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

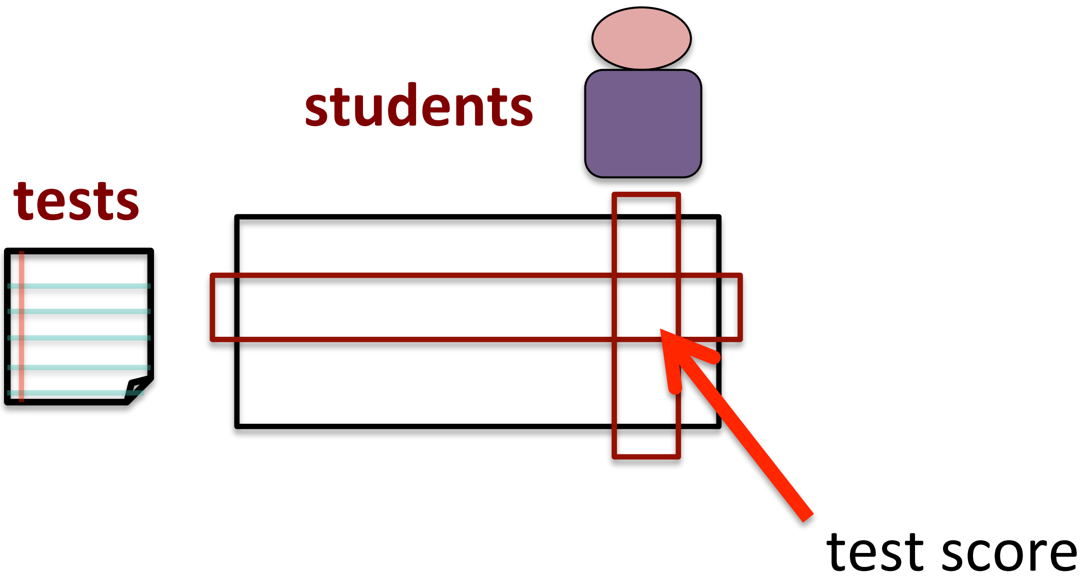
Charles Spearman (1904): There are two types of intelligence, *eductive* and *reproductive*

eductive (adj): the ability to make sense out of complexity
reproductive (adj): the ability to store and reproduce information

He devised the following experiment to test his theory…

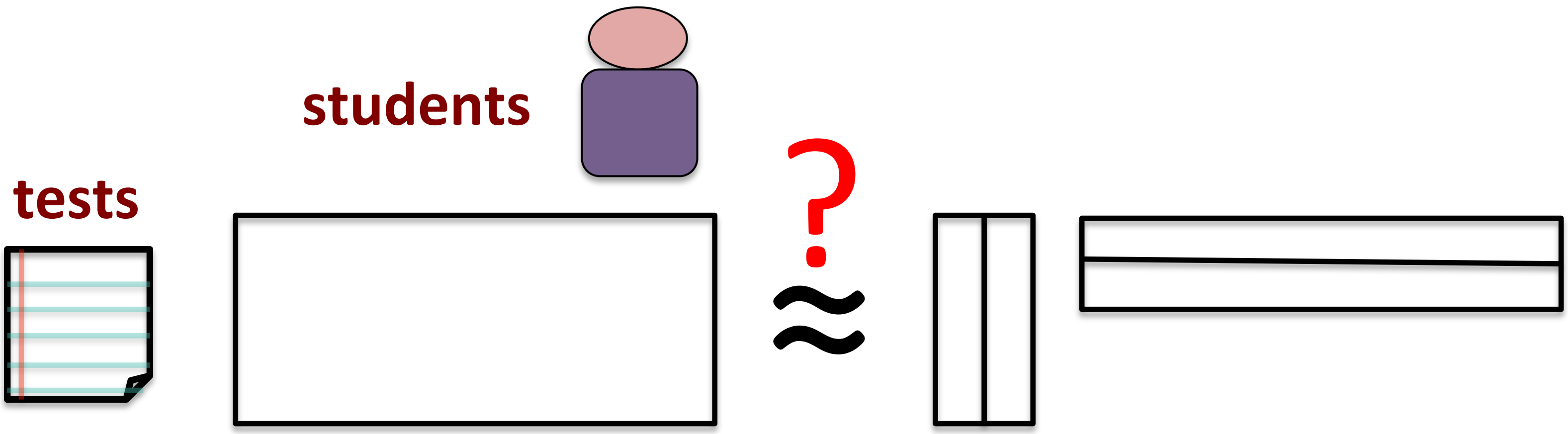
Spearman’s Hypothesis

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



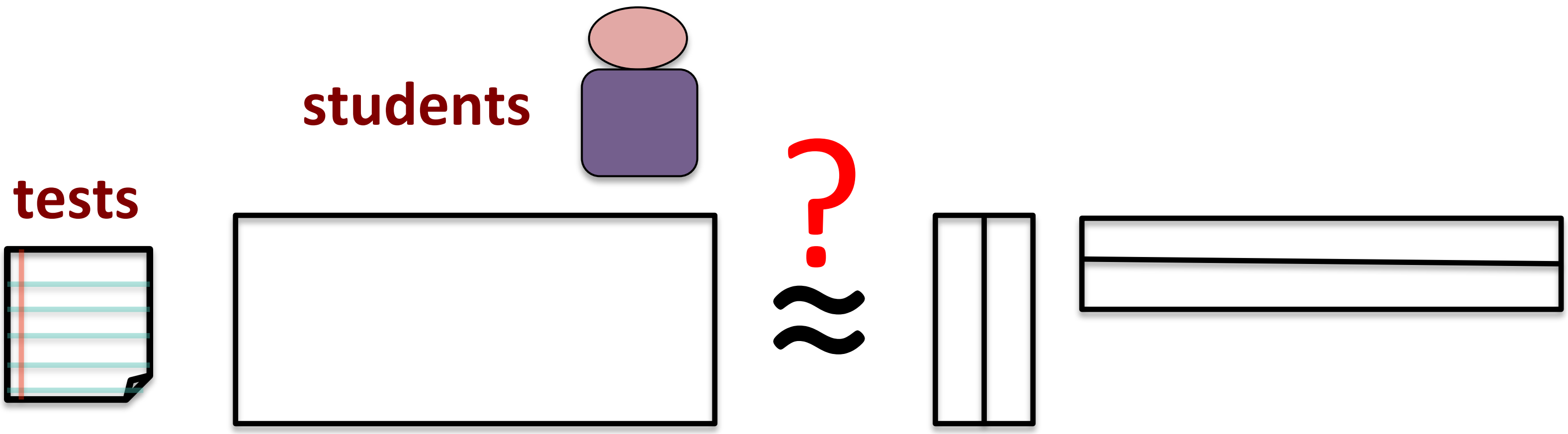
Spearman's Hypothesis

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



Spearman's Hypothesis

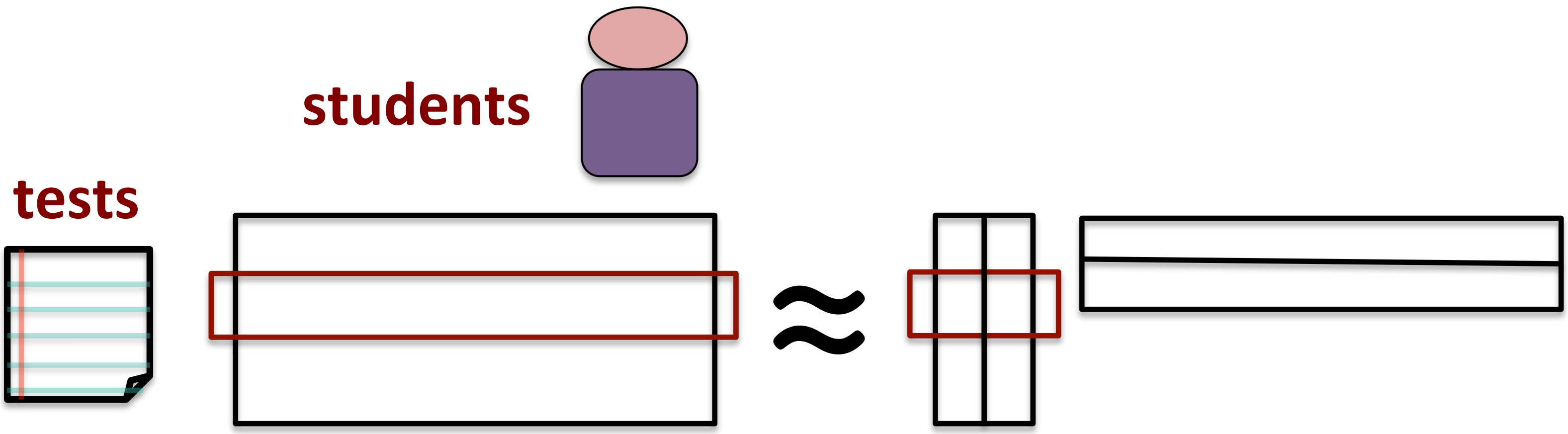
Charles Spearman (1904): There are two types of intelligence, *eductive* and *reproductive*



Hope: There is an **interpretable**, low-rank approximation

Spearman’s Hypothesis

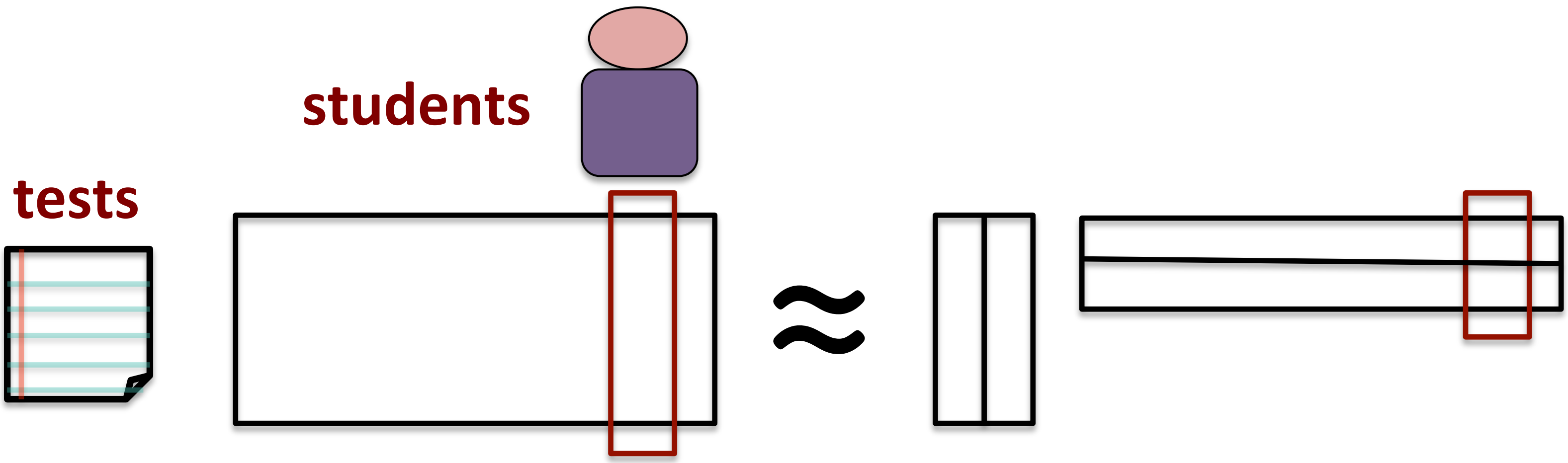
Charles Spearman (1904): There are two types of intelligence, *eductive* and *reproductive*



How much does it **test** { *eductive* / *reproductive* } reasoning?

Spearman's Hypothesis

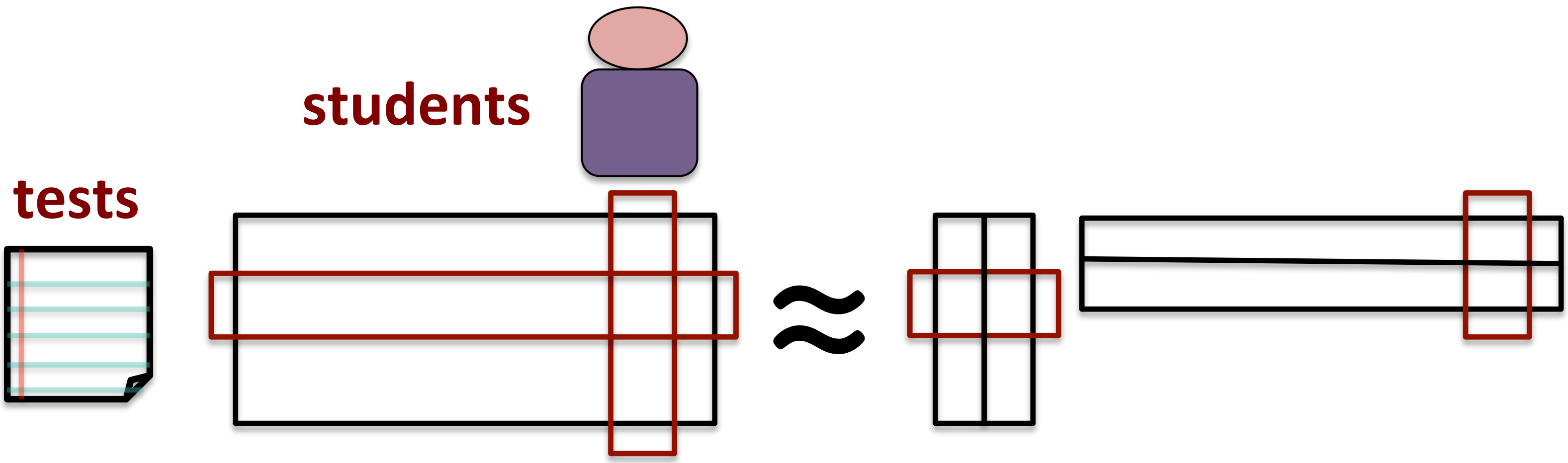
Charles Spearman (1904): There are two types of intelligence, *eductive* and *reproductive*



Student's **aptitude** for {*eductive* / *reproductive*} reasoning?

