Smoothed Analysis of Tensor Decompositions and Learning

Aravindan Vijayaraghavan

CMU $\Rightarrow$ Northwestern University

based on joint works with

Aditya Bhaskara, Google Research

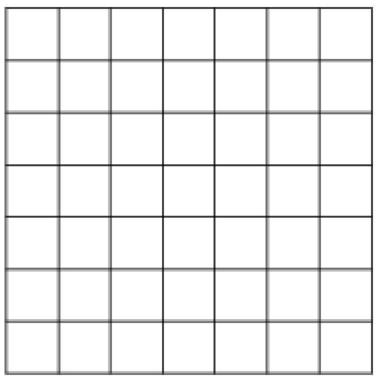
Moses Charikar, Princeton

Ankur Moitra, MIT

Factor analysis

Explain using few unobserved variables

M

d 

d

Assumption: matrix has a "simple explanation"

- Sum of "few" rank one matrices ($k < d$)

$$M = a_1 \otimes b_1 + a_2 \otimes b_2 + \cdots + a_k \otimes b_k$$

Qn [Spearman]. Can we find the ``desired'' explanation ?

The rotation problem

Any suitable "rotation" of the vectors gives a different decomposition

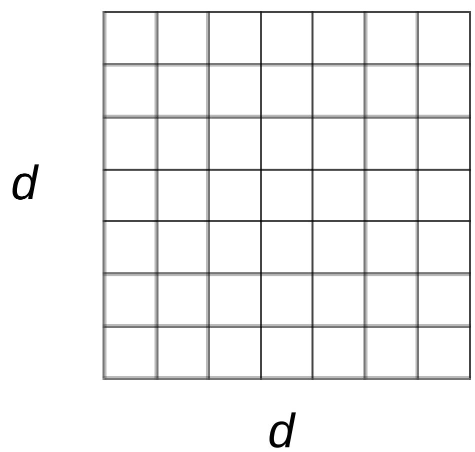
$$M = a_1 \otimes b_1 + a_2 \otimes b_2 + \cdots + a_k \otimes b_k$$

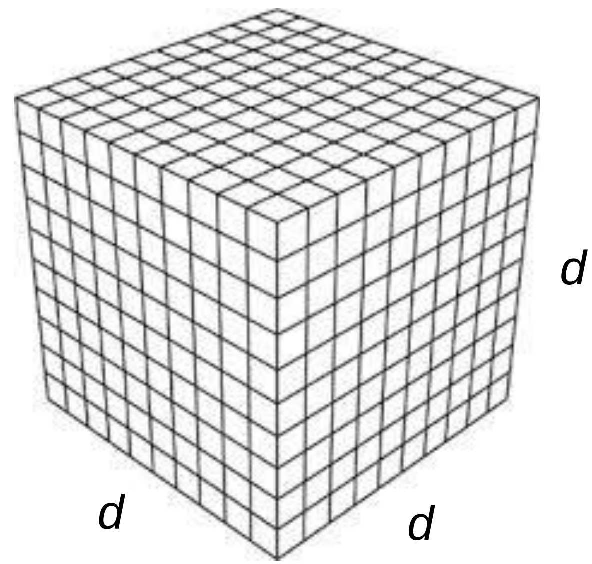
$$A \; B^T = A \; Q \; Q^T \; B^T$$

Often difficult to find "desired" decomposition..

Tensors

Multi-dimensional arrays





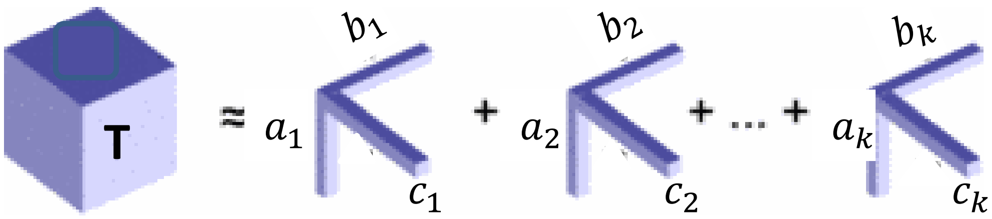
- t dimensional array $\equiv$ tensor of order $t \equiv t$-tensor
- Represent higher order correlations, partial derivatives, etc.
- Collection of matrix (or smaller tensor) slices

3-way factor analysis

Tensor can be written as a sum of few rank-one tensors

3-Tensors:

$$T \;= \sum_{i=1}^{k} a_i \otimes b_i \otimes c_i$$



Rank(T) = smallest k s.t. T written as sum of k rank-1 tensors

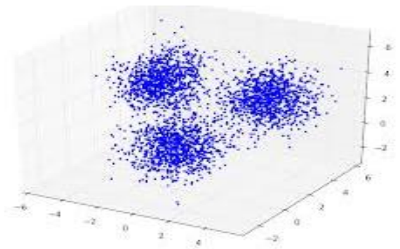
- Rank of 3-tensor $T_{d\times d\times d} \le \; d^2$. Rank of t-tensor $T_{d\times\cdots\times d} \le \; d^{t-1}$

Thm [Harshman'70, Kruskal'77]. Rank-k decompositions for 3-tensors (and higher orders) unique under mild conditions.

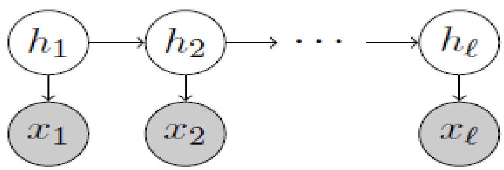
3-way decompositions overcome rotation problem !

Learning Probabilistic Models: Parameter Estimation

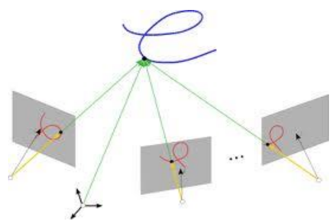
Question: Can given data be "explained" by a simple probabilistic model?



