A Robust Form of Kruskal's Identifiability Theorem

Aditya Bhaskara (Google NYC)

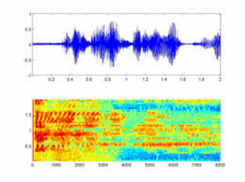
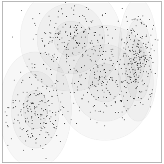
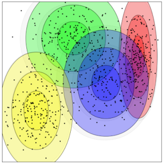
Joint work with

Moses Charikar (Princeton University)

Aravindan Vijayaraghavan (NYU $\rightarrow$ Northwestern)

Background: “understanding” data

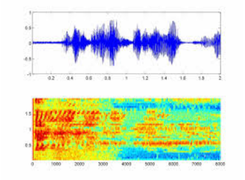
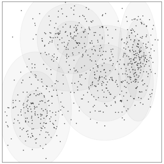
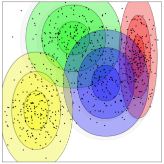
Data from various sources:

QUESTION (“simple explanation”): can we think of data as being *generated* from a model with a small number of parameters?

Background: “understanding” data

Data from various sources:

QUESTION (“simple explanation”): can we think of data as being *generated* from a model with a small number of parameters?

Very successful; e.g. topic models for documents, hidden markov models for speech, . . .

Topic model for documents

Large collection of documents

Carricature model: each document is about a topic, and each topic is a distribution over n words;

Topic model for documents

Large collection of documents

Carricature model: each document is about a topic, and each topic is a distribution over n words; e.g.

	aardvark	apple	ball	lion	...	zebra
sports	0	0.0002	0.005	0.0001	...	
wildlife	0.0006	0	0.0001	0.005	...	

Topic model for documents

Large collection of documents

Carricature model: each document is about a topic, and each topic is a distribution over n words; e.g.

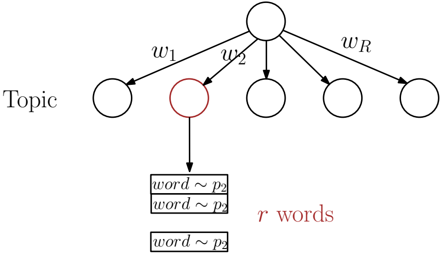
	aardvark	apple	ball	lion	...	zebra
sports	0	0.0002	0.005	0.0001	...	
wildlife	0.0006	0	0.0001	0.005	...	

Model parameters:

- probability that document is on topic i : w_i (sum to 1)
- word probability vector for topic i : $p_i \in \mathbb{R}^n$

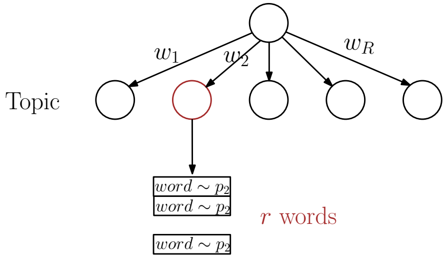
Topic model for documents

Generating a document: say r-word document



Topic model for documents

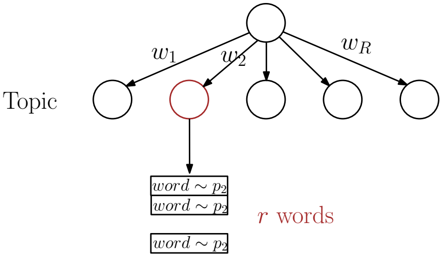
Generating a document: say r-word document



ASSUMPTION: picking a document *from corpus* at random, randomly sampling r words $\equiv$ picking r word document as above

Topic model for documents

Generating a document: say r-word document



ASSUMPTION: picking a document *from corpus* at random, randomly sampling r words $\equiv$ picking r word document as above

Goal: find the $\{w_r, p_r\}$

What about tensors?

Experiment: Pick three random words from random document

QUESTION: What is

$$\Pr[\text{word}_1 = i, \text{word}_2 = j, \text{word}_3 = k]?$$

What about tensors?

Experiment: Pick three random words from random document

QUESTION: What is

$$\Pr[\text{word}_1 = i, \text{word}_2 = j, \text{word}_3 = k]?$$

Precisely equal to $\sum_{r=1}^{R} w_r p_r(i) p_r(j) p_r(k)$.