Spearman's Hypothesis

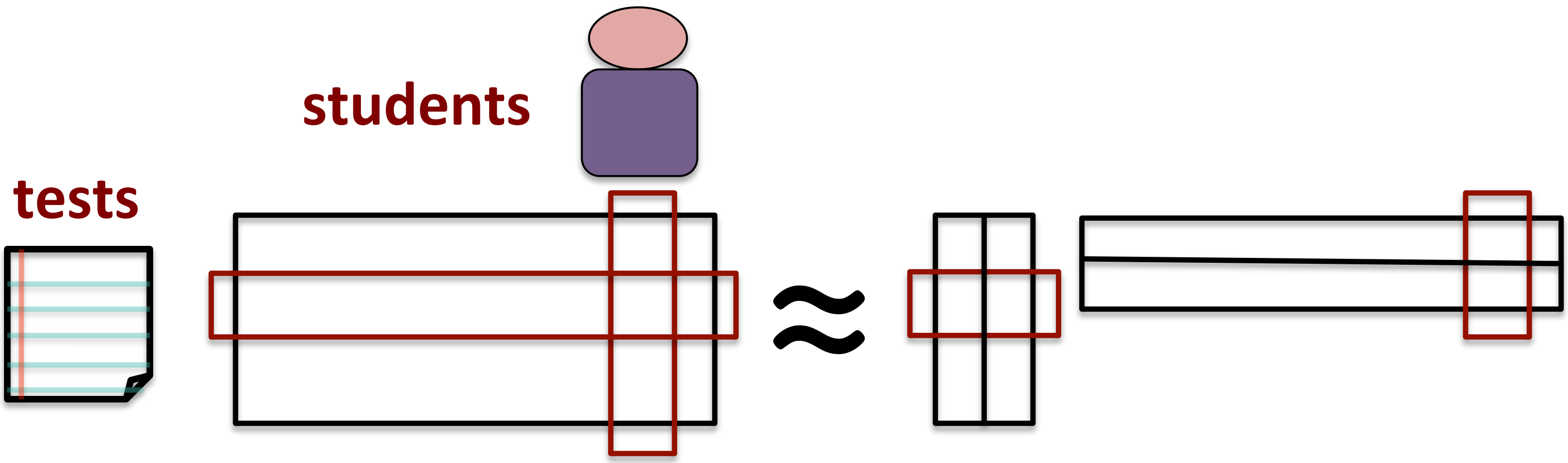
Charles Spearman (1904): There are two types of intelligence, *eductive* and *reproductive*



Factor analysis: Explain away observations using fewer latent (unobserved) variables

Spearman's Hypothesis

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



Factor analysis: Explain away observations using fewer latent (unobserved) variables

If it exists, how can we find an interpretable factorization?

The Rotation Problem

The Rotation Problem

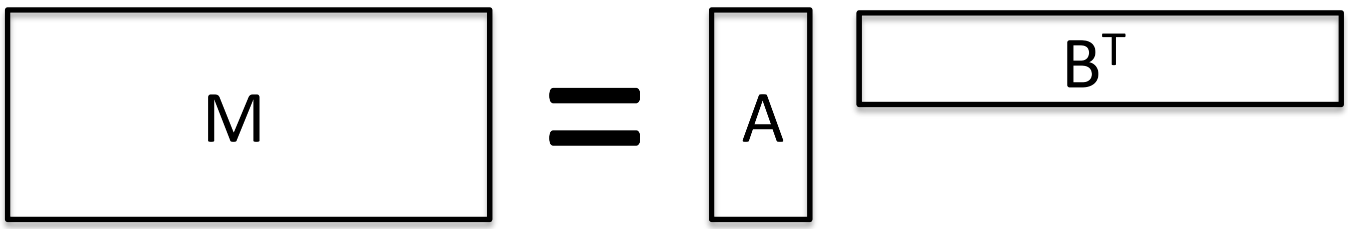
If there is a **true** factorization:

$$M = A \; B^T$$

(given) (hidden)

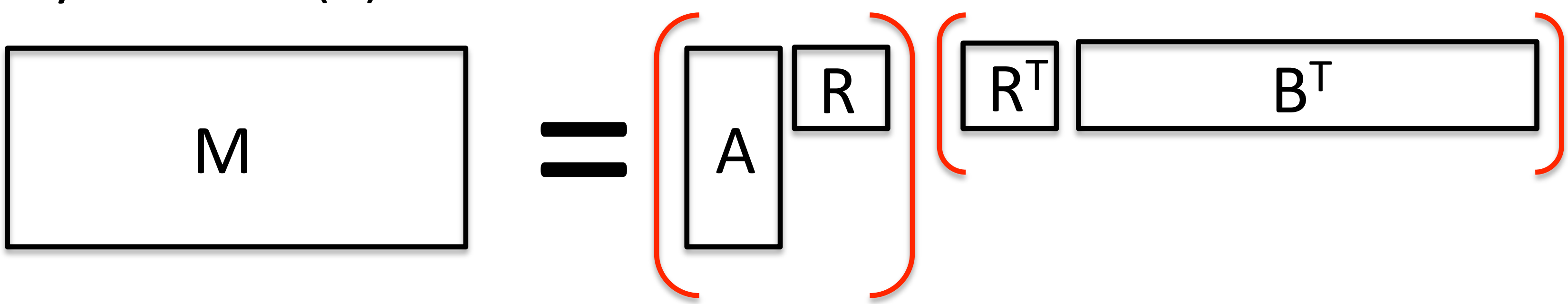
The Rotation Problem

If there is a **true** factorization:



(given) (hidden)

any **rotation** (R) of it is valid too



The Rotation Problem

Alternatively if there is a **true** factorization:

$$M = \sum_{i=1}^{R} a^{(i)} b^{(i)\,T}$$

The Rotation Problem

Alternatively if there is a **true** factorization:

$$M = \sum_{i=1}^{R} a^{(i)} b^{(i)\,T}$$

it cannot be uniquely determined from just M

(without extra conditions on $a^{(i)}$, $b^{(i)}$)

The Rotation Problem

Alternatively if there is a **true** factorization:

$$M = \sum_{i=1}^{R} a^{(i)} b^{(i)\,T}$$

it cannot be uniquely determined from just M

(without extra conditions on $a^{(i)}$, $b^{(i)}$)

Low-rank tensor decompositions are unique in ways that matrix decompositions are not!

Outline

The focus of this tutorial is on algorithms & applications

Part I: Tensor Decompositions

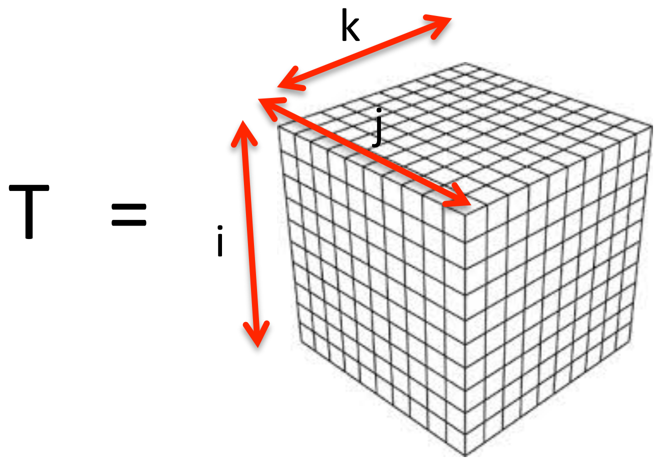
- The Rotation Problem
- A Primer on Tensors
- Jennrich's Algorithm

Part II: Applications

- Phylogenetic Reconstruction
- Pure Topic Models

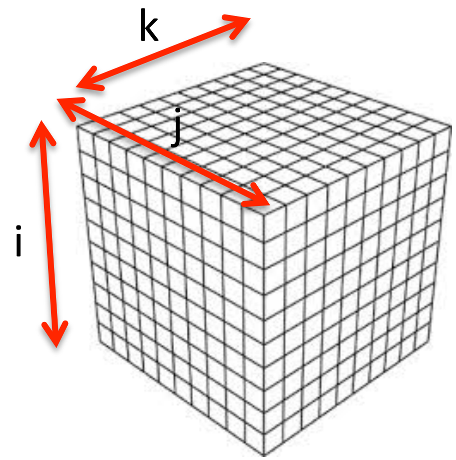
Third-Order Tensors

…are collections of numbers indexed by triples (i,j,k)



Third-Order Tensors

…are collections of numbers indexed by triples (i,j,k)

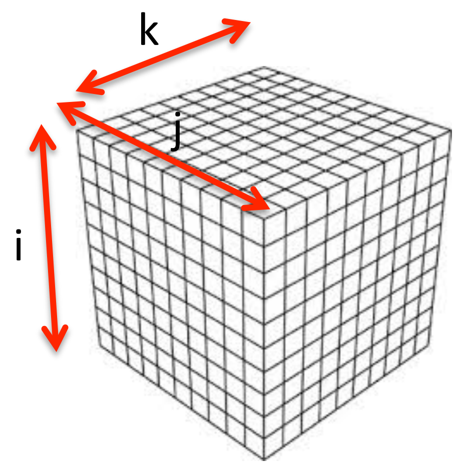
$T =$ 

T is **rank** one if there are vectors a, b and c s.t.

$$T_{i,j,k} = a_i \, b_j \, c_k \quad \forall_{i,j,k}$$

Third-Order Tensors

…are collections of numbers indexed by triples (i,j,k)

$T =$ 

T is **rank** one if there are vectors a, b and c s.t.

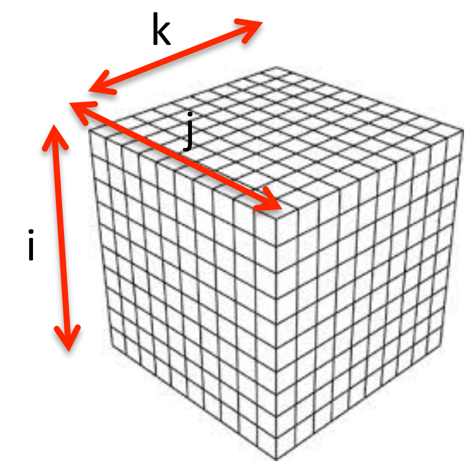
$$T_{i,j,k} = a_i\, b_j\, c_k \quad \forall_{i,j,k}$$

Notation: $T = a \otimes b \otimes c$

Third-Order Tensors

…are collections of numbers indexed by triples (i,j,k)

$$T =$$



T is **rank** one if there are vectors a, b and c s.t.

$$T_{i,j,k} = a_i \, b_j \, c_k \quad \forall_{i,j,k}$$

Notation: $T = a \otimes b \otimes c$ — i.e. $a \otimes b = ab^T$

Low Rank Tensors

T is **rank** at most R if there are vectors $a^{(1)}$, $b^{(1)}$, $c^{(1)}$, …

$$T = \sum_{i=1}^{R} a^{(i)} \otimes b^{(i)} \otimes c^{(i)}$$

Low Rank Tensors

T is **rank** at most R if there are vectors $a^{(1)}, b^{(1)}, c^{(1)}, \ldots$

$$T = \sum_{i=1}^{R} a^{(i)} \otimes b^{(i)} \otimes c^{(i)}$$

Then **any** slice through it is a low-rank matrix

$$T_{(\bullet,\bullet,k)} = \sum_{i=1}^{R} \left[a^{(i)} \otimes b^{(i)} \right] c_k^{(i)}$$

Low Rank Tensors

T is **rank** at most R if there are vectors $a^{(1)}$, $b^{(1)}$, $c^{(1)}$, …

$$T = \sum_{i=1}^{R} a^{(i)} \otimes b^{(i)} \otimes c^{(i)}$$

Then **any** slice through it is a low-rank matrix

$$T_{(\bullet,\bullet,k)} = \sum_{i=1}^{R} \left[a^{(i)} \otimes b^{(i)} \right] c_k^{(i)}$$

They all share the same row and column space too

Low Rank Tensors

T is **rank** at most R if there are vectors $a^{(1)}$, $b^{(1)}$, $c^{(1)}$, …

$$T = \sum_{i=1}^{R} a^{(i)} \otimes b^{(i)} \otimes c^{(i)}$$

different scalings of same rank one terms

Then **any** slice through it is a low-rank matrix

$$T_{(\bullet,\bullet,k)} = \sum_{i=1}^{R} \left[a^{(i)} \otimes b^{(i)} \right] c_k^{(i)}$$

They all share the same row and column space too

Low Rank Tensors

Key Idea: Subtracting off scalings of the same rank one matrix

$$T_{(\bullet,\bullet,k)} - c_k \left[a \otimes b \right]$$

decreases the rank of each slice iff $a = a^{(i)}$, $b = b^{(i)}$, $c = c^{(i)}$ for some i

(under some natural conditions)

Low Rank Tensors

Key Idea: Subtracting off scalings of the same rank one matrix

$$T_{(\bullet,\bullet,k)} - c_k \left[a \otimes b \right]$$

decreases the rank of each slice iff $a = a^{(i)}$, $b = b^{(i)}$, $c = c^{(i)}$ for some i

(under some natural conditions)

This is what makes tensors more **rigid** than matrices

Low Rank Tensors

Key Idea: Subtracting off scalings of the same rank one matrix

$$T_{(\bullet,\bullet,k)} - c_k \left[a \otimes b \right]$$

decreases the rank of each slice iff $a = a^{(i)}$, $b = b^{(i)}$, $c = c^{(i)}$ for some i

(under some natural conditions)

This is what makes tensors more **rigid** than matrices

For matrices, there are many rank one terms we can subtract off to reduce its rank

The Trouble with Tensors

Theorem [Hastad 1990]: Computing the rank of a tensor is NP-hard

The Trouble with Tensors

Theorem [Hastad 1990]: Computing the rank of a tensor is NP-hard

Fact: There are rank three tensors that can be **arbitrarily** well-approximated by rank two tensors

The Trouble with Tensors

Theorem [Hastad 1990]: Computing the rank of a tensor is NP-hard

Fact: There are rank three tensors that can be **arbitrarily** well-approximated by rank two tensors

Fact: The best rank k and the best rank k+1 approximations need not share **any** rank one factors in common

The Trouble with Tensors

Theorem [Hastad 1990]: Computing the rank of a tensor is NP-hard

Fact: There are rank three tensors that can be **arbitrarily** well-approximated by rank two tensors