Mixture of Gaussians for clustering points



HMMs for speech recognition



Multiview models

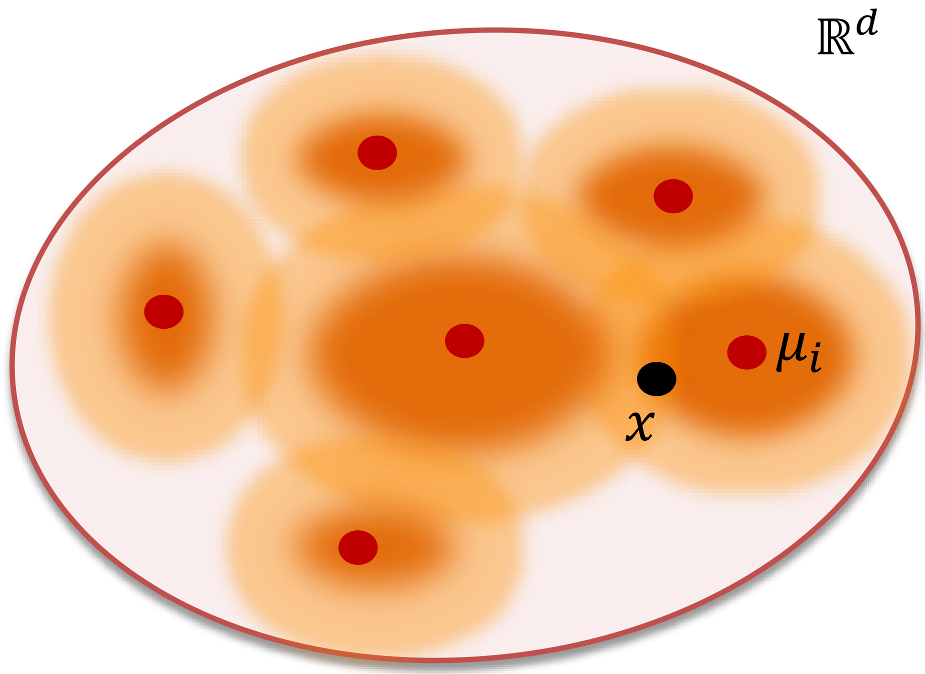
Learning goal: Can the parameters of the model be learned from polynomial samples generated by the model ?

- Algorithms have *exponential* time & sample complexity
- EM algorithm – used in practice, but converges to local optima

Mixtures of (axis-aligned) Gaussians

Probabilistic model for Clustering in d-dims

Parameters
- Mixing weights: $w_1, w_2, \dots, w_k$
- Gaussian G_i : (μ_i, Σ_i)
  mean μ_i , covariance Σ_i : diagonal



Learning problem: Given many sample points, find (w_i, μ_i, Σ_i)

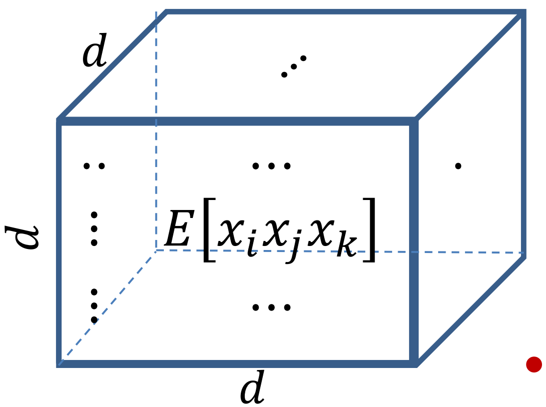
- Algorithms use $\mathbf{O(\exp(\mathit{k}) . \mathit{poly}(\mathit{d}))}$ samples and time [FOS'06, MV'10]
- Lower bound of $\Omega(\mathbf{\exp(\mathit{k})})$ [MV'10] in worst case

Aim: $\boldsymbol{poly(k, d)}$ ***guarantees in realistic settings***

Method of Moments and Tensor decompositions

step 1. compute a tensor whose decomposition *encodes* model parameters

step 2. find decomposition (and hence parameters)

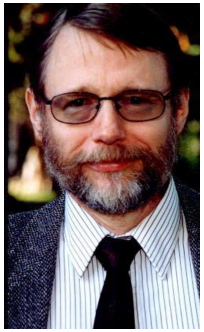


$$\boldsymbol{T} = \sum_{i=1}^{k} \boldsymbol{w_i}\, \boldsymbol{\mu_i} \otimes \boldsymbol{\mu_i} \otimes \boldsymbol{\mu_i}$$

- *Uniqueness $\Longrightarrow$ Recover parameters w_i and μ_i*
- *Algorithm for Decomposition $\Longrightarrow$ efficient learning*

[Chang] [Allman, Matias, Rhodes]
[Anandkumar,Ge,Hsu, Kakade, Telgarsky]

What is known about Tensor Decompositions ?



Thm [Jennrich via Harshman'70]. Find unique rank-k decompositions for 3-tensors when $k \leq d$!

- Uniqueness proof is *algorithmic* !
- Called Full-rank case. No symmetry or orthogonality needed.
- Rediscovered in **[Leurgans et al 1993] [Chang 1996]**

Thm [Kruskal'77]. Rank-k decompositions for 3-tensors unique (non-algorithmic) when $k \leq 3d/2$!

Thm [Chiantini Ottaviani'12].
Uniqueness (non-algorithmic) of 3-tensors of rank $k \leq c.\, d^2$ *generically*

Thm [DeLathauwer, Castiang, Cardoso'07].
Algorithm for 4-tensors of rank k *generically* when $k \leq c.\, d^2$

Robustness to Errors

Beware : Sampling error

Empirical estimate $T =_\epsilon \sum_{i=1}^{k} w_i \ \mu_i \otimes \mu_i \otimes \mu_i$

With $\mathrm{poly}(\mathrm{d},\mathrm{k})$ samples, error $\epsilon \approx 1/\mathrm{poly}(\mathrm{d},\mathrm{k})$

Uniqueness and Algorithms resilient to noise of 1/poly(d,k) ?

Thm. Jennrich's polynomial time algorithm for Tensor Decompositions robust up to $1/poly(d,k)$ error

Thm [BCV'14]**.** Robust version of Kruskal Uniqueness theorem (non-algorithmic) with $1/poly(d,k)$ error

Open Problem: Robust version of generic results[De Lauthewer et al]?

Algorithms for Tensor Decompositions

Polynomial time algorithms when rank $k \leq d$ [Jennrich]

NP-hard when rank $k > d$ in worst case [Hastad, Hillar-Lim]

This talk

Overcome worst-case intractability using Smoothed Analysis

- **Polynomial time algorithms* for robust Tensor decompositions for rank k >> d (rank is any polynomial in dimension)**

*Algorithms $poly(d, k, 1/\epsilon)$ for recovery up to ϵ error in $\|.\|_F$

Implications for Learning

Known only in restricted cases:

No. of clusters $k \leq$ No. of dims d

``Full rank'' or ``Non-degenerate'' setting

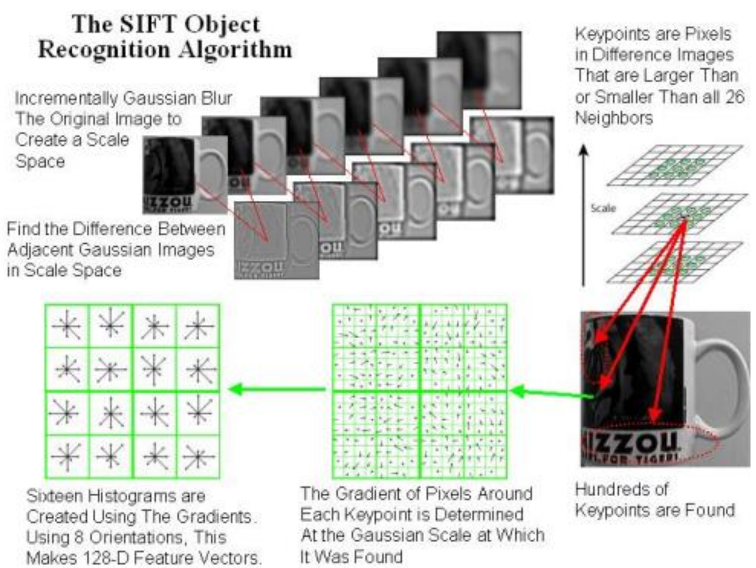
Efficient Learning when no. of clusters/ topics k $\leq$ dimension d