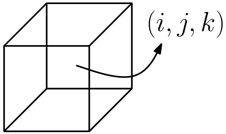
What about tensors?

Experiment: Pick three random words from random document

QUESTION: What is

$$\Pr[\text{word}_1 = i, \text{word}_2 = j, \text{word}_3 = k]?$$

Precisely equal to $\sum_{r=1}^{R} w_r p_r(i) p_r(j) p_r(k)$.



$$= \quad \sum_{r=1}^{R} w_r (p_r \otimes p_r \otimes p_r).$$

$\therefore$ Finding parameters $\equiv$ tensor decomposition!

Recipe for tensor methods in mixture models

Algebraic statistics literature: [Allman, Mathias, Rhodes 09], ...

1. Estimate a tensor whose decomposition allows reading off the model parameters

Recipe for tensor methods in mixture models

Algebraic statistics literature: [Allman, Mathias, Rhodes 09], ...

1. Estimate a tensor whose decomposition allows reading off the model parameters
2. Use tensor decomposition

Recipe for tensor methods in mixture models

Algebraic statistics literature: [Allman, Mathias, Rhodes 09], ...

1. Estimate a tensor whose decomposition allows reading off the model parameters
2. Use tensor decomposition

Many applications: mixtures of gaussians, hidden markov models, communities, crabs, ...

Are we done?

Two caveats

Efficiency

We need polynomial time algorithms for decomposition!

Given $T = \sum_{r=1}^{R} p_r \otimes p_r \otimes p_r$, can we find $\{p_r\}_{r=1}^{R}$?

Two caveats

Efficiency

We need polynomial time algorithms for decomposition!

Given $T = \sum_{r=1}^{R} p_r \otimes p_r \otimes p_r$, can we find $\{p_r\}_{r=1}^{R}$?

Robustness

Algorithm needs to work with *noisy* tensor

In applications, we estimate the tensor from samples. N samples $\implies 1/\sqrt{N}$ error per entry, typically.

Two caveats

Efficiency

We need polynomial time algorithms for decomposition!

Given $T = \sum_{r=1}^{R} p_r \otimes p_r \otimes p_r$, can we find $\{p_r\}_{r=1}^{R}$?

Robustness

Algorithm needs to work with *noisy* tensor

In applications, we estimate the tensor from samples. N samples $\implies 1/\sqrt{N}$ error per entry, typically.

GOAL: Given, target accuracy ε, and

$$T = \sum_{r=1}^{R} p_r \otimes p_r \otimes p_r + \mathcal{N}, \text{ with } \|\mathcal{N}\| < \varepsilon/\text{poly}(n),$$

recover $\{p_r\}$ up to error ε.

A success story: the full rank case

$$\text{Given } T = \sum_{r=1}^{R} p_r \otimes p_r \otimes p_r + \mathcal{N}, \quad \|\mathcal{N}\| < \varepsilon/\text{poly}(n)$$

Define P : matrix $(n \times R)$ with columns $\{p_r\}$.

A success story: the full rank case

$$\text{Given } T = \sum_{r=1}^{R} p_r \otimes p_r \otimes p_r + \mathcal{N}, \quad \|\mathcal{N}\| < \varepsilon/\text{poly}(n)$$

Define P : matrix $(n \times R)$ with columns $\{p_r\}$.

Theorem

If $\sigma_R(P) > 1/poly'(n)$, then can find $\{p_r\}$ up to error ε in polynomial time.

A success story: the full rank case

$$\text{Given } T = \sum_{r=1}^{R} p_r \otimes p_r \otimes p_r + \mathcal{N}, \quad \|\mathcal{N}\| < \varepsilon/\text{poly}(n)$$

Define P : matrix $(n \times R)$ with columns $\{p_r\}$.

Theorem

If $\sigma_R(P) > 1/poly'(n)$, then can find $\{p_r\}$ up to error ε in polynomial time.

Discovered many times: [Harshman 72], [Leurgans, et al. 93], [Chang 96], [Anandkumar, et al. 10], [Goyal et al. 13], ...

Beyond non degeneracy

QUESTION: Can we approximately recover parameters under *weaker* conditions? What about the case $R > n$?

Beyond non degeneracy

QUESTION: Can we approximately recover parameters under *weaker* conditions? What about the case $R > n$?

This talk: easier problem of *identifiability*.

Theorem (Informal theorem.)

If the Kruskal rank condition holds, then it is possible to recover the decomposition up to error ε.