Fact: The best rank k and the best rank k+1 approximations need not share **any** rank one factors in common

Fact: Even for tensors with real entries, may need complex numbers to find lowest rank decomposition **($rank_R \neq rank_C$)**

The Trouble with Tensors

Theorem [Hastad 1990]: Computing the rank of a tensor is NP-hard

Fact: There are rank three tensors that can be **arbitrarily** well-approximated by rank two tensors

Fact: The best rank k and the best rank k+1 approximations need not share **any** rank one factors in common

Fact: Even for tensors with real entries, may need complex numbers to find lowest rank decomposition **($\text{rank}_R \neq \text{rank}_C$)**

$\vdots$

[Hillar, Lim] "Most Tensor Problems are NP-Hard"

Table I. Tractability of Tensor Problems

Problem	**Complexity**
Bivariate Matrix Functions over $\mathbb{R}$, $\mathbb{C}$	Undecidable (Proposition 12.2)
Bilinear System over $\mathbb{R}$, $\mathbb{C}$	NP-hard (Theorems 2.6, 3.7, 3.8)
Eigenvalue over $\mathbb{R}$	NP-hard (Theorem 1.3)
Approximating Eigenvector over $\mathbb{R}$	NP-hard (Theorem 1.5)
Symmetric Eigenvalue over $\mathbb{R}$	NP-hard (Theorem 9.3)
Approximating Symmetric Eigenvalue over $\mathbb{R}$	NP-hard (Theorem 9.6)
Singular Value over $\mathbb{R}$, $\mathbb{C}$	NP-hard (Theorem 1.7)
Symmetric Singular Value over $\mathbb{R}$	NP-hard (Theorem 10.2)
Approximating Singular Vector over $\mathbb{R}$, $\mathbb{C}$	NP-hard (Theorem 6.3)
Spectral Norm over $\mathbb{R}$	NP-hard (Theorem 1.10)
Symmetric Spectral Norm over $\mathbb{R}$	NP-hard (Theorem 10.2)
Approximating Spectral Norm over $\mathbb{R}$	NP-hard (Theorem 1.11)
Nonnegative Definiteness	NP-hard (Theorem 11.2)
Best Rank-1 Approximation	NP-hard (Theorem 1.13)
Best Symmetric Rank-1 Approximation	NP-hard (Theorem 10.2)
Rank over $\mathbb{R}$ or $\mathbb{C}$	NP-hard (Theorem 8.2)
Enumerating Eigenvectors over $\mathbb{R}$	#P-hard (Corollary 1.16)
Combinatorial Hyperdeterminant	NP-, #P-, VNP-hard (Theorems 4.1 , 4.2, Corollary 4.3)
Geometric Hyperdeterminant	Conjectures 1.9, 13.1
Symmetric Rank	Conjecture 13.2
Bilinear Programming	Conjecture 13.4
Bilinear Least Squares	Conjecture 13.5

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

$$T = \sum_{i=1}^{R} a^{(i)} \otimes b^{(i)} \otimes c^{(i)}$$

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

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

$$T = \sum_{i=1}^{R} a^{(i)} \otimes b^{(i)} \otimes c^{(i)}$$

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

There is a simple algorithm to compute the factors too!

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

$$T = \sum_{i=1}^{R} a^{(i)} \otimes b^{(i)} \otimes c^{(i)}$$

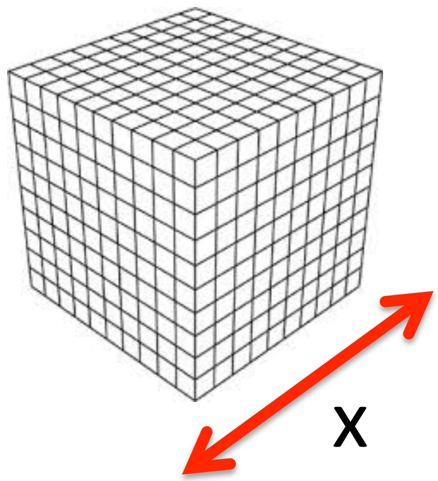
is unique up to permuting the rank one terms and rescaling the factors.

There is a simple algorithm to compute the factors too!

Rediscovered in **[Chang]**, **[Leurgans et al.]**, **[Anandkumar et al.]**, **[Goyal et al.]** …

JENNRICH'S ALGORITHM

- Compute $T(\bullet, \bullet, x)$

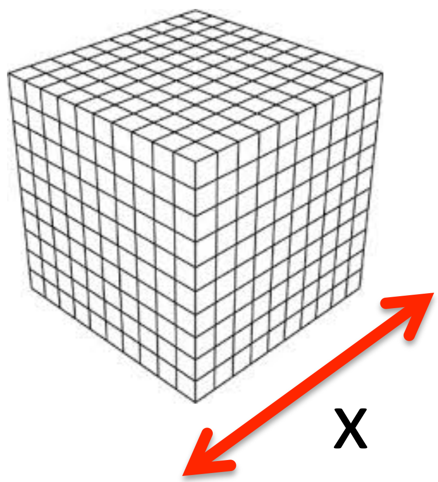


i.e. add up matrix slices

$$\sum x_i \, T_{(i,\bullet,\bullet)}$$

JENNRICH'S ALGORITHM

- Compute $T(\bullet, \bullet, x)$



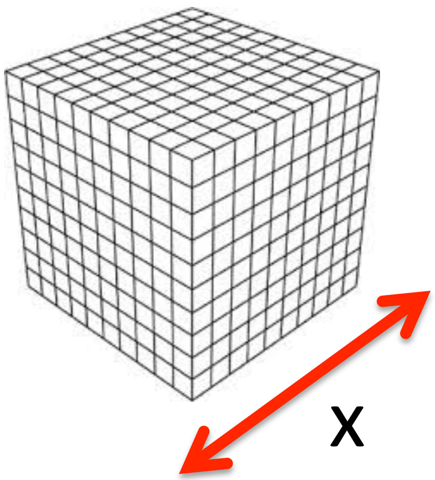
i.e. add up matrix slices

$$\sum x_i \, T_{(i,\bullet,\bullet)}$$

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

JENNRICH'S ALGORITHM

- Compute $T(\bullet, \bullet, x)$

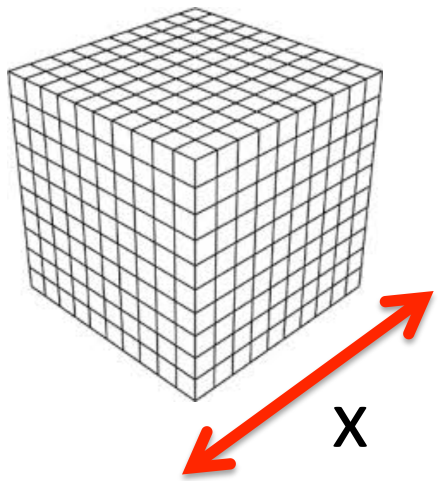


i.e. add up matrix slices

$$\sum x_i \, T_{(i,\bullet,\bullet)}$$

JENNRICH'S ALGORITHM

- Compute $T(\bullet, \bullet, x) = \sum \langle c^{(i)}, x \rangle \, a^{(i)} \otimes b^{(i)}$

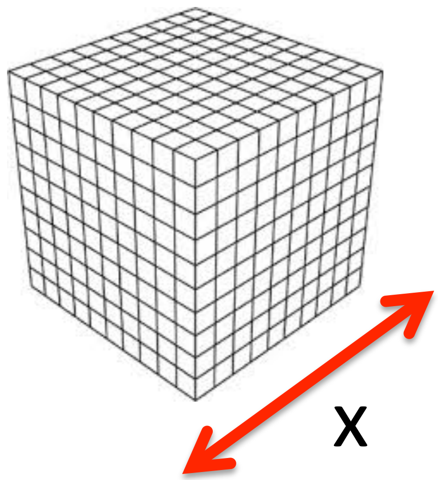


i.e. add up matrix slices

$$\sum x_i \, T_{(i,\bullet,\bullet)}$$

JENNRICH'S ALGORITHM

Compute $T(\bullet, \bullet, x) = \sum \langle c^{(i)}, x \rangle \, a^{(i)} \otimes b^{(i)}$



i.e. add up matrix slices

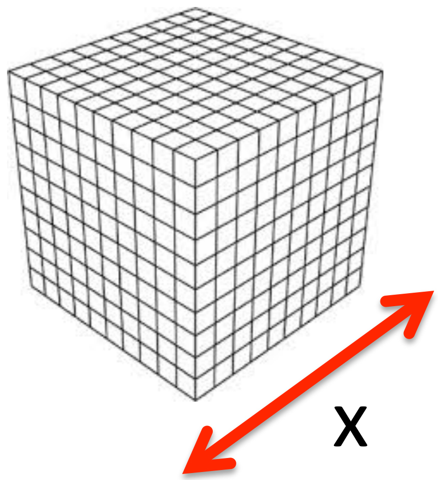
$$\sum x_i \, T_{(i,\bullet,\bullet)}$$

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

JENNRICH'S ALGORITHM

$\mathrm{Diag}(\langle c^{(i)}, x \rangle)$

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



i.e. add up matrix slices

$$\sum x_i\, T_{(i,\bullet,\bullet)}$$

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

JENNRICH'S ALGORITHM

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

JENNRICH'S ALGORITHM

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

JENNRICH'S ALGORITHM

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

JENNRICH'S ALGORITHM

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

JENNRICH'S ALGORITHM

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

JENNRICH'S ALGORITHM

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

Claim: whp (over x,y) the eigenvalues are distinct, so the Eigendecomposition is unique and recovers $a^{(i)}$'s

JENNRICH'S ALGORITHM

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

JENNRICH'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 $T(\bullet, \bullet, x)^{-1}\ T(\bullet, \bullet, y)$

JENNRICH'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 $T(\bullet, \bullet, x)^{-1}\ T(\bullet, \bullet, y)$
- Match up the factors (their eigenvalues are reciprocals) and find $\{c^{(i)}\}$ by solving a linear system

Outline

The focus of this tutorial is on algorithms & applications

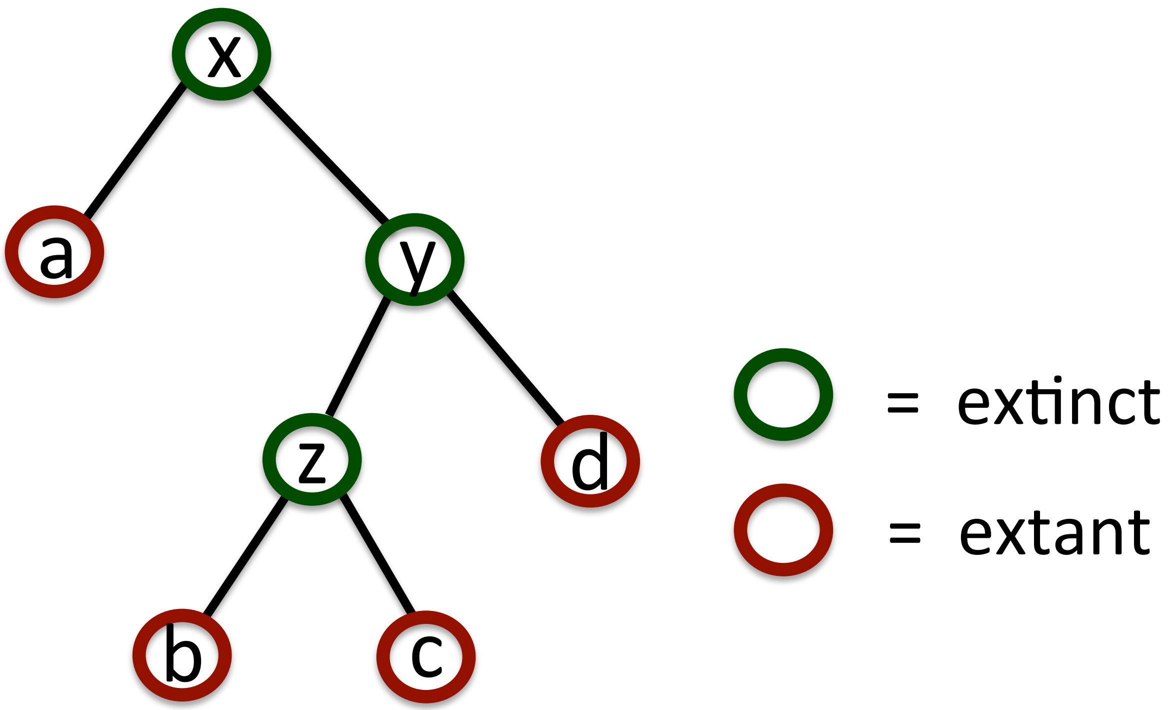
Part I: Tensor Decompositions

- The Rotation Problem
- A Primer on Tensors
- Jennrich's Algorithm

Part II: Applications

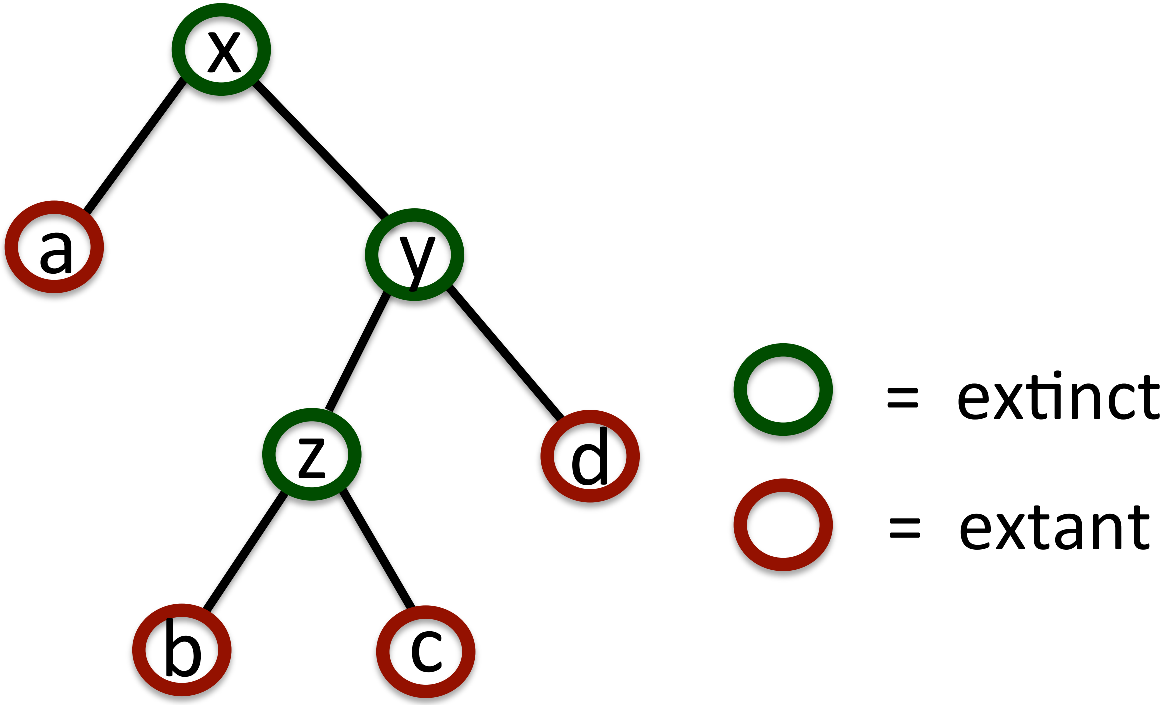
- Phylogenetic Reconstruction
- Pure Topic Models