[Chang 96, Mossel-Roch 06, Anandkumar et al. 09-14]

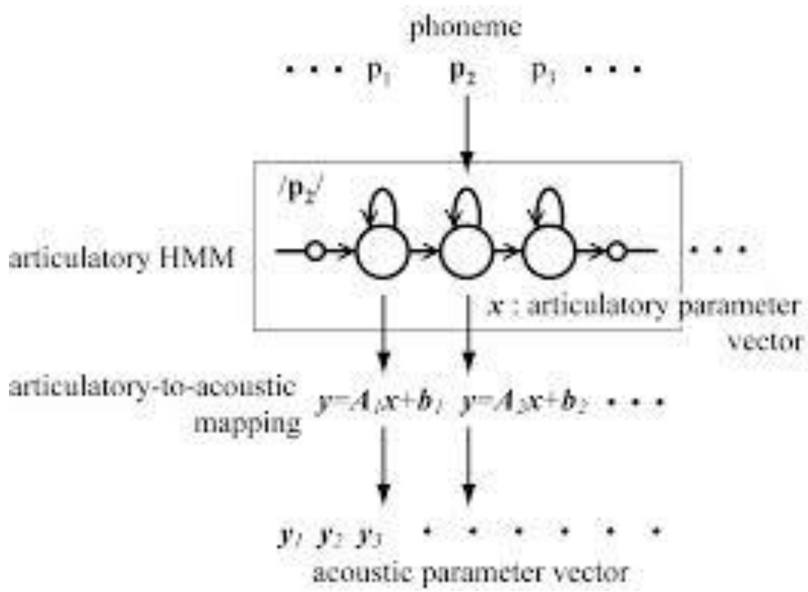
- Learning Phylogenetic trees [Chang,MR]
- Axis-aligned Gaussians [HK]
- Parse trees [ACHKSZ,BHD,B,SC,PSX,LIPPX]
- HMMs [AHK,DKZ,SBSGS]
- Single Topic models [AHK], LDA [AFHKL]
- ICA [GVX] …
- Overlapping Communities [AGHK] …

Overcomplete Learning Setting

Number of clusters/topics/states $\mathbf{k} \gg$ dimension $\mathbf{d}$



Computer Vision

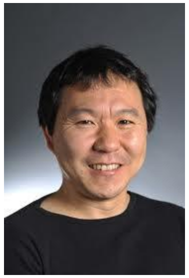


Speech

✗ Previous algorithms do not work when $k > d$!

Need polytime decomposition of Tensors of rank $k \gg d$?

Smoothed Analysis

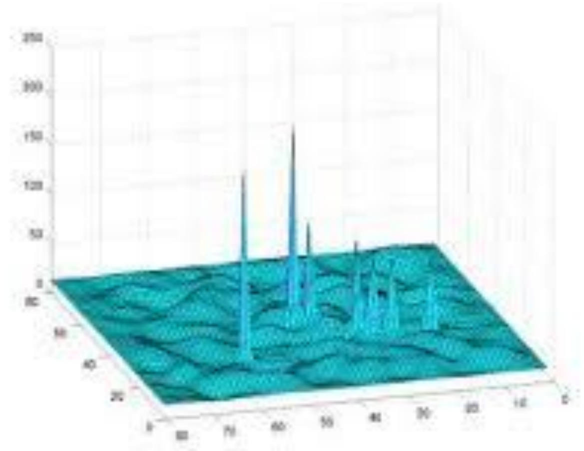


Simplex algorithm solves LPs efficiently (explains practice).

[Spielman & Teng 2000]

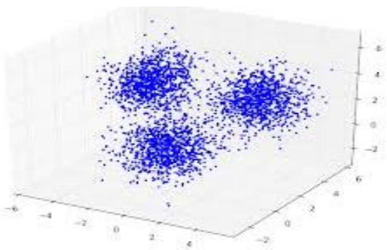
Smoothed analysis guarantees:

- Worst instances are isolated
- Small random perturbation of input makes instances easy
- Best polytime guarantees in the absence of any worst-case guarantees

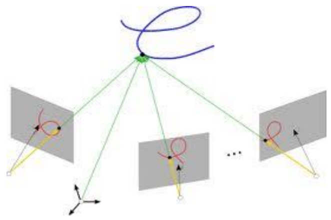


Today's talk: Smoothed Analysis for Learning [BCMV STOC'14]

- First Smoothed Analysis treatment for Unsupervised Learning



Mixture of Gaussians



Multiview models

Thm. Polynomial time algorithms for learning axis-aligned Gaussians, Multview models etc. *even in ``overcomplete settings''*.

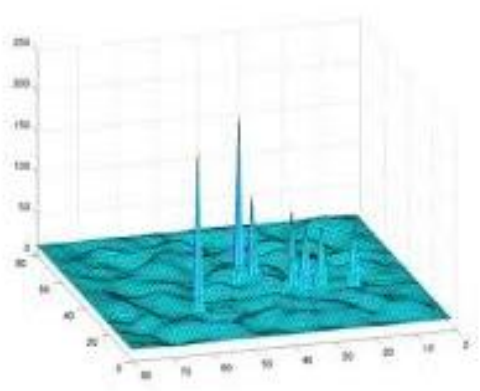
based on

Thm. Polynomial time algorithms for tensor decompositions in smoothed analysis setting.

Smoothed Analysis for Learning

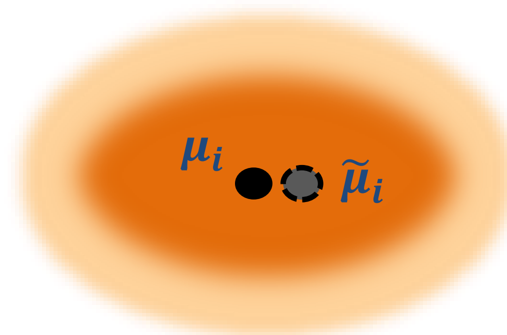
Learning setting (e.g. Mixtures of Gaussians)

Worst-case instances: Means $\{\boldsymbol{\mu_i}\}$ in pathological configurations



Means not in adversarial configurations in real-world!