Beyond non degeneracy

QUESTION: Can we approximately recover parameters under *weaker* conditions? What about the case $R > n$?

This talk: easier problem of *identifiability*.

Theorem (Informal theorem.)

If the Kruskal rank condition holds, then it is possible to recover the decomposition up to error ε.

- Not efficient ☹; open problem to do it efficiently

Beyond non degeneracy

QUESTION: Can we approximately recover parameters under *weaker* conditions? What about the case $R > n$?

This talk: easier problem of *identifiability*.

Theorem (Informal theorem.)

If the Kruskal rank condition holds, then it is possible to recover the decomposition up to error ε*.*

- Not efficient ☹; open problem to do it efficiently
- Can be done if the components are *nearly orthogonal* [Anandkumar et al. 14]

Kruskal's theorem (1977)

$$T = \sum_{r=1}^{R} a_i \otimes b_i \otimes c_i$$

GOAL: Find conditions under which the decomposition unique.

Kruskal's theorem (1977)

$$T = \sum_{r=1}^{R} a_i \otimes b_i \otimes c_i$$

GOAL: Find conditions under which the decomposition unique.

Kruskal rank

For matrix $A(n \times R)$, the *rank* is the largest integer $k(A)$ s.t. every $k(A)$ columns of A are linearly independent.

Kruskal's theorem (1977)

$$T = \sum_{r=1}^{R} a_i \otimes b_i \otimes c_i$$

GOAL: Find conditions under which the decomposition unique.

Kruskal rank

For matrix $A(n \times R)$, the *rank* is the largest integer $k(A)$ s.t. every $k(A)$ columns of A are linearly independent.

Note: Much stronger than the *usual* notion of rank; reminiscent of *restricted isometry*

Kruskal's theorem (1977)

$$T = \sum_{r=1}^{R} a_i \otimes b_i \otimes c_i$$

GOAL: Find conditions under which the decomposition unique.

Kruskal rank

For matrix $A(n \times R)$, the *rank* is the largest integer $k(A)$ s.t. every $k(A)$ columns of A are linearly independent.

Note: Much stronger than the *usual* notion of rank; reminiscent of *restricted isometry*

Theorem (Kruskal)

Suppose T is defined as above, and A, B, C are $n \times R$ matrices with columns a_r, b_r, c_r, respectively.
Then a sufficient condition for uniqueness of decomposition is

$$k(A) + k(B) + k(C) \geq 2R + 2$$

Our result

$$T = \sum_{r=1}^{R} a_i \otimes b_i \otimes c_i + \mathcal{N}$$

GOAL: Find conditions under which decomposition is unique*.

Our result

$$T = \sum_{r=1}^{R} a_i \otimes b_i \otimes c_i + \mathcal{N}$$

GOAL: Find conditions under which decomposition is unique*.

Robust Kruskal rank

For matrix $A(n \times R)$ and parameter $\tau > 0$, the *rank* is the largest integer $k_\tau(A)$ s.t. every $n \times k_\tau(A)$ submatrix has condition number $< \tau$.

Note: Recall that condition number of B is $\sigma_{\max}(B)/\sigma_{\min}(B)$.

Our result

$$T = \sum_{r=1}^{R} a_i \otimes b_i \otimes c_i + \mathcal{N}$$

GOAL: Find conditions under which decomposition is unique*.

Robust Kruskal rank

For matrix $A(n \times R)$ and parameter $\tau > 0$, the *rank* is the largest integer $k_\tau(A)$ s.t. every $n \times k_\tau(A)$ submatrix has condition number $< \tau$.

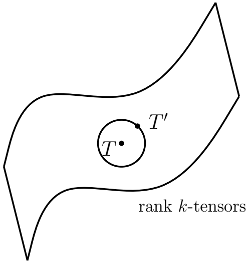
Note: Recall that condition number of B is $\sigma_{\max}(B)/\sigma_{\min}(B)$.

Theorem (Rough)

Let T be as above, and A, B, C be $n \times R$ matrices as before. Then the decomposition is robustly *unique if*

$$k_\tau(A) + k_\tau(B) + k_\tau(C) \geq 2R + 2$$

Our result



We show: For any $\varepsilon > 0$, there is an $\varepsilon' = \varepsilon/\text{poly}(n)$, such that $T =_{\varepsilon'} T'$ implies the decompositions are ε-close, up to a permutation.