PHYLOGENETIC RECONSTRUCTION



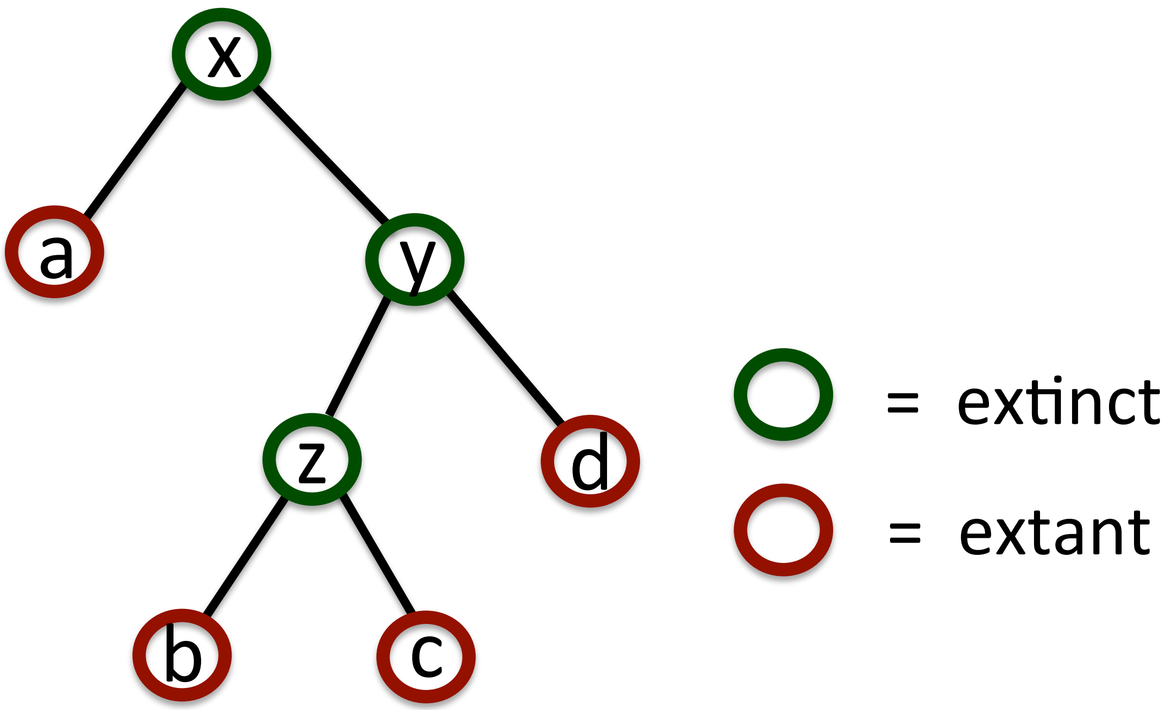
"Tree of Life"

PHYLOGENETIC RECONSTRUCTION



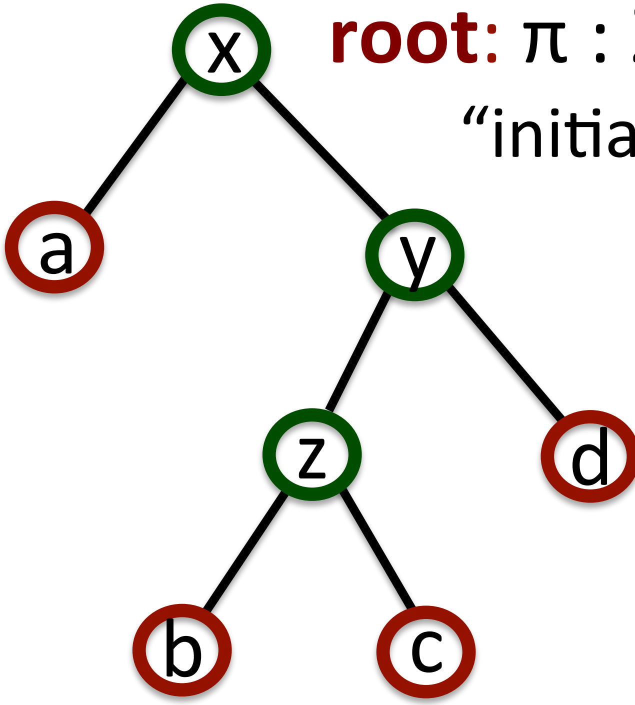
PHYLOGENETIC RECONSTRUCTION

If we've **aligned** sequences…



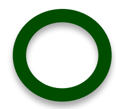
PHYLOGENETIC RECONSTRUCTION

If we've **aligned** sequences…

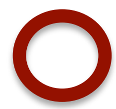


root: $\pi : \Sigma \rightarrow R^+$

"initial distribution"



= extinct



= extant

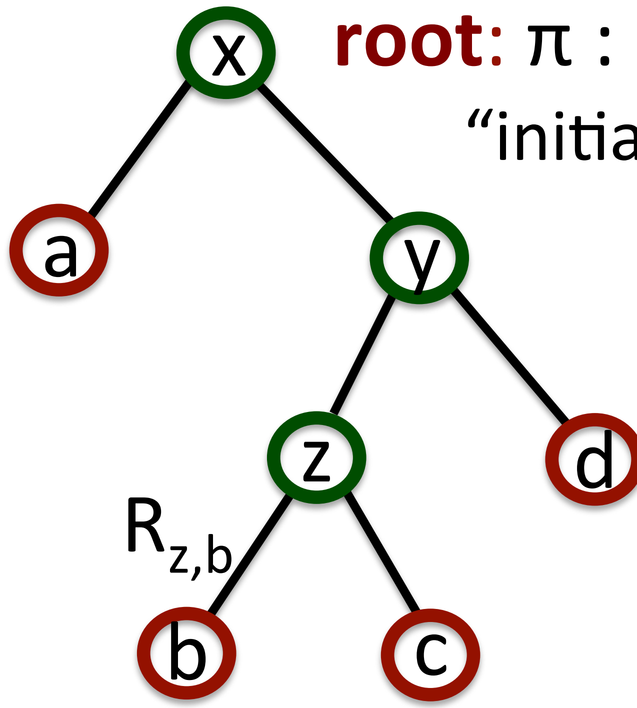
Σ = alphabet

PHYLOGENETIC RECONSTRUCTION

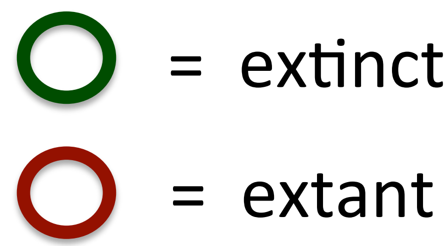
If we've **aligned** sequences…

root: $\pi : \Sigma \rightarrow R^+$

"initial distribution"



"conditional distribution"



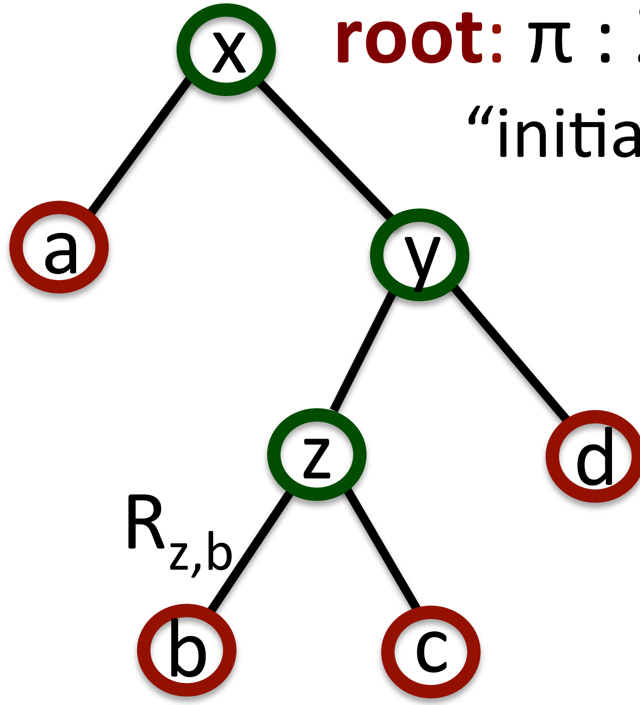
Σ = alphabet

PHYLOGENETIC RECONSTRUCTION

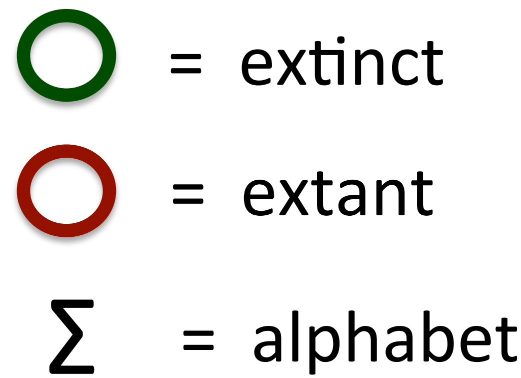
If we've **aligned** sequences…

root: $\pi : \Sigma \rightarrow R^{+}$

"initial distribution"

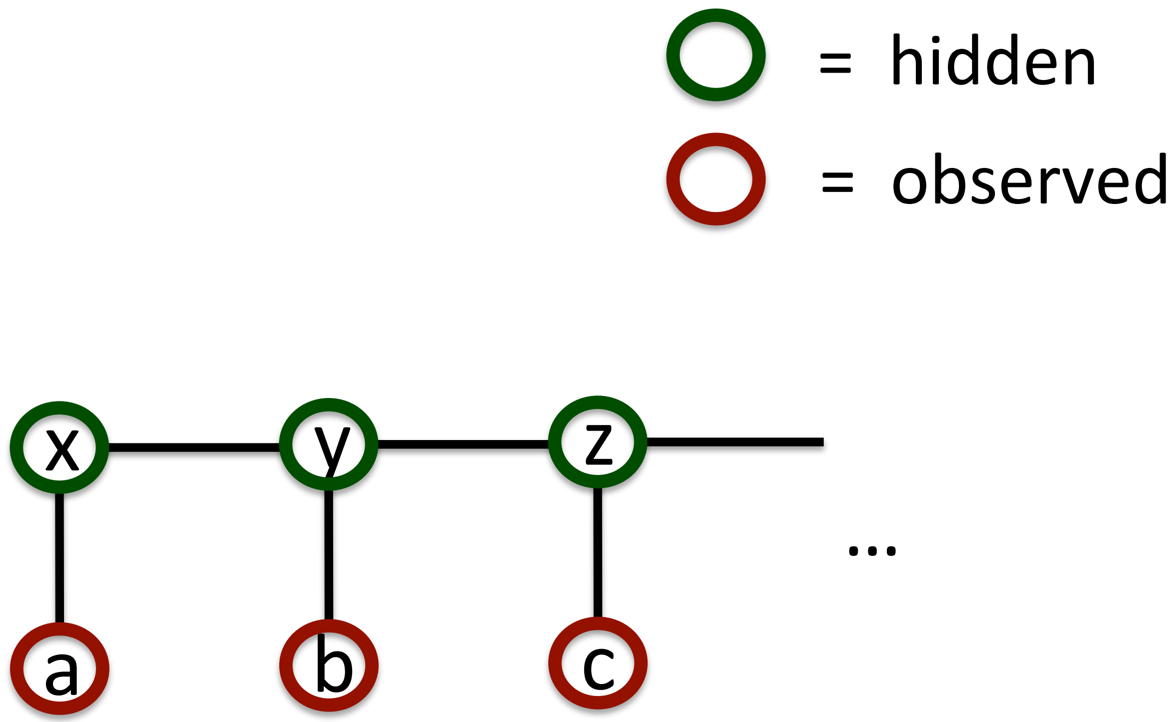


"conditional distribution"