What if means $\{\boldsymbol{\mu_i}\}$ perturbed slightly ?

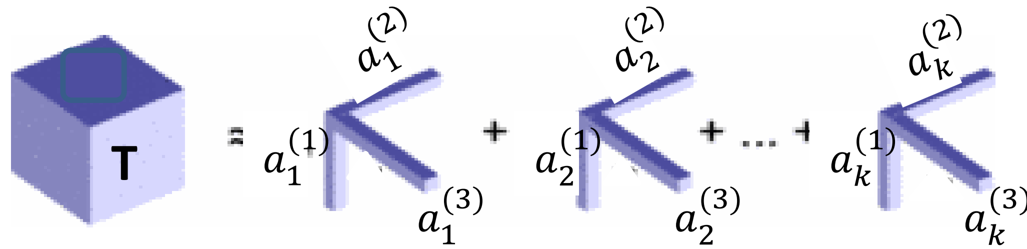


Generally, parameters of the model are perturbed slightly.

Smoothed Analysis for Tensor Decompositions

Factors of the Decomposition are perturbed

1. Adversary chooses tensor



$$T_{d\times d\times\cdots\times d} = \sum_{i=1}^{k} a_i^{(1)} \otimes a_i^{(2)} \otimes \ldots \otimes a_i^{(t)}$$

2. $\tilde{a}_i^{(j)}$ is random ρ-perturbation of $a_i^{(j)}$

i.e. add independent (gaussian) random vector of length $\approx \rho$.

3. Input: $\tilde{T}$. Analyse algorithm on $\tilde{T}$.

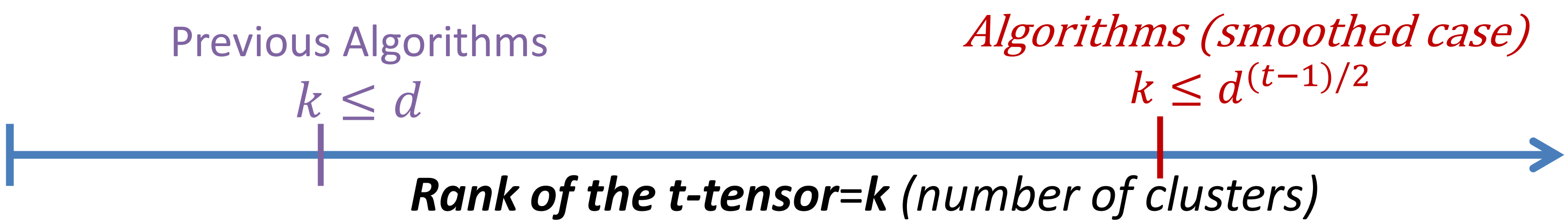
$$\tilde{T} = \sum_{i=1}^{k} \tilde{a}_i^{(1)} \otimes \tilde{a}_i^{(2)} \otimes \ldots \otimes \tilde{a}_i^{(t)} + \text{noise}$$

Algorithmic Guarantees

Thm [BCMV'14]*. Polynomial time algorithm** for decomposing t-tensor (d-dim) in smoothed analysis model when ***rank $\boldsymbol{k \leq d^{(t-1)/2}}$ w.h.p.

Running time, sample complexity = $\boldsymbol{poly_t\left(d, k, \frac{1}{\rho}\right)}$.

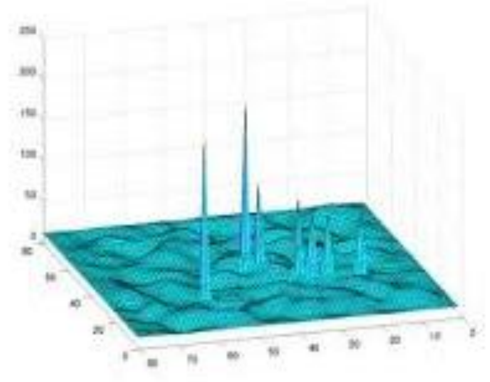
<u>Guarantees for order-t tensors in d-dims (each)</u>



Corollary. Polytime algorithms (smoothed analysis) for Mixtures of axis-aligned Gaussians, Multiview models etc. even in overcomplete setting i.e. no. of clusters $\mathrm{k} \leq \mathrm{dim}^{\mathrm{C}}$ for any constant C w.h.p.

Interpreting Smoothed Analysis Guarantees

Time, sample complexity = $poly_t\left(d, k, \frac{1}{\rho}\right)$.



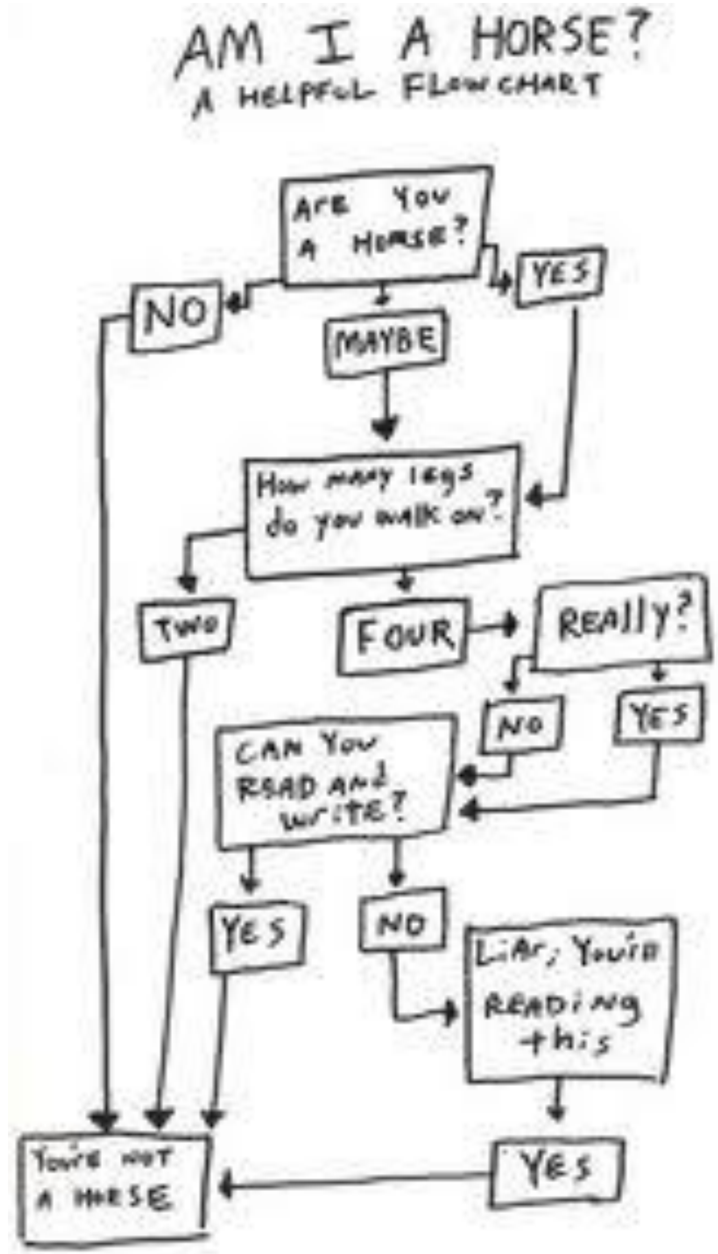
Works with probability $1\text{-}exp(-\rho d^{3^{-t}})$