Our result

$$T = \sum_{r=1}^{R} a_i \otimes b_i \otimes c_i \quad ; \quad T' = \sum_{r=1}^{R} a'_i \otimes b'_i \otimes c'_i$$

Theorem (Formal)

Suppose T, T' are defined as above, and A, B, C, A', B', C' are $n \times R$ matrices. Further, suppose for some $\tau > 0$, that

$$k_\tau(A) + k_\tau(B) + k_\tau(C) \geq 2R + 2.$$

Then for any $\varepsilon > 0$, there is an $\varepsilon' = \varepsilon/poly(n, \tau)$ such that if $\|T - T'\| < \varepsilon'$, then there exist diagonal matrices $\Gamma_A, \Gamma_B, \Gamma_C$, and a permutation Π such that $\Gamma_A \Gamma_B \Gamma_C = I$, and

$$A' =_\varepsilon \Gamma_A \Pi A, \quad B' =_\varepsilon \Gamma_B \Pi B, \quad C' =_\varepsilon \Gamma_C \Pi C.$$

Remarks

- Conditions only about $k_\tau(A), k_\tau(B), k_\tau(C)$; nothing is needed about A', B', C' (except an upper bound on column lengths)

Remarks

- Conditions only about $k_\tau(A), k_\tau(B), k_\tau(C)$; nothing is needed about A', B', C' (except an upper bound on column lengths)
- Implies that no tensor *close* to T has rank $< R$

Remarks

- Conditions only about $k_\tau(A), k_\tau(B), k_\tau(C)$; nothing is needed about A', B', C' (except an upper bound on column lengths)
- Implies that no tensor *close* to T has rank $< R$
- Naturally generalizes to higher order tensors (similar to De Lathauwer's extension of Kruskal's theorem)

Remarks

- Conditions only about $k_\tau(A), k_\tau(B), k_\tau(C)$; nothing is needed about A', B', C' (except an upper bound on column lengths)
- Implies that no tensor *close* to T has rank $< R$
- Naturally generalizes to higher order tensors (similar to De Lathauwer's extension of Kruskal's theorem)

Main difficulty in proof: handling $1/\text{poly}(n)$ noise

(If $\varepsilon' = \varepsilon/\exp(n)$, much easier to show that decompositions are ε-close)

Proof overview

Suppose A, B, C, A', B', C' are $n \times R$, column lengths $\leq \rho$, and

$$k_\tau(A) = k_\tau(B) = k_\tau(C) = n \quad ; \quad R = 4n/3.$$

(I.e., any n columns of A, B, C are well conditioned (thus lin.ind.))

Proof overview

Suppose A, B, C, A', B', C' are $n \times R$, column lengths $\leq \rho$, and

$$k_\tau(A) = k_\tau(B) = k_\tau(C) = n \quad ; \quad R = 4n/3.$$

(I.e., any n columns of A, B, C are well conditioned (thus lin.ind.))

1. Show that A' is a scaled permutation of A, B' of B, C' of C
2. Show that permutations are equal, and scalings multiply to identity

Proof overview

Suppose A, B, C, A', B', C' are $n \times R$, column lengths $\leq \rho$, and

$$k_\tau(A) = k_\tau(B) = k_\tau(C) = n \quad ; \quad R = 4n/3.$$

(I.e., any n columns of A, B, C are well conditioned (thus lin.ind.))

1. Show that A' is a scaled permutation of A, B' of B, C' of C
2. Show that permutations are equal, and scalings multiply to identity

Crux: first part – "permutation lemma"

Permutation lemma

Sufficient conditions for A, A' having same columns (up to permutation)

(Suppose for now, that no two cols of A' are parallel)**

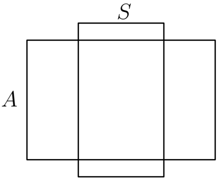
Permutation lemma

Sufficient conditions for A, A' having same columns (up to permutation)

(Suppose for now, that no two cols of A' are parallel)**

Key idea. Look at the spaces spanned by subsets of columns of A, A'.

Definition: $\langle A_S \rangle$:= span of columns indexed by $S \subseteq [R]$



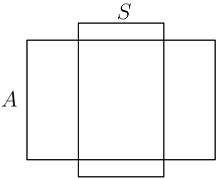
Permutation lemma

Sufficient conditions for A, A' having same columns (up to permutation)

(Suppose for now, that no two cols of A' are parallel)**

Key idea. Look at the spaces spanned by subsets of columns of A, A'.