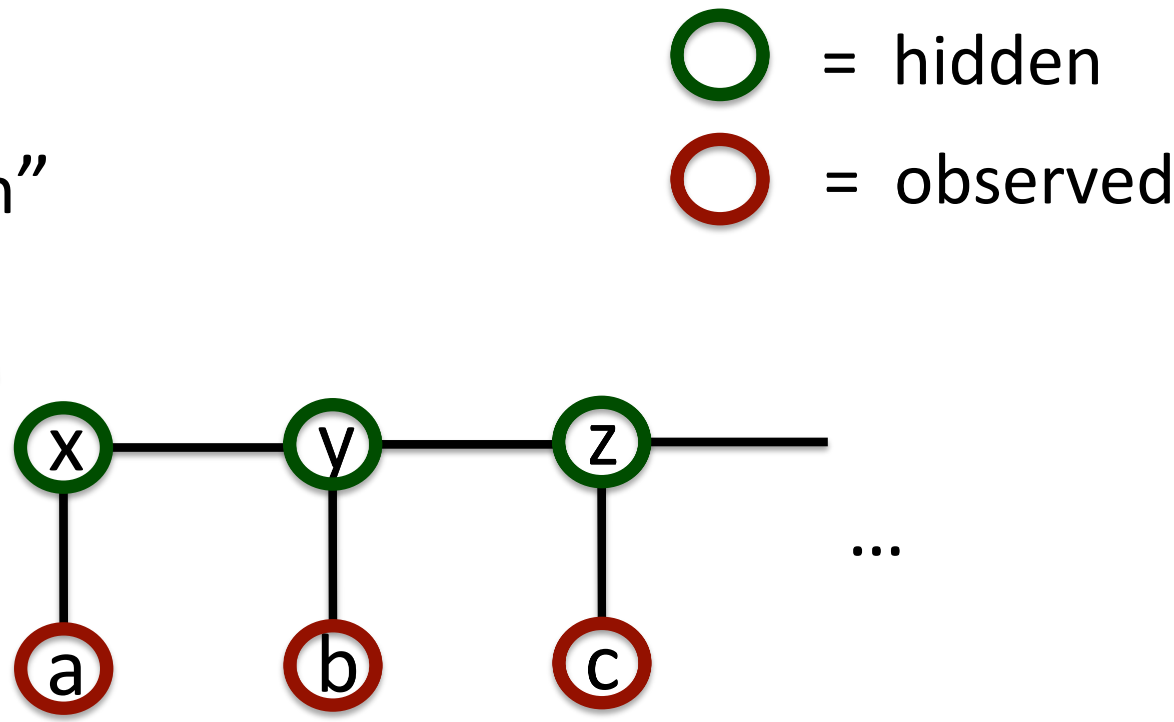
In each sample, we observe a symbol (Σ) at each extant ( ) node where we sample from π for the root, and propagate it using $R_{x,y}$, etc

HIDDEN MARKOV MODELS



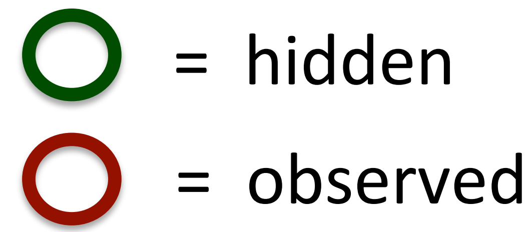
HIDDEN MARKOV MODELS

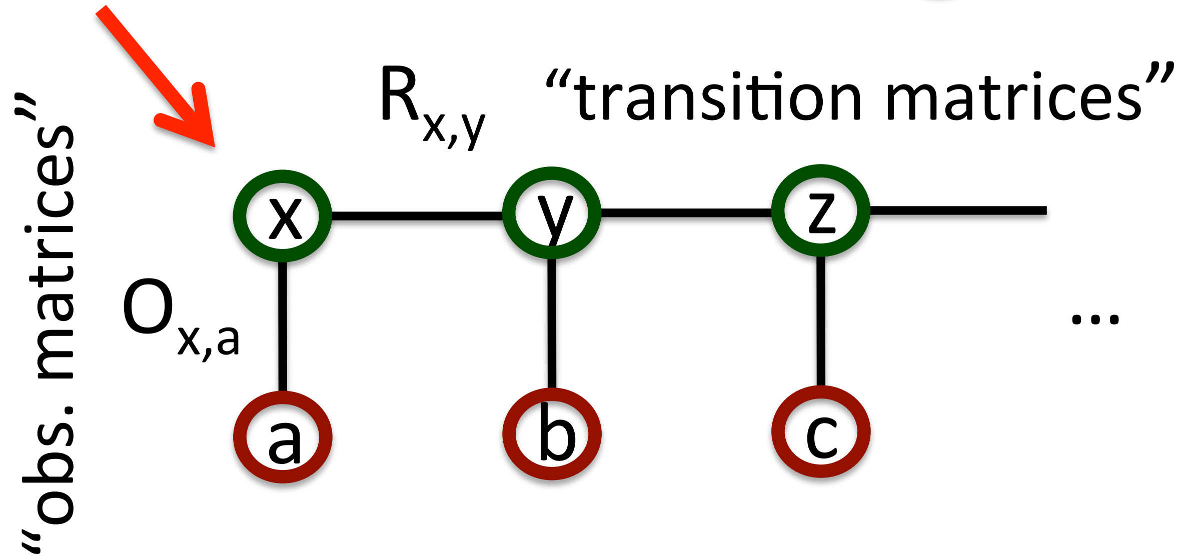
$\pi : \Sigma_s \rightarrow R^+$
"initial distribution"



HIDDEN MARKOV MODELS

$\pi : \Sigma_s \rightarrow R^+$
"initial distribution"

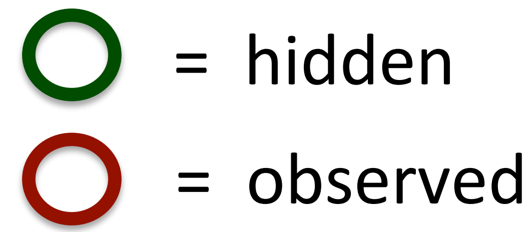


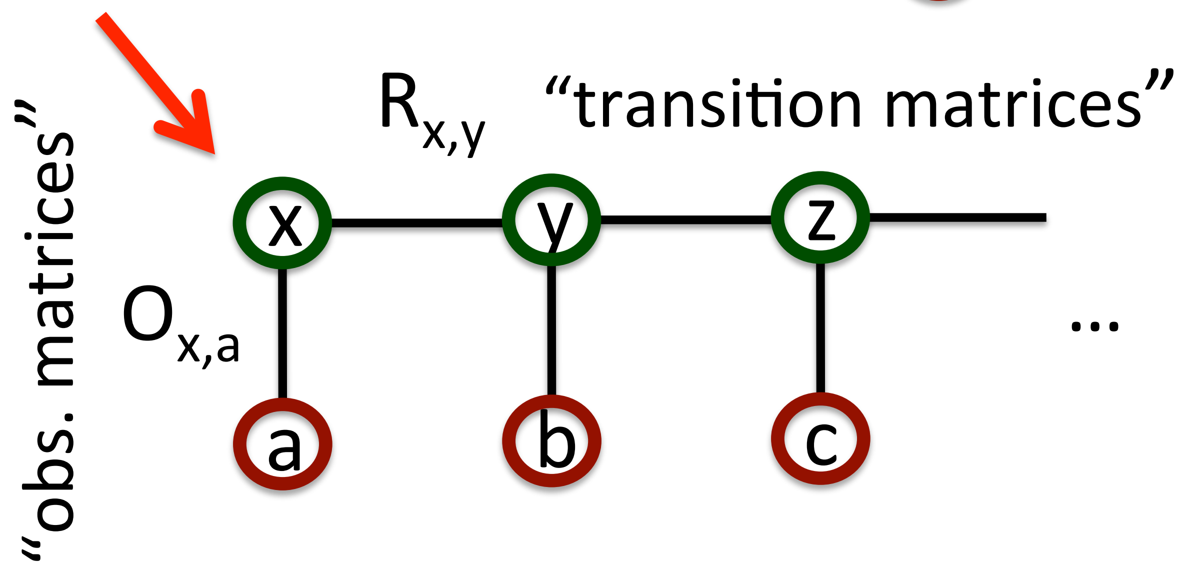


HIDDEN MARKOV MODELS

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

"initial distribution"





In each sample, we observe a symbol (Σ_o) at each obs. ( ) node where we sample from π for the start, and propagate it using $R_{x,y}$, etc (Σ_s)

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

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

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

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

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

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

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

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

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

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!)

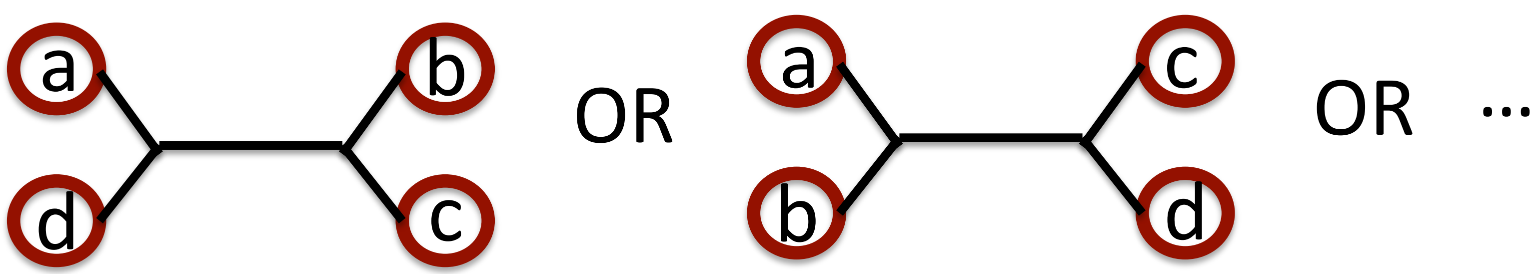
Question: Can we reconstruct just the topology from random samples?

Usually, we assume $R_{x,y}$, 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

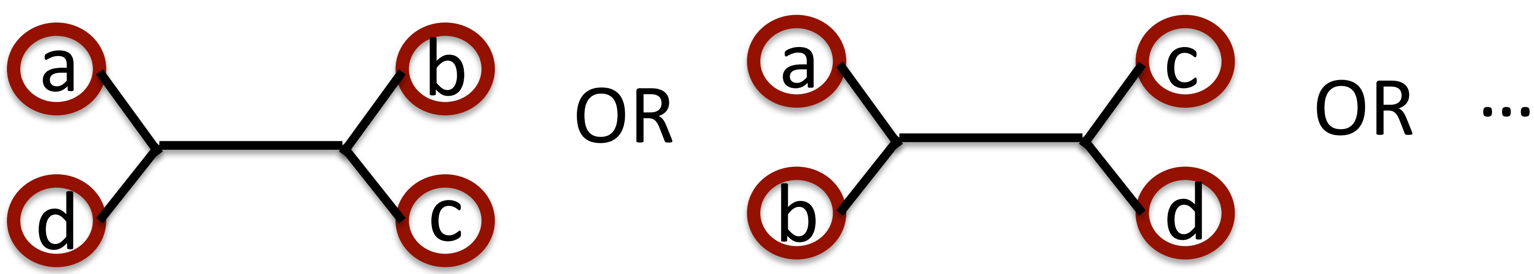


to reconstruction the topology

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

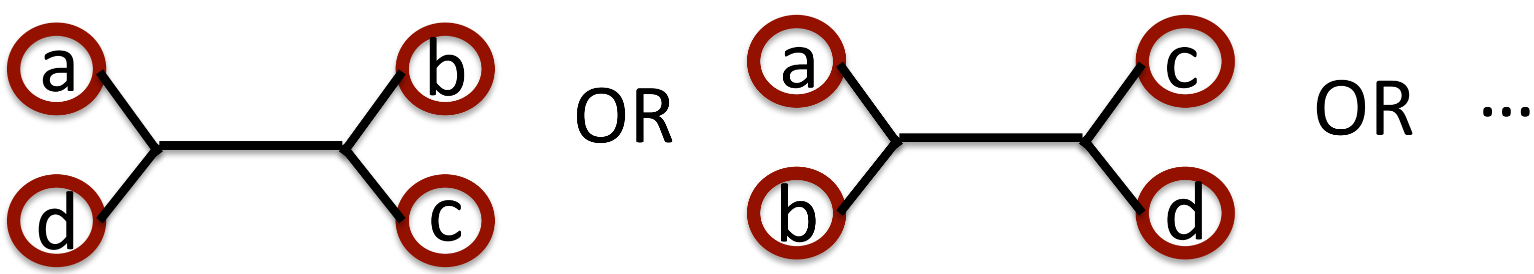


to reconstruction the topology, from polynomially many samples

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

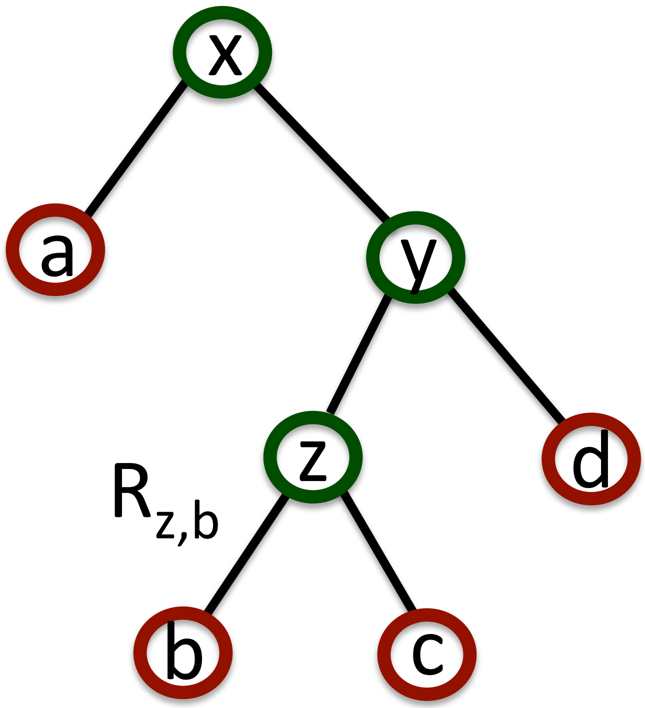


to reconstruction the topology, from polynomially many samples

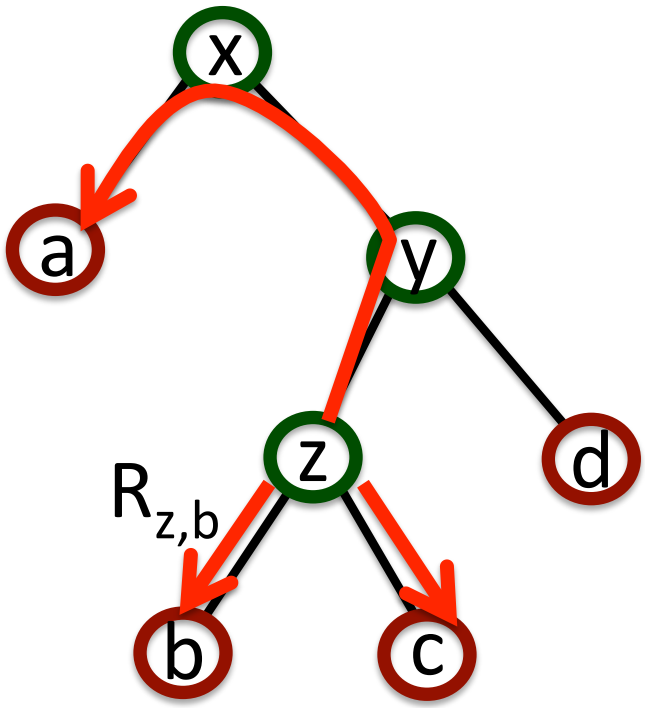
For many problems (e.g. HMMs) finding the transition matrices is the main issue…