- Exponential small failure probability (for constant order t)

Smooth Interpolation between Worst-case and Average-case

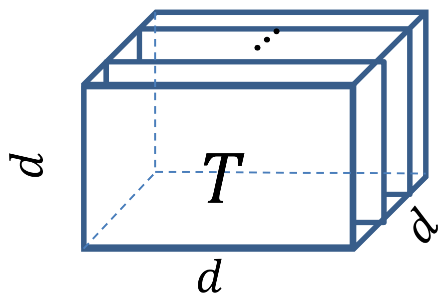
- $\rho = 0$: worst-case
- ρ is large: almost random vectors.
- Can handle ρ inverse-polynomial in d, k

Algorithm Details

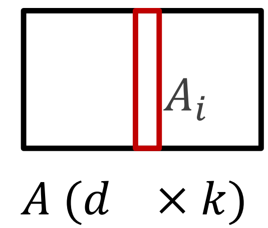


Algorithm Outline

1. An algorithm for 3-tensors in the ``full rank setting'' ($\mathbf{k} \leq d$).



<u>Recall:</u> $T = \sum_{i=1}^{k} A_i \otimes B_i \otimes C_i$



<u>Aim:</u> *Recover A, B, C*

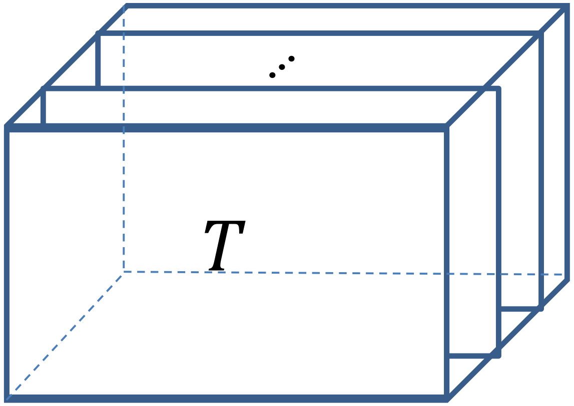
> **[Jennrich 70] A simple (robust) algorithm for 3-tensor T when:**
> $\sigma_k(A), \sigma_k(B), \sigma_2(C) \geq 1/poly(d, k)$

- Any algorithm for full-rank (non-orthogonal) tensors suffices

2. For higher order tensors using ``*tensoring / flattening*''.

- Helps handle the over-complete setting ($\mathrm{k} \gg d$)

Blast from the Past



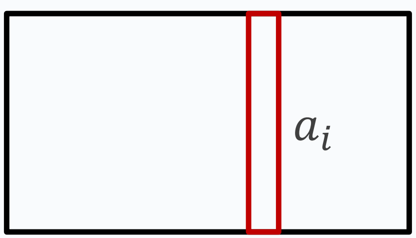
[Jennrich via Harshman 70]

Algorithm for 3-tensor $T = \sum_{i=1}^{k} a_i \otimes b_i \otimes c_i$

- A, B are full rank (rank=k)
- C has rank $\geq$2
- Reduces to matrix eigen-decompositions

Recall

$$\boldsymbol{T} \approx_{\epsilon} \sum_{i=1}^{k} a_i \otimes b_i \otimes c_i$$



Aim: *Recover A, B, C*

Qn. Is this algorithm robust to errors ?

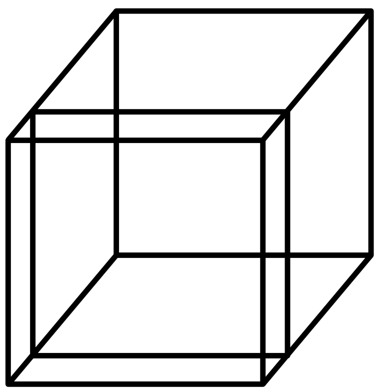
Yes ! Needs perturbation bounds for eigenvectors. [Stewart-Sun]

Thm. Efficiently decompose T$=_{\epsilon} \sum_{i=1}^{k} a_i \otimes b_i \otimes c_i$ and recover A, B, C upto $\epsilon.poly(d,k)$ error

when 1) *A, B are min-singular-value ≥ 1/poly(d)*

2) *C doesn't have parallel columns.*

Slices of tensors



Consider rank 1 tensor $x \otimes y \otimes z$

s'th slice: $y_s \cdot (x \otimes z)$

$$T = \sum_{i=1}^{k} a_i \otimes b_i \otimes c_i$$

s'th slice: $\sum_{i=1}^{k} b_i(s).(a_i \otimes c_i)$

All slices have a common diagonalization (A, C)!

Random combination w of slices: $\sum_{i=1}^{k} \langle b_i, w \rangle .(a_i \otimes c_i)$

Simultaneous diagonalization

Two matrices with common diagonalization (X, Y)

$$M_1 = X D_1 Y^T$$

$$M_2 = X D_2 Y^T$$

$$M_1 M_2^{-1} = X D_1 D_2^{-1} X^{-1}$$

If 1) X, Y are invertible and

2) D_1, D_2 have unequal non-zero entries,

We can find X, Y by matrix diagonalization!

Decomposition algorithm [Jennrich]

$$T \approx_{\epsilon} \sum_{i=1}^{k} a_i \otimes b_i \otimes c_i$$

Algorithm:

1. Take random combination along w_1 as M_1.
2. Take random combination along w_2 as M_2.
3. Find eigen-decomposition of $M_1 M_2^{\dagger}$ to get A. Similarly B,C.