Definition: $\langle A_S \rangle$:= span of columns indexed by $S \subseteq [R]$



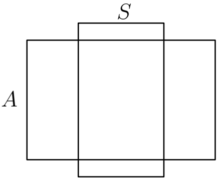
Informal: Suppose for all S of size $(n-1)$, $\langle A'_S \rangle$ "contains" at least $|S|$ columns of A. Then the same is true for all S.

Permutation lemma

Sufficient conditions for A, A' having same columns (up to permutation)

(Suppose for now, that no two cols of A' are parallel)**

Key idea. Look at the spaces spanned by subsets of columns of A, A'.

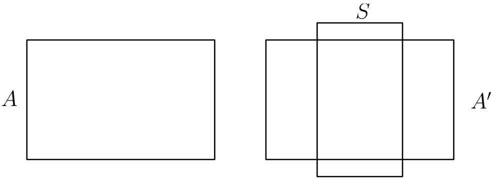


Definition: $\langle A_S \rangle$:= span of columns indexed by $S \subseteq [R]$

Informal: Suppose for all S of size $(n-1)$, $\langle A'_S \rangle$ "contains" at least $|S|$ columns of A. Then the same is true for all S.

(If true for $|S| = 1$, then every column of A' is parallel to some column of A)

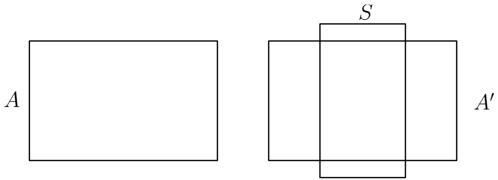
Downward induction: base case $|S| = (n-1)$



$$a_1 \otimes b_1 \otimes c_1 + a_2 \otimes b_2 \otimes c_2 + \cdots \approx a_1' \otimes b_1' \otimes c_1' + a_2' \otimes b_2' \otimes c_2' + \ldots$$

For any w, $\sum_{r=1}^{4n/3} \langle w, a_r \rangle (b_r \otimes c_r) \approx \sum_{r=1}^{4n/3} \langle w, a_r' \rangle (b_r' \otimes c_r')$.

Downward induction: base case $|S| = (n-1)$

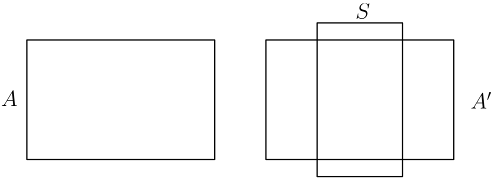


$$a_1 \otimes b_1 \otimes c_1 + a_2 \otimes b_2 \otimes c_2 + \dots \approx a'_1 \otimes b'_1 \otimes c'_1 + a'_2 \otimes b'_2 \otimes c'_2 + \dots$$

For any w, $\sum_{r=1}^{4n/3} \langle w, a_r \rangle (b_r \otimes c_r) \approx \sum_{r=1}^{4n/3} \langle w, a'_r \rangle (b'_r \otimes c'_r)$.

1. Pick w to be orthogonal to $\langle A'_S \rangle$

Downward induction: base case $|S| = (n-1)$

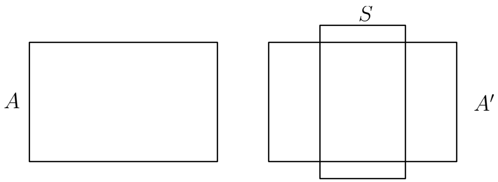


$$a_1 \otimes b_1 \otimes c_1 + a_2 \otimes b_2 \otimes c_2 + \cdots \approx a_1' \otimes b_1' \otimes c_1' + a_2' \otimes b_2' \otimes c_2' + \dots$$

For any w, $\sum_{r=1}^{4n/3} \langle w, a_r \rangle (b_r \otimes c_r) \approx \sum_{r=1}^{4n/3} \langle w, a_r' \rangle (b_r' \otimes c_r')$.