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

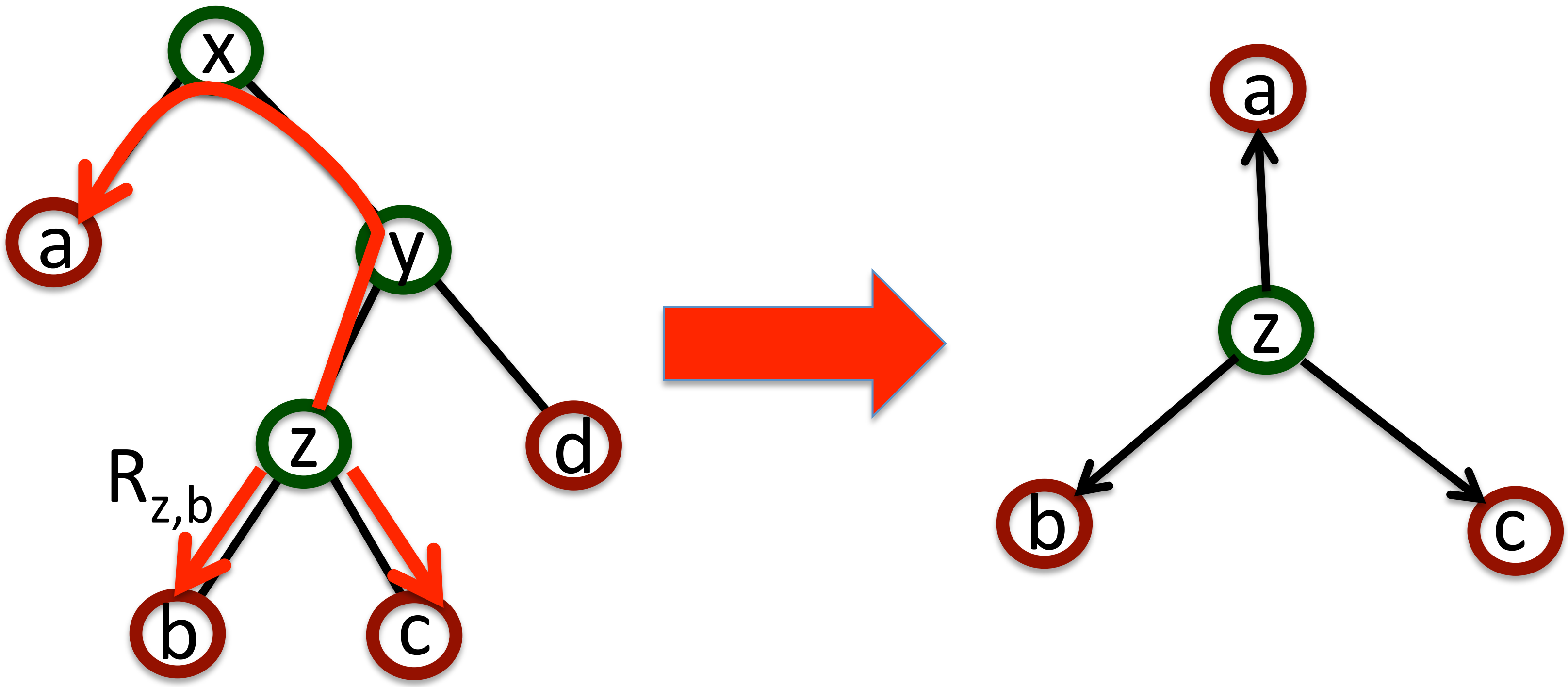
[Chang, 1996]: The model is identifiable (if R's are full rank)



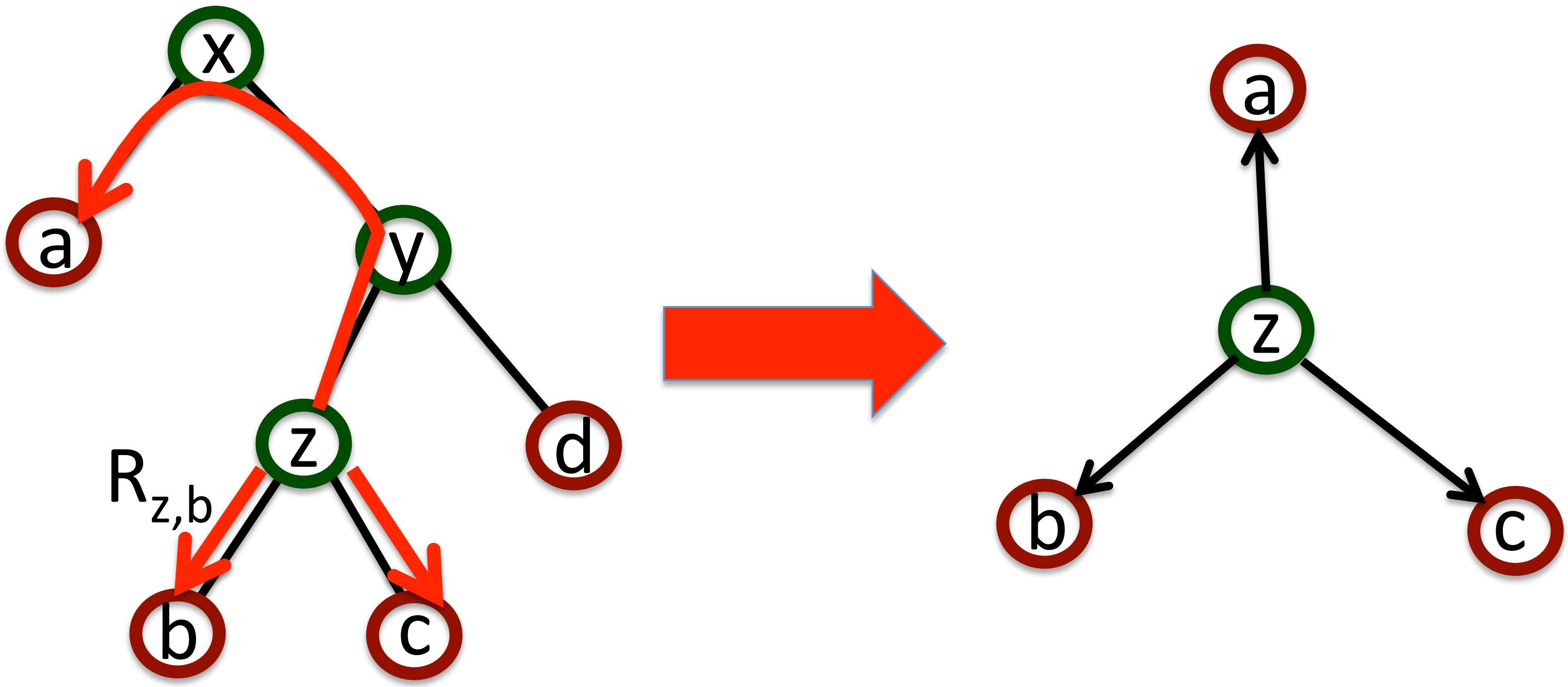
[Chang, 1996]: The model is identifiable (if R's are full rank)



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



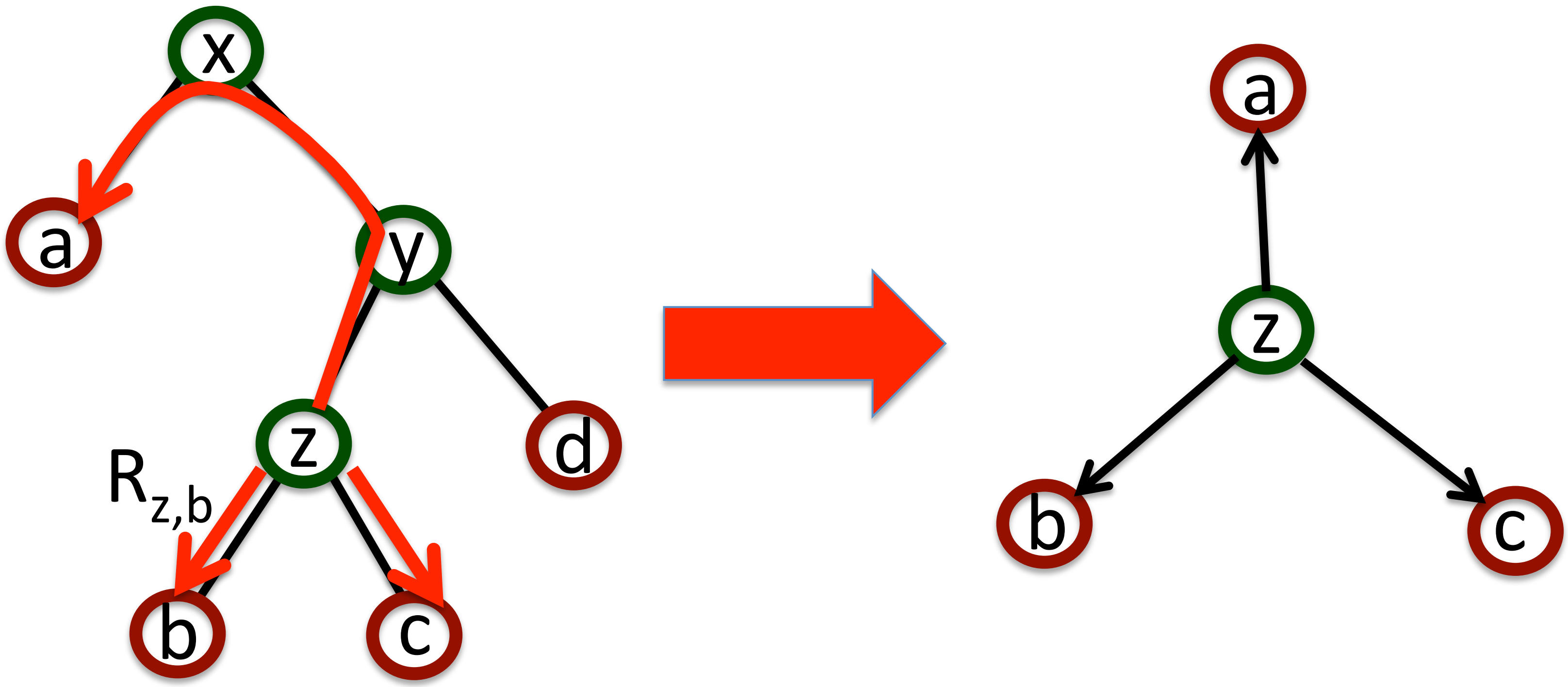
[Chang, 1996]: The model is identifiable (if R’s are full rank)



Joint distribution over (a, b, c):

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

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



Joint distribution over (a, b, c):

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

[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]**

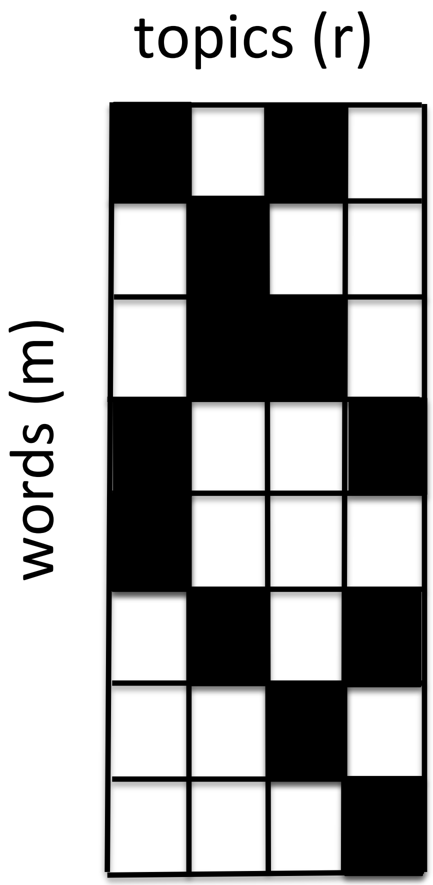
(It's now used as a hard problem to build cryptosystems!)