Thm. Efficiently decompose T$=_{\epsilon} \sum_{i=1}^{k} a_i \otimes b_i \otimes c_i$ and recover A, B, C up to $\epsilon . poly(d, k)$ error (in Frobenius norm) when

1) *A, B are full rank i.e. min-singular-value $\geq$ 1/poly(d)*

2) *C doesn't have parallel columns (in a robust sense).*

Overcomplete Case

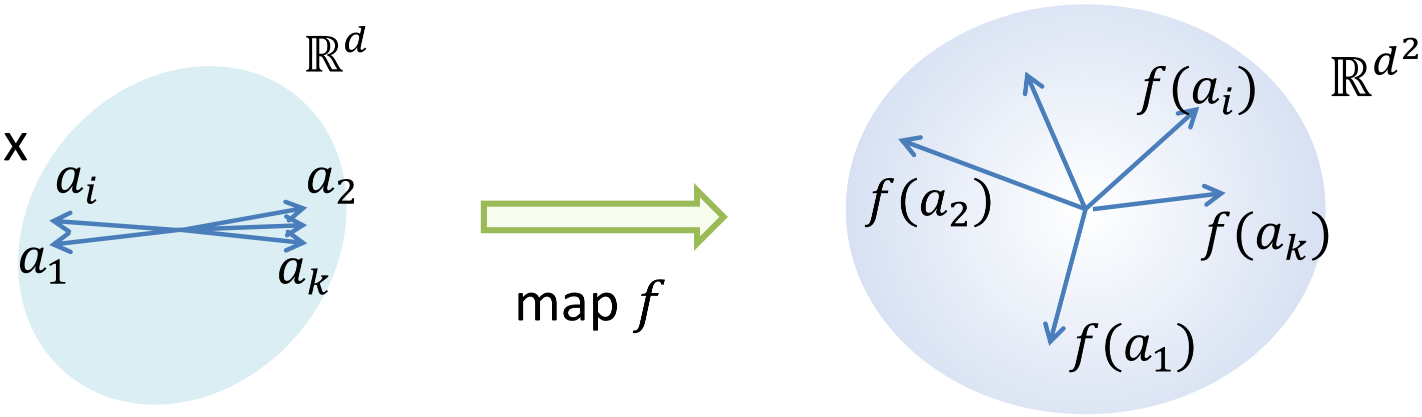
into Techniques

Mapping to Higher Dimensions

How do we handle the case rank $k = \Omega(d^2)$?

(or even vectors with "many" linear dependencies?)

Factor matrix A



***f* maps parameter/factor vectors to higher dimensions s.t.**

1. Tensor corresponding to map f computable using the data x
2. $f(a_1), f(a_2), \ldots, f(a_k)$ are linearly independent (min singular value)

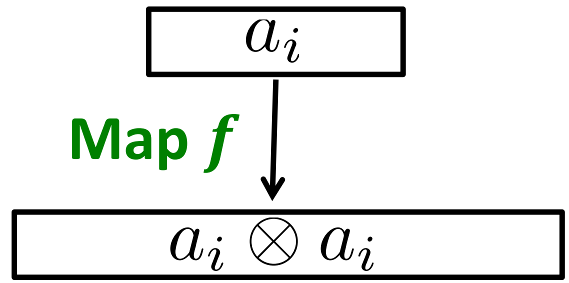
- Reminiscent of Kernels in SVMs

A mapping to higher dimensions

Outer product / Tensor products:

Map $f(a_i) = a_i \otimes a_i$

- Tensor is $E[x^{\otimes 2} \otimes x^{\otimes 2} \otimes x^{\otimes 2}]$



Basic Intuition:

1. $a_i \otimes a_i$ has d^2 dimensions.

2. For non-parallel unit vectors a_i and a_j, distance increases:

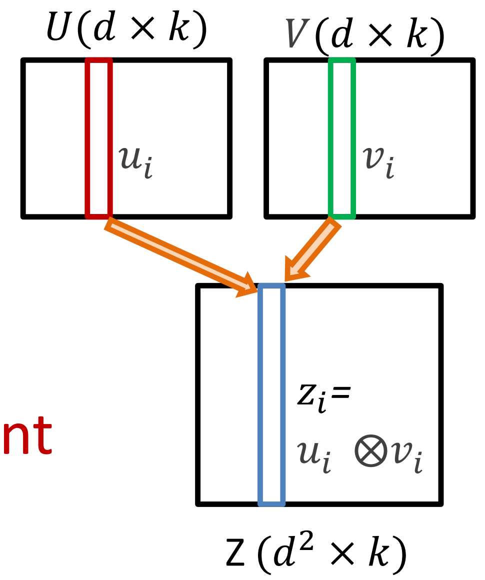
$$\langle a_i \otimes a_i \,, a_j \otimes a_j \rangle = \langle a_i, a_j \rangle^2 < |\langle a_i, a_j \rangle|$$

Qn: are *these* vectors $a_i \otimes a_i$ linearly independent?

Is ``essential dimension'' $\Omega(d^2)$?

Bad cases

U, V have rank=d. Vectors $z_i = u_i \otimes v_i \in \mathbb{R}^{d^2}$



Lem. Dimension (K-rank) under tensoring is **additive.**

Bad example where $k > 2d$:

- Every d vectors of U and V are linearly independent
- But $(2d - 1)$ vectors of Z are linearly dependent !

Strategy does not work in the worst-case

But, bad examples are pathological and hard to construct!

Beyond Worst-case analysis

Can we hope for "dimension" to multiply "typically"?

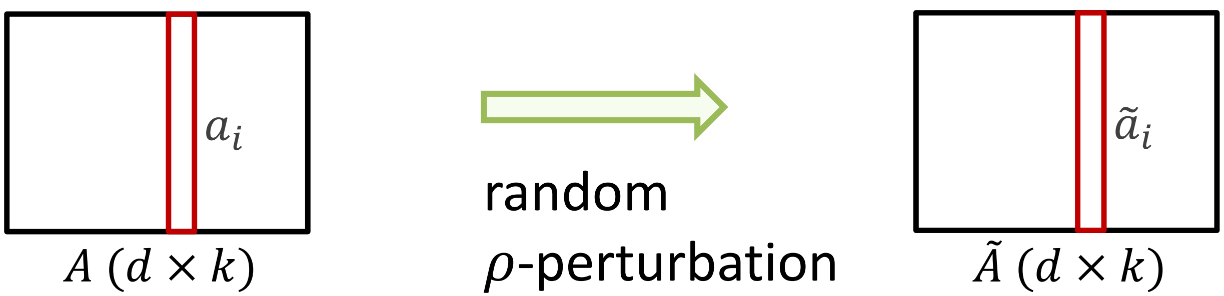
Product vectors & linear structure

Map $\boldsymbol{f(a_i) = a_i^{\otimes t}}$

- Easy to compute tensor with $f(a_i)$ as factors / parameters

  (``Flattening'' of 3t-order moment tensor)

- New factor matrix is full rank using ***Smoothed Analysis.***



Theorem. For any matrix $A_{d\times k}$, for $k < d^t/2$,

$$\sigma_k(\tilde{A}) \geq 1/poly\left(k, d, \frac{1}{\rho}\right)$$ with probability *1- exp(-poly(d)).*

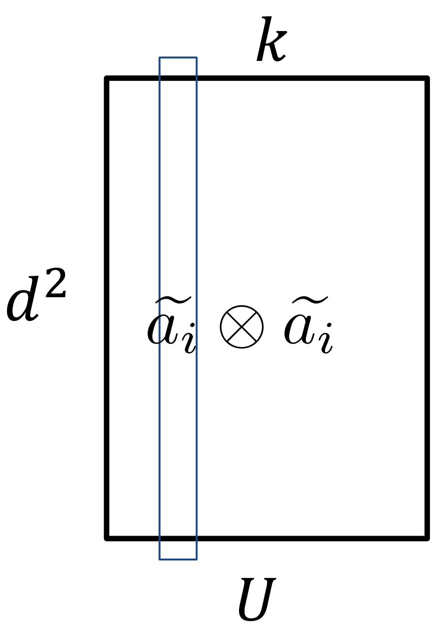
Proof sketch (t=2)

Prop. For any matrix A, matrix U below $(k < d^2/2)$ has

$$\sigma_k(\tilde{A}) \geq 1/poly\left(k, d, \frac{1}{\rho}\right)$$ with probability *1- exp(-poly(d)).*

$$a_i \rightarrow \widetilde{a}_i$$

$$\widetilde{a}_i = a_i + \varepsilon_i$$



Main Issue: perturbation *before* product..