1. Pick w to be orthogonal to $\langle A_S' \rangle$
2. Only $n/3 + 1$ terms remain on RHS $\implies$ at most this number must remain on LHS! (by Kruskal condition)

Downward induction: base case $|S| = (n-1)$

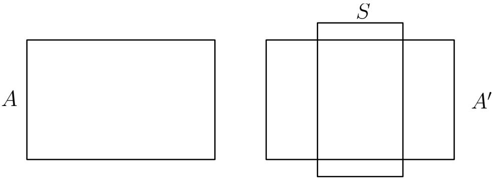


$$a_1 \otimes b_1 \otimes c_1 + a_2 \otimes b_2 \otimes c_2 + \dots \approx a'_1 \otimes b'_1 \otimes c'_1 + a'_2 \otimes b'_2 \otimes c'_2 + \dots$$

For any w, $\sum_{r=1}^{4n/3} \langle w, a_r \rangle (b_r \otimes c_r) \approx \sum_{r=1}^{4n/3} \langle w, a'_r \rangle (b'_r \otimes c'_r)$.

1. Pick w to be orthogonal to $\langle A'_S \rangle$
2. Only $n/3 + 1$ terms remain on RHS $\implies$ at most this number must remain on LHS! (by Kruskal condition)
3. Implies at least $(n-1)$ cols $\{a_r\}$ belong to $\langle A'_S \rangle$

Downward induction: base case $|S| = (n-1)$



$$a_1 \otimes b_1 \otimes c_1 + a_2 \otimes b_2 \otimes c_2 + \dots \approx a_1' \otimes b_1' \otimes c_1' + a_2' \otimes b_2' \otimes c_2' + \dots$$

For any w, $\sum_{r=1}^{4n/3} \langle w, a_r \rangle (b_r \otimes c_r) \approx \sum_{r=1}^{4n/3} \langle w, a_r' \rangle (b_r' \otimes c_r')$.

1. Pick w to be orthogonal to $\langle A_S' \rangle$
2. Only $n/3 + 1$ terms remain on RHS $\implies$ at most this number must remain on LHS! (by Kruskal condition)
3. Implies at least $(n-1)$ cols $\{a_r\}$ belong to $\langle A_S' \rangle$ □

Downward induction

Show for $|S| = (n-2), \ldots, 1$:

Downward induction

Show for $|S| = (n-2), \ldots, 1$:

- start with some S of size $(n-2)$
- define $S_i = S \cup \{i\}$, for various $i \notin S$ — $(n/3) + 2$ such..

Downward induction

Show for $|S| = (n-2), \ldots, 1$:

- start with some S of size $(n-2)$
- define $S_i = S \cup \{i\}$, for various $i \notin S$ — $(n/3) + 2$ such..
- let T_i be indices of cols of A that lie in $\langle A'_{S_i} \rangle$; $|T_i| \geq (n-1)$

Downward induction

Show for $|S| = (n-2), \ldots, 1$:

- start with some S of size $(n-2)$
- define $S_i = S \cup \{i\}$, for various $i \notin S$ — $(n/3) + 2$ such..
- let T_i be indices of cols of A that lie in $\langle A'_{S_i} \rangle$; $|T_i| \geq (n-1)$

Main idea. Any col in $T_i \cap T_j$ must be contained in $\langle A'_S \rangle$

Downward induction

Show for $|S| = (n-2), \ldots, 1$:

- start with some S of size $(n-2)$
- define $S_i = S \cup \{i\}$, for various $i \notin S$ — $(n/3) + 2$ such..
- let T_i be indices of cols of A that lie in $\langle A'_{S_i} \rangle$; $|T_i| \geq (n-1)$

Main idea. Any col in $T_i \cap T_j$ must be contained in $\langle A'_S \rangle$

1. Say we have: $a = \sum_{r \in S} \alpha_r a'_r + \alpha_i a'_i = \sum_{r \in S} \beta_r a'_r + \beta_j a'_j$

Downward induction

Show for $|S| = (n-2), \ldots, 1$:

- start with some S of size $(n-2)$
- define $S_i = S \cup \{i\}$, for various $i \notin S$ — $(n/3) + 2$ such..
- let T_i be indices of cols of A that lie in $\langle A'_{S_i} \rangle$; $|T_i| \geq (n-1)$

Main idea. Any col in $T_i \cap T_j$ must be contained in $\langle A'_S \rangle$

1. Say we have: $a = \sum_{r \in S} \alpha_r a'_r + \alpha_i a'_i = \sum_{r \in S} \beta_r a'_r + \beta_j a'_j$
2. Since a'_i, a'_j are not parallel, this means $\alpha_i = \beta_j = 0$