THE POWER OF CONDITIONAL INDEPENDENCE

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

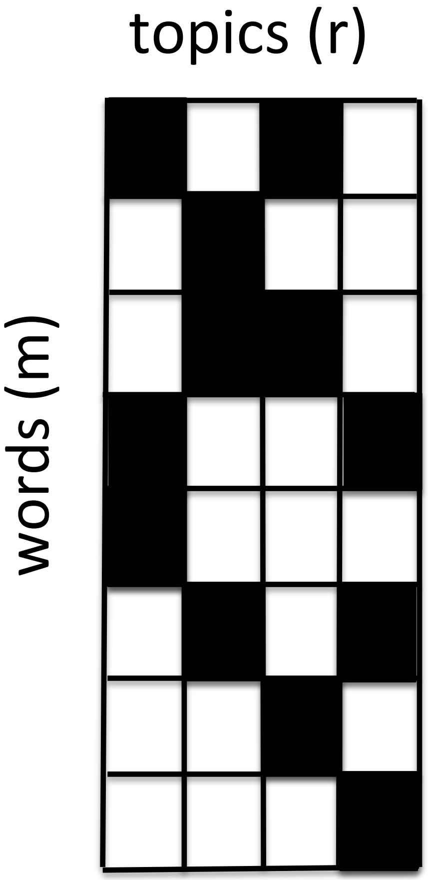
$$\sum_{\sigma} \Pr[z = \sigma] \Pr[a \mid z = \sigma] \otimes \Pr[b \mid z = \sigma] \otimes \Pr[c \mid z = \sigma]$$

PURE TOPIC MODELS



- Each topic is a distribution on words

PURE TOPIC MODELS

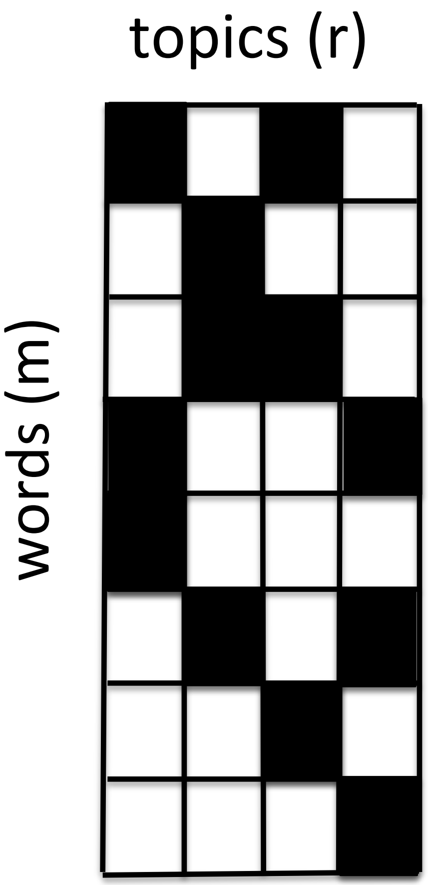


- Each topic is a distribution on words

- **Each document is about only one topic**

  (stochastically generated)

PURE TOPIC MODELS

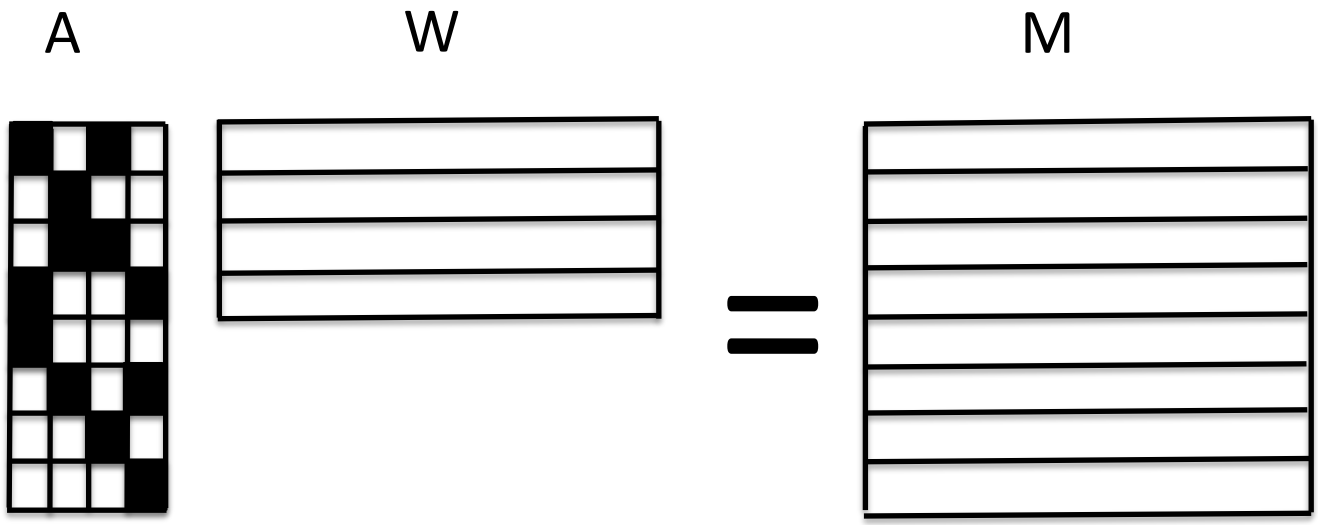


- Each topic is a distribution on words
- **Each document is about only one topic**

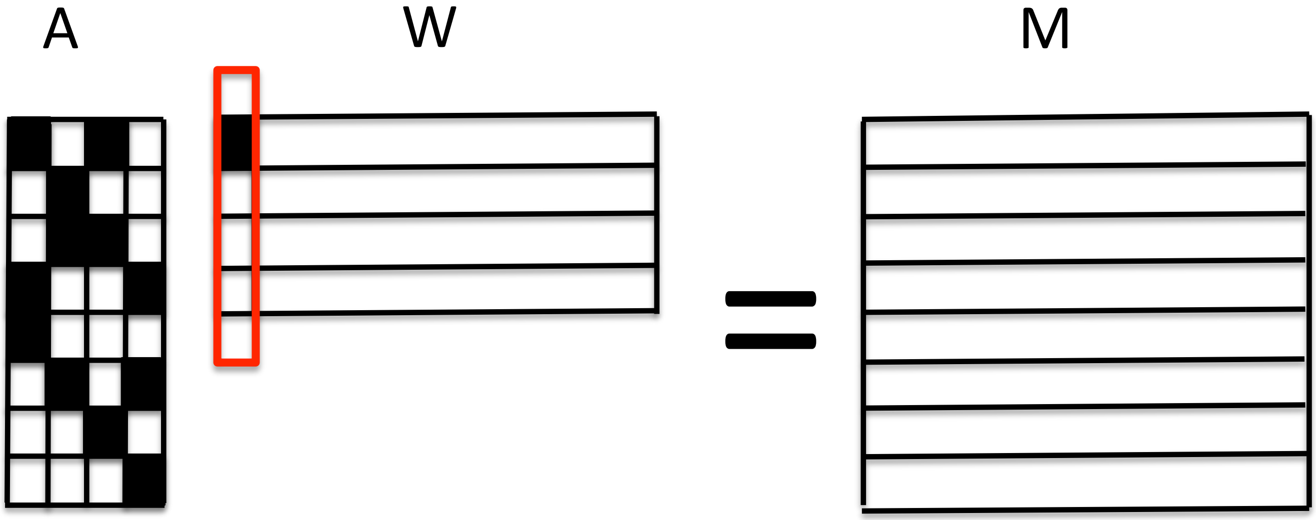
  (stochastically generated)