- easy if columns perturbed after tensor product (simple anti-concentration bounds)
- only $2d$ bits of randomness in d^2 dims
- Block dependencies

Technical component

show perturbed product vectors behave like random vectors in R^{d^2}

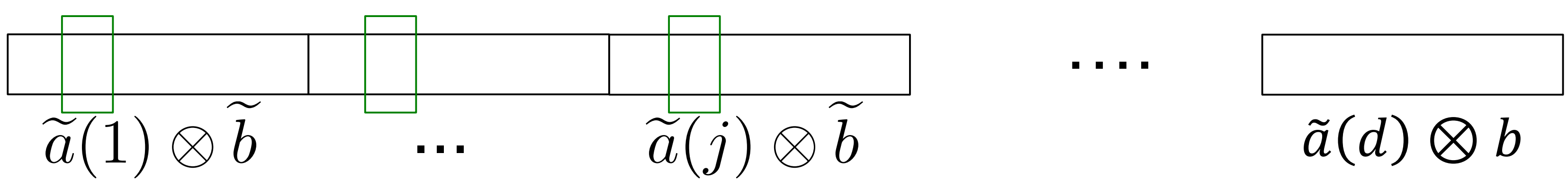
Projections of product vectors

Question. Given any vector $a \in \mathbb{R}^d$ and gaussian ρ-perturbation $\tilde{a} = a + \epsilon$, does $\tilde{\boldsymbol{a}} \otimes \tilde{\boldsymbol{a}}$ have projection $poly(\rho, \frac{1}{d})$ onto *any* given $d^2/2$ dimensional subspace $S \subset R^{d^2}$ with prob. $1 - \exp(-\sqrt{d})$?

anti-concentration for polynomials implies this with probability 1-1/poly

Easy : Take d^2 dimensional x, ρ-perturbation to x will have projection $> 1/poly(\rho)$ on to S w.h.p.

Much tougher for product of perturbations!
(inherent block structure)

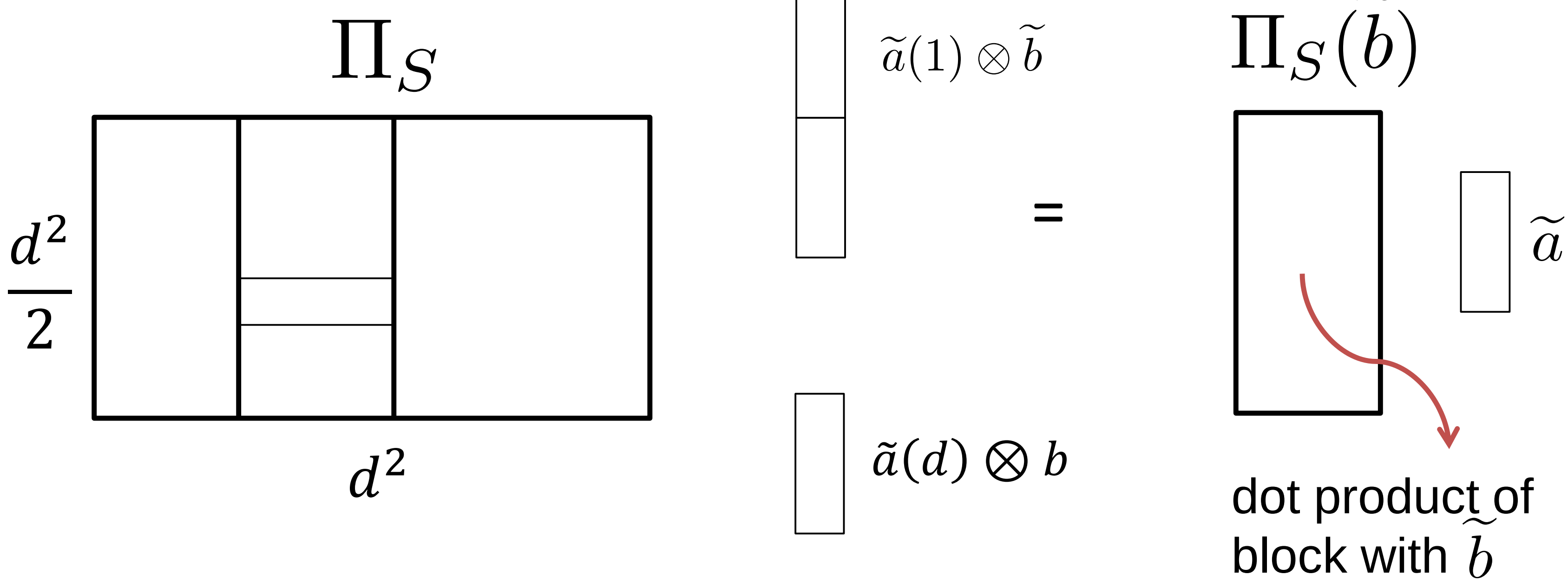


Projections of product vectors

Question. Given any vector $a, b \in \mathbb{R}^d$ and gaussian ρ-perturbation $\tilde{a}, \tilde{b}$, does $\tilde{\boldsymbol{a}} \otimes \tilde{\boldsymbol{b}}$ have projection $poly(\rho, \frac{1}{d})$ onto *any* given $d^2/2$ dimensional subspace $S \subset R^{d^2}$ with prob. $1 - \exp(-\sqrt{d})$?

Π_S is projection matrix onto S

$\Pi_S(x)$ is a $\frac{d^2}{2} \times d$ matrix



Two steps of Proof..