Downward induction

Show for $|S| = (n-2), \ldots, 1$:

- start with some S of size $(n-2)$
- define $S_i = S \cup \{i\}$, for various $i \notin S$ — $(n/3) + 2$ such..
- let T_i be indices of cols of A that lie in $\langle A'_{S_i} \rangle$; $|T_i| \geq (n-1)$

Main idea. Any col in $T_i \cap T_j$ must be contained in $\langle A'_S \rangle$

Thus if T is the set of cols contained in $\langle A'_S \rangle$,
then for *any* i, j, $T_i \cap T_j = T$

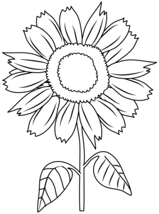
Downward induction

Show for $|S| = (n-2), \ldots, 1$:

- start with some S of size $(n-2)$
- define $S_i = S \cup \{i\}$, for various $i \notin S$ — $(n/3) + 2$ such..
- let T_i be indices of cols of A that lie in $\langle A'_{S_i} \rangle$; $|T_i| \geq (n-1)$

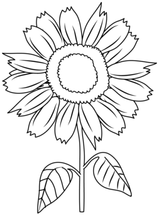
Main idea. Any col in $T_i \cap T_j$ must be contained in $\langle A'_S \rangle$

Thus if T is the set of cols contained in $\langle A'_S \rangle$,
then for *any* i, j, $T_i \cap T_j = T$
Thus T_i form a *sunflower family*



Downward induction

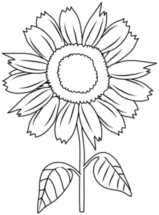
Thus if T is the set of cols contained in $\langle A'_S \rangle$, then for *any* i, j, $T_i \cap T_j = T$



Thus T_i form a *sunflower family*

Downward induction

Thus if T is the set of cols contained in $\langle A'_S \rangle$, then for *any* i, j, $T_i \cap T_j = T$



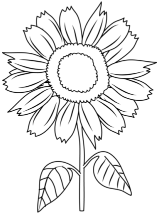
Thus T_i form a *sunflower family*

Claim. Implies that the *core* has size $\geq (n-2)$

Downward induction

Thus if T is the set of cols contained in $\langle A'_S \rangle$, then for *any* i, j, $T_i \cap T_j = T$

Thus T_i form a *sunflower family*



Claim. Implies that the *core* has size $\geq (n-2)$

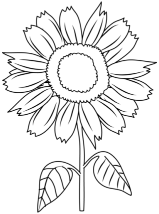
Proof.

Basic counting argument. We know $|T_i| \geq n-1$ for all i.

Downward induction

Thus if T is the set of cols contained in $\langle A'_S \rangle$, then for *any* i, j, $T_i \cap T_j = T$

Thus T_i form a *sunflower family*



Claim. Implies that the *core* has size $\geq (n-2)$

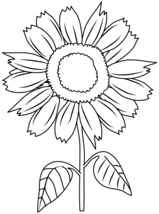
Proof.

Basic counting argument. We know $|T_i| \geq n-1$ for all i.

Say $|T| = n-3$, then $|T_i \setminus T| \geq 2$ for all i.

Downward induction

Thus if T is the set of cols contained in $\langle A'_S \rangle$, then for *any* i, j, $T_i \cap T_j = T$



Thus T_i form a *sunflower family*

Claim. Implies that the *core* has size $\geq (n-2)$

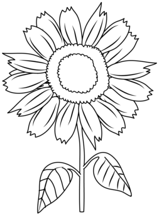
Proof.

Basic counting argument. We know $|T_i| \geq n-1$ for all i.

Say $|T| = n-3$, then $|T_i \setminus T| \geq 2$ for all i.

These don't intersect for different i (sunflower!)

Downward induction



Thus if T is the set of cols contained in $\langle A'_S \rangle$, then for *any* i, j, $T_i \cap T_j = T$

Thus T_i form a *sunflower family*

Claim. Implies that the *core* has size $\geq (n-2)$

Proof.

Basic counting argument. We know $|T_i| \geq n-1$ for all i.

Say $|T| = n-3$, then $|T_i \setminus T| \geq 2$ for all i.

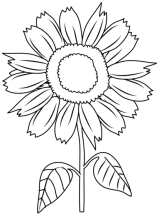
These don't intersect for different i (sunflower!)

