank}_\gamma(\{b_i\}) = R$$

for  $R = n^2/2$  and  $\gamma = \text{poly}(1/n, \varepsilon)$  with **exponentially** small failure probability ( $\delta$ )

Hence we can apply Jennrich's Algorithm to  $\text{flat}(T)$  with  $R \gg n$

**Note:** These bounds are easy to prove with inverse **polynomial** failure probability, but then  $\gamma$  depends  $\delta$

**[Bhaskara, Charikar, Moitra, Vijayaraghavan, 2014]:** Suppose the vectors  $\{a_i\}$  are  $\varepsilon$ -perturbed. Then

$$k\text{-rank}_\gamma(\{b_i\}) = R$$

for  $R = n^2/2$  and  $\gamma = \text{poly}(1/n, \varepsilon)$  with **exponentially** small failure probability ( $\delta$ )

Hence we can apply Jennrich's Algorithm to  $\text{flat}(T)$  with  $R \gg n$

**Note:** These bounds are easy to prove with inverse **polynomial** failure probability, but then  $\gamma$  depends  $\delta$

This can be extended to any constant order Khatri-Rao product

**[Bhaskara, Charikar, Moitra, Vijayaraghavan, 2014]:** Suppose the vectors  $\{a_i\}$  are  $\varepsilon$ -perturbed. Then

$$k\text{-rank}_\gamma(\{b_i\}) = R$$

for  $R = n^2/2$  and  $\gamma = \text{poly}(1/n, \varepsilon)$  with **exponentially** small failure probability ( $\delta$ )

Hence we can apply Jennrich's Algorithm to  $\text{flat}(T)$  with  $R \gg n$

**[Bhaskara, Charikar, Moitra, Vijayaraghavan, 2014]:** Suppose the vectors  $\{a_i\}$  are  $\varepsilon$ -perturbed. Then

$$k\text{-rank}_\gamma(\{b_i\}) = R$$

for  $R = n^2/2$  and  $\gamma = \text{poly}(1/n, \varepsilon)$  with **exponentially** small failure probability ( $\delta$ )

Hence we can apply Jennrich's Algorithm to  $\text{flat}(T)$  with  $R \gg n$

---

**Sample application:** Algorithm for learning mixtures of  $n^{O(1)}$  spherical Gaussians in  $R^n$ , if their means are  $\varepsilon$ -perturbed

This was also obtained independently by **[Anderson, Belkin, Goyal, Rademacher, Voss, 2014]**

# Any Questions?

## Summary:

- Tensor decompositions are **unique** under much more general conditions, compared to matrix decompositions
- Jennrich's Algorithm (rediscovered many times!), and its many applications in learning/statistics
- Introduced **new models** to study overcomplete problems ( $R \gg n$ )
- Are there algorithms for order- $k$  tensors that work with  $R = n^{0.51 k}$ ?