1. W.h.p. (over perturbation of *b*), $\Pi_S(\tilde{b})$ has at least r eigenvalues $> poly(\rho, \frac{1}{d})$

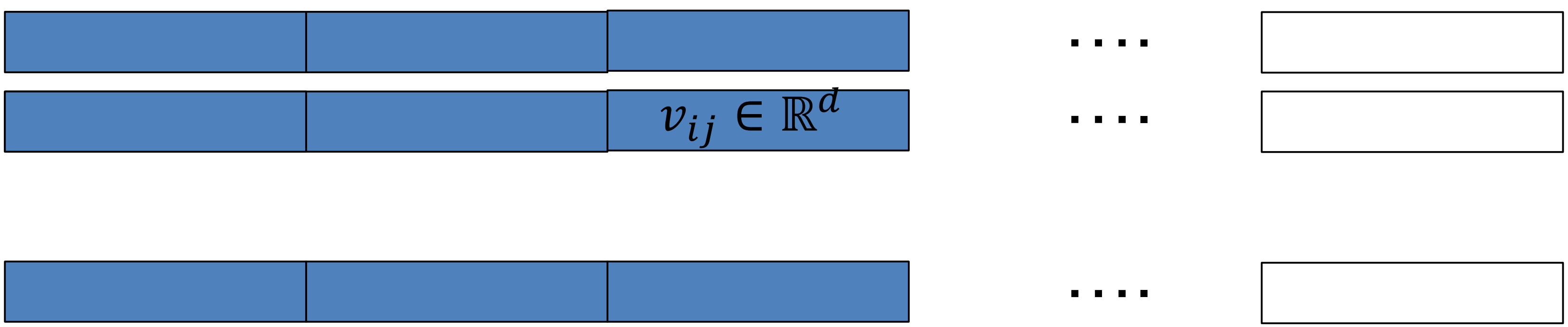
will show with $r = \sqrt{d}$

2. If $\Pi_S(\tilde{b})$ has r eigenvalues $> poly(\rho, \frac{1}{d})$, then w.p. $1 - \exp(-r)$ (over perturbation of $\tilde{a}$), $\tilde{\boldsymbol{a}} \otimes \tilde{\boldsymbol{b}}$ has large projection onto S.

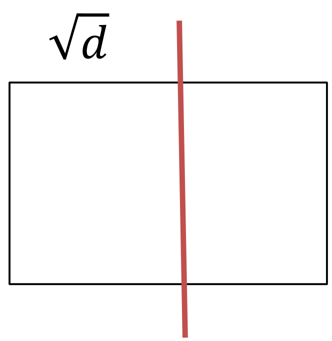
follows easily analyzing projection of a vector to a dim-*k* space

Structure in any subspace S

Suppose: Choose Π_S first $\sqrt{d} \times \sqrt{d}$ “blocks” in Π_S were orthogonal...



$$\Pi_S(\tilde{b})|_{\sqrt{d}} =$$



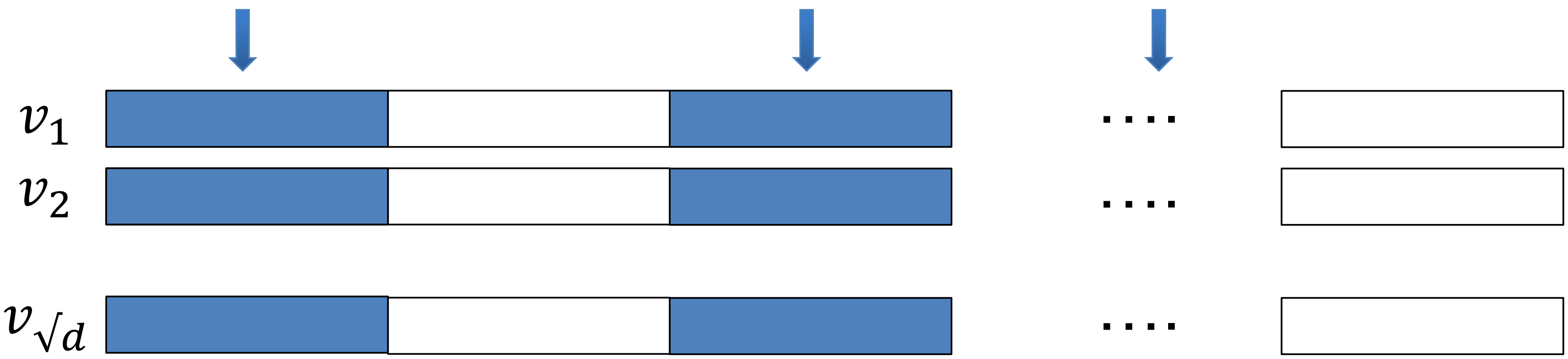
(restricted to $\sqrt{d}$ cols)

- Entry (i,j) is: $\langle v_{i,j}, b + \varepsilon \rangle$
- Translated i.i.d. Gaussian matrix!

has many big eigenvalues

Finding Structure in any subspace S

Main claim: every $c.d^2$ dimensional space S has $\sim\sqrt{d}$ vectors with such a structure..



Property: picked blocks (*d* dim vectors) have "reasonable" component orthogonal to span of rest..

Earlier argument goes through even with blocks not fully orthogonal!

Main claim (sketch)..

crucially use the fact that we have a $\Omega(d^2)$ dim subspace

Idea: obtain "good" columns one by one..

- Show there exists a block with many linearly independent "choices"
- Fix some choices and argue the same property holds, …

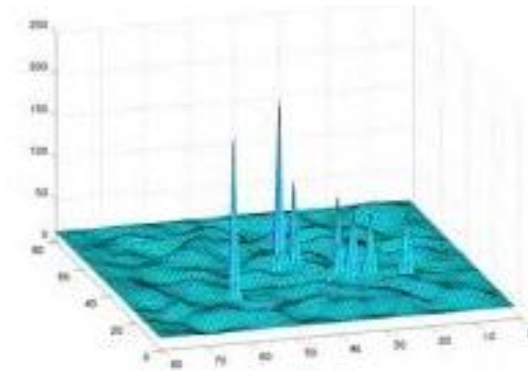
Q.E.D.

Generalization: similar result holds for higher order products, implies main result.

- Uses a delicate inductive argument

Summary

- **Smoothed Analysis for Learning Probabilistic models.**



- **Polynomial time Algorithms in Overcomplete settings:**

Guarantees for order-t tensors in d-dims (each)

Previous Algorithms

$k \leq d$

Algorithms (smoothed case)

$k \leq d^{(t-1)/2}$

Rank of the t-tensor=k *(number of clusters)*

- **Flattening gets beyond full-rank conditions:**
  Plug into results on Spectral Learning of Probabilistic models

Future Directions

Better Robustness to Errors

- Modelling errors?
- Tensor decomposition algorithms that more robust to errors ?

  promise: [Barak-Kelner-Steurer'14] using Lasserre hierarchy

Better dependence on rank k vs dim d (esp. 3 tensors)

- Next talk by Anandkumar: Random/ Incoherent decompositions

Better guarantees using Higher-order moments

- Better bounds w.r.t. smallest singular value ?

Smoothed Analysis for other Learning problems ?

Thank You!

Questions?