Thus the #(cols) of A is $> (n-3) + 2 \times (n/3 + 2) > (4n/3)$..

Downward induction

Thus if T is the set of cols contained in $\langle A'_S \rangle$, then for *any* i, j, $T_i \cap T_j = T$

Thus T_i form a *sunflower family*



Claim. Implies that the *core* has size $\geq (n-2)$

Proof.

Basic counting argument. We know $|T_i| \geq n-1$ for all i.

Say $|T| = n-3$, then $|T_i \setminus T| \geq 2$ for all i.

These don't intersect for different i (sunflower!)

Thus the #(cols) of A is $> (n-3) + 2 \times (n/3 + 2) > (4n/3)$.. □

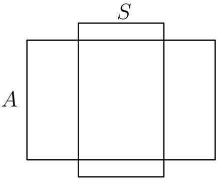
Making things robust – permutation lemma

Sufficient conditions for A, A' having *approximately* same columns (up to permutation)

Making things robust – permutation lemma

Sufficient conditions for A, A' having *approximately* same columns (up to permutation)

Key idea. Look at the spaces spanned by subsets of columns of A, A'.



Informal: Suppose for all S of size $(n-1)$, $\langle A'_S \rangle$ contains at least $|S|$ columns of A, up to error ε. Then the same is true for all S, with error ε'.

(If true for singletons, then every col of A' is ε'-close to a scaling of a col of A)

Issue with inductive proof

Suppose we have claim for $|S| = (n-1)$ with error ε; then can prove claim for $|S| = (n-2)$ with error only $\varepsilon \cdot n$.

Issue with inductive proof

Suppose we have claim for $|S| = (n-1)$ with error ε; then can prove claim for $|S| = (n-2)$ with error only $\varepsilon \cdot n$.

Continuing this way, we only get the claim for $|S| = 1$ with error $\varepsilon \cdot \exp(n)$ ☹

Means the initial error should have been exponentially small..

Issue with inductive proof

Suppose we have claim for $|S| = (n-1)$ with error ε; then can prove claim for $|S| = (n-2)$ with error only $\varepsilon \cdot n$.

Continuing this way, we only get the claim for $|S| = 1$ with error $\varepsilon \cdot \exp(n)$ ☹

Means the initial error should have been exponentially small..

The fix

- More combinatorial invariant (different definition of sets T)
- Stronger sunflower argument
- Better "base case"

Issue with inductive proof

Suppose we have claim for $|S| = (n-1)$ with error ε; then can prove claim for $|S| = (n-2)$ with error only $\varepsilon \cdot n$.

Continuing this way, we only get the claim for $|S| = 1$ with error $\varepsilon \cdot \exp(n)$ ☹

Means the initial error should have been exponentially small..

The fix

- More combinatorial invariant (different definition of sets T)
- Stronger sunflower argument
- Better "base case"

□

Directions

Are there polynomial time algorithms under Kruskal's conditions?

What about other tensor based algorithms in the case $R > n$?

Directions

Are there polynomial time algorithms under Kruskal's conditions?

What about other tensor based algorithms in the case $R > n$?
(a) Higher order tensors: [Cardoso 92], next talk (also [Goyal et al. 13])

Directions

Are there polynomial time algorithms under Kruskal's conditions?

What about other tensor based algorithms in the case $R > n$?
(a) Higher order tensors: [Cardoso 92], next talk (also [Goyal et al. 13])
(b) Recent work assuming *incoherence* [Anandkumar, et al. 14]

Directions

Are there polynomial time algorithms under Kruskal's conditions?

What about other tensor based algorithms in the case $R > n$?
(a) Higher order tensors: [Cardoso 92], next talk (also [Goyal et al. 13])
(b) Recent work assuming *incoherence* [Anandkumar, et al. 14]

Can we show "robust" uniqueness under more algebraic conditions?

Directions

Are there polynomial time algorithms under Kruskal's conditions?

What about other tensor based algorithms in the case $R > n$?
(a) Higher order tensors: [Cardoso 92], next talk (also [Goyal et al. 13])
(b) Recent work assuming *incoherence* [Anandkumar, et al. 14]

Can we show "robust" uniqueness under more algebraic conditions?

Can we show that a generic $n \times n \times n$ tensor has a "stable" unique decomposition up to rank $n^2/4$?

Thank you!

Questions?