- Each document, we sample L words from its distribution

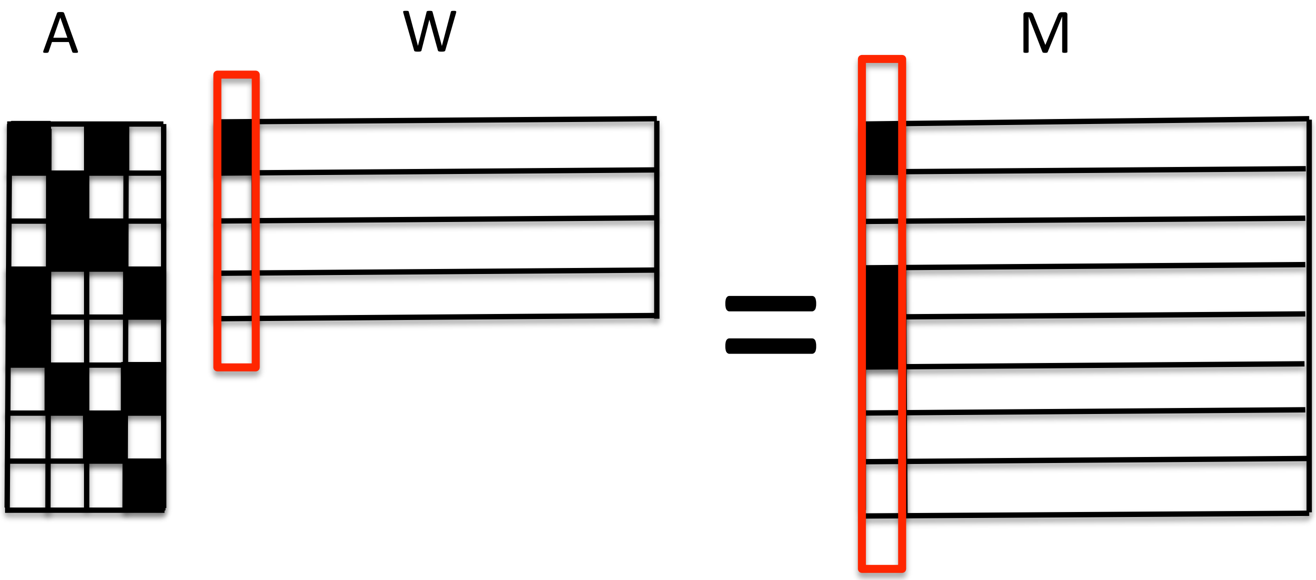
PURE TOPIC MODELS



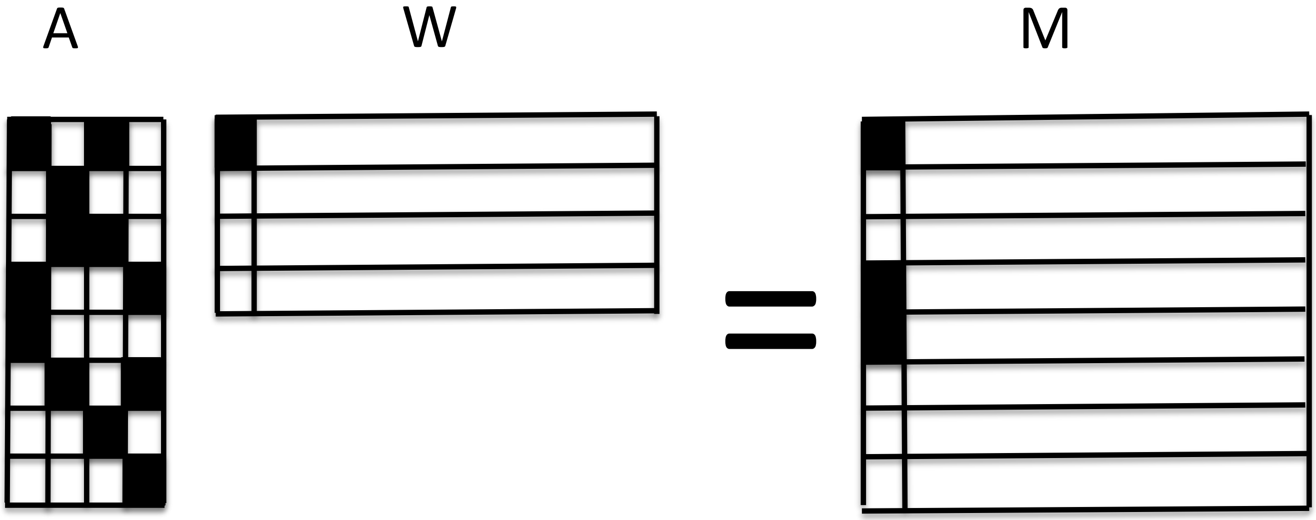
PURE TOPIC MODELS



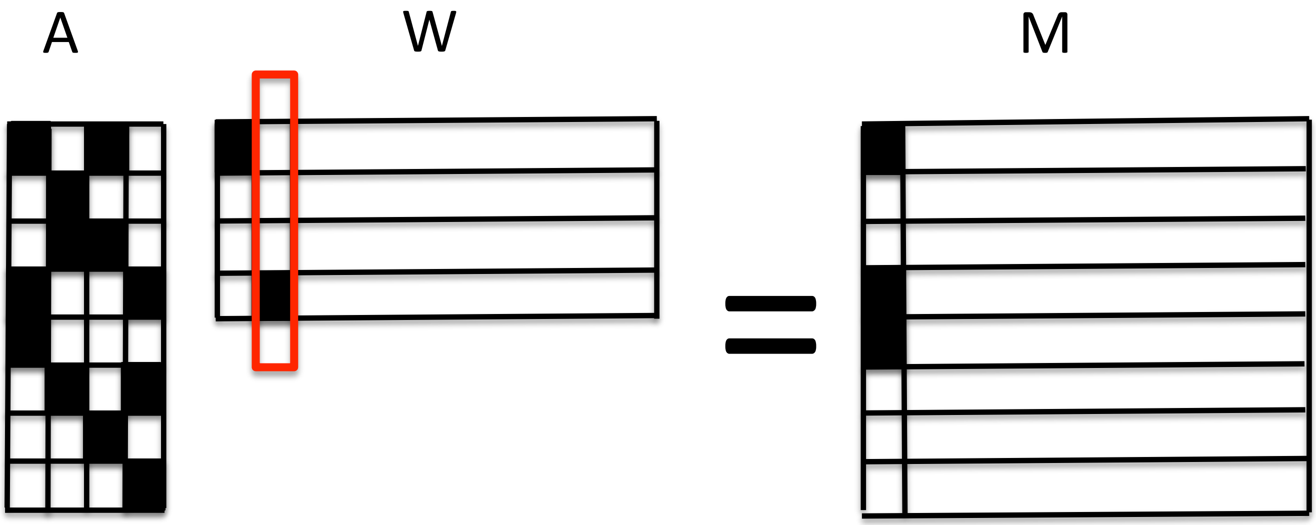
PURE TOPIC MODELS



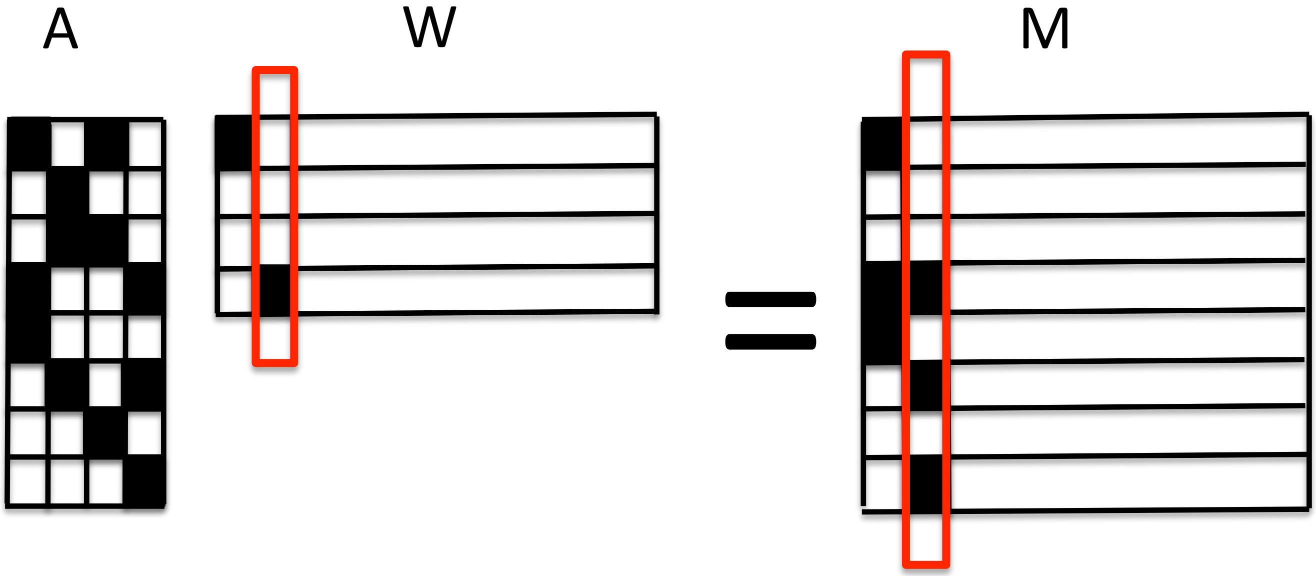
PURE TOPIC MODELS



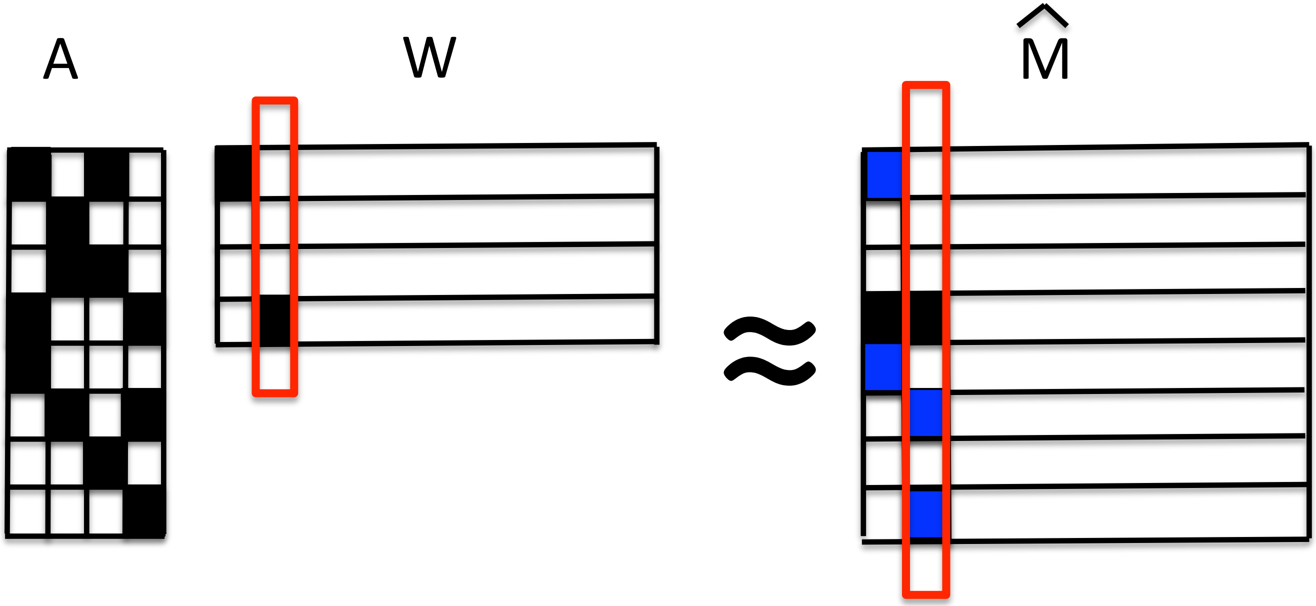
PURE TOPIC MODELS



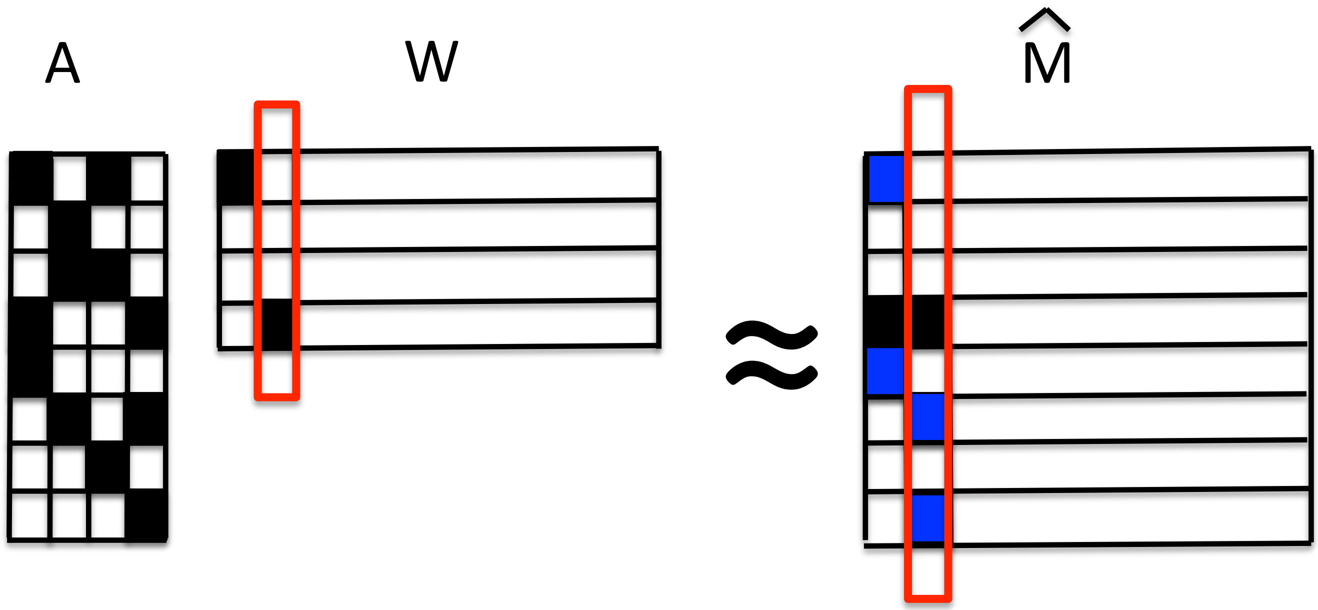
PURE TOPIC MODELS



PURE TOPIC MODELS

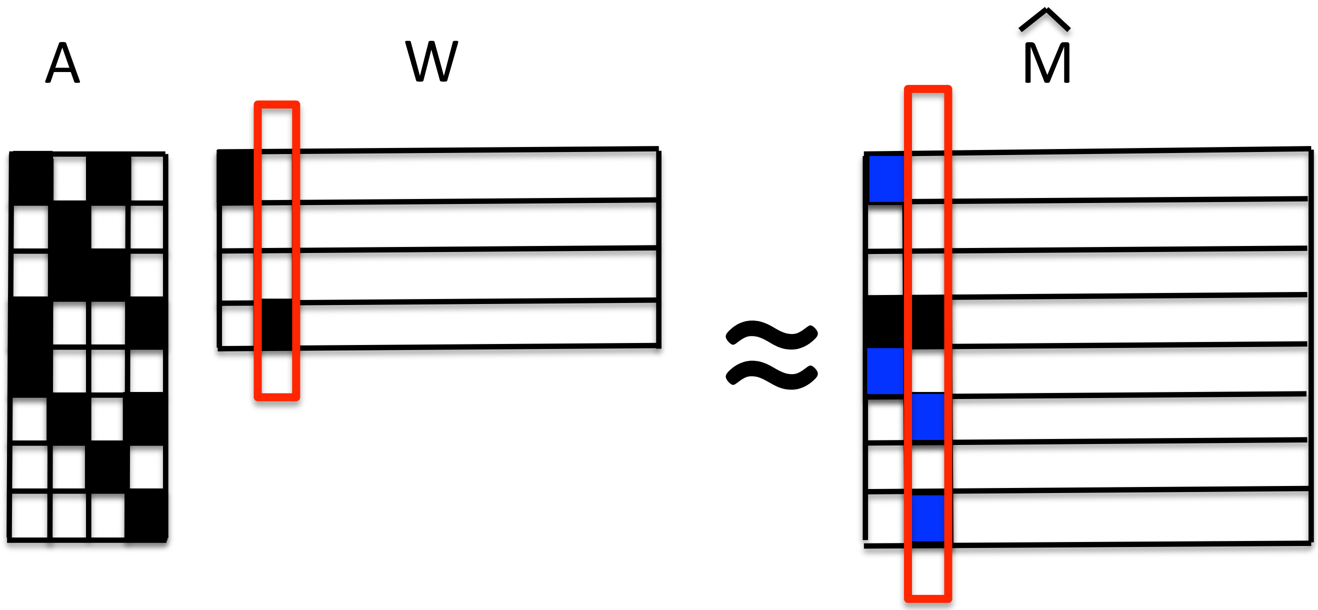


PURE TOPIC MODELS



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

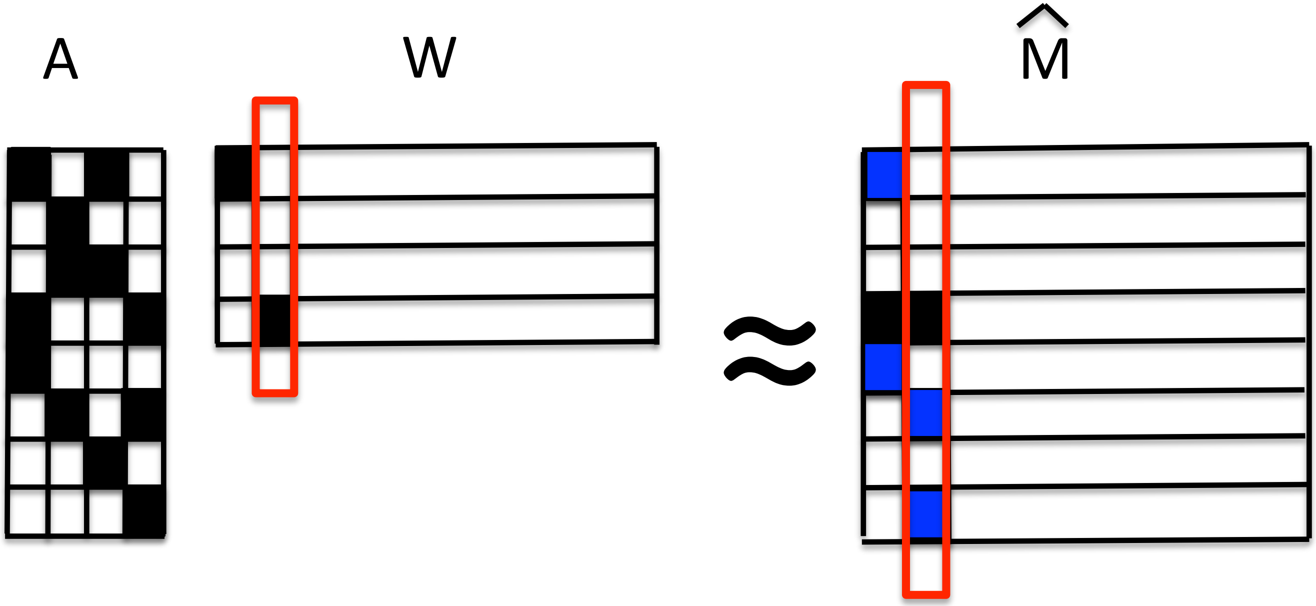
PURE TOPIC MODELS



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

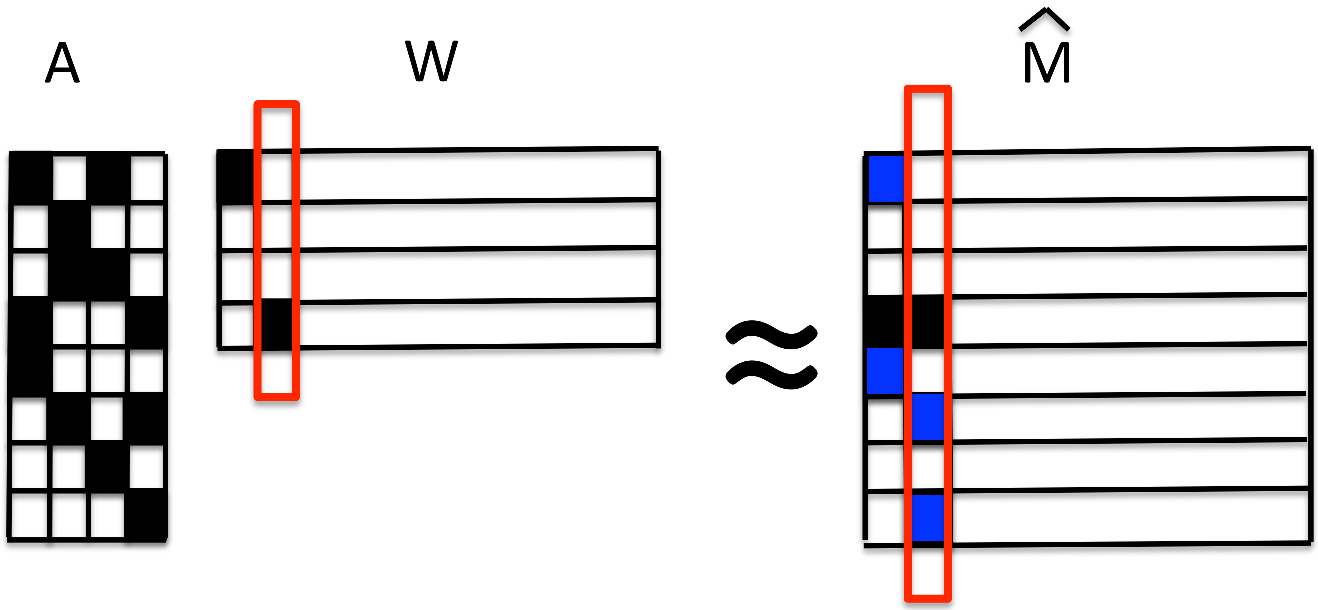
Question: Where can we find three conditionally independent random variables?

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)

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

THE POWER OF CONDITIONAL INDEPENDENCE

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

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

[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)

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

[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)

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

Any Questions?

Summary:

- Spearman's Hypothesis, factor analysis and the rotation problem
- Jennrich's Algorithm
- Applications to phylogenetic trees and topic models
- Are there algorithms for third order tensor decomp. that work with $R = (1+\varepsilon)n$?