# Learning Latent Variable Models through Tensor Methods

**Anima Anandkumar**

U.C. Irvine

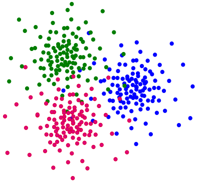
# Challenges in Unsupervised Learning

- Learn a latent variable model without labeled examples.
- E.g. topic models, hidden Markov models, Gaussian mixtures, community detection.

- Maximum likelihood is NP-hard in most scenarios.
- Practice: EM, Variational Bayes have no consistency guarantees.
- Efficient computational and sample complexities?

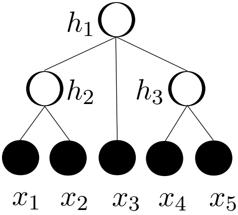
In this talk: guaranteed and efficient learning through tensor methods

# How to model hidden effects?



Basic Approach: mixtures/clusters

- Hidden variable $h$ is categorical.




Advanced: Probabilistic models

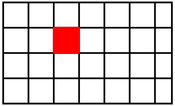
- Hidden variable $h$ has more general distributions.
- Can model mixed memberships.

# Moment Based Approaches

## Multivariate Moments

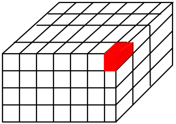
$$M_1 := \mathbb{E}[x], \quad M_2 := \mathbb{E}[x \otimes x], \quad M_3 := \mathbb{E}[x \otimes x \otimes x].$$

## Matrix



- $\mathbb{E}[x \otimes x] \in \mathbb{R}^{d \times d}$ is a second order tensor.
- $\mathbb{E}[x \otimes x]_{i_1, i_2} = \mathbb{E}[x_{i_1} x_{i_2}]$.
- For matrices: $\mathbb{E}[x \otimes x] = \mathbb{E}[x x^\top]$.

## Tensor



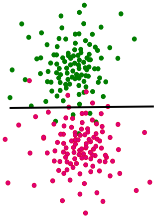
- $\mathbb{E}[x \otimes x \otimes x] \in \mathbb{R}^{d \times d \times d}$ is a third order tensor.
- $\mathbb{E}[x \otimes x \otimes x]_{i_1, i_2, i_3} = \mathbb{E}[x_{i_1} x_{i_2} x_{i_3}]$.

# Outline

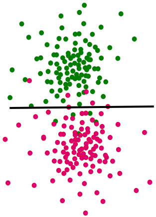
# Classical Spectral Methods: Matrix PCA

## Learning through Spectral Clustering

- Dimension reduction through PCA (on data matrix)
- Clustering on projected vectors (e.g. $k$-means).

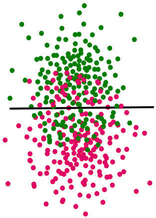


# Classical Spectral Methods: Matrix PCA



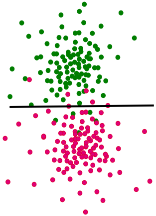
## Learning through Spectral Clustering

- Dimension reduction through PCA (on data matrix)
- Clustering on projected vectors (e.g. $k$-means).



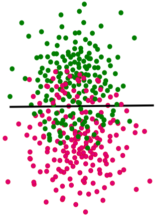
- Basic method works only for single memberships.
- Failure to cluster under small separation.
- Require long documents for good concentration bounds.

# Classical Spectral Methods: Matrix PCA



## Learning through Spectral Clustering

- Dimension reduction through PCA (on data matrix)
- Clustering on projected vectors (e.g. $k$-means).



- Basic method works only for single memberships.
- Failure to cluster under small separation.
- Require long documents for good concentration bounds.

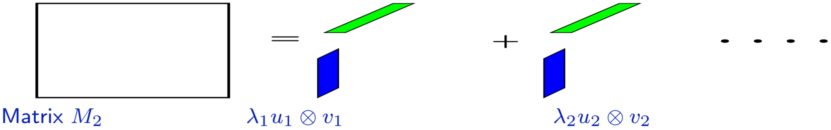
Efficient Learning Without Separation Constraints?

# Beyond SVD: Spectral Methods on Tensors

- How to learn the mixture components without separation constraints?
  - Are higher order moments helpful?
- Unified framework?
  - Moment-based Estimation of probabilistic latent variable models?
- SVD gives spectral decomposition of matrices.
  - What are the analogues for tensors?

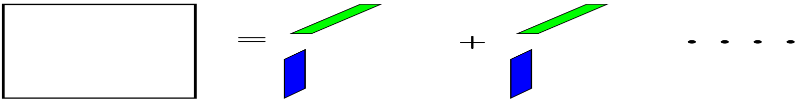
# Spectral Decomposition

$$\boxed{M_2 = \sum_i \lambda_i u_i \otimes v_i}$$

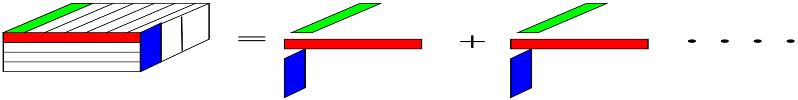


# Spectral Decomposition

$$\boxed{M_2 = \sum_i \lambda_i u_i \otimes v_i}$$




$$\boxed{M_3 = \sum_i \lambda_i u_i \otimes v_i \otimes w_i}$$




- $u \otimes v \otimes w$ is a rank-1 tensor since its $(i_1, i_2, i_3)^{\text{th}}$ entry is $u_{i_1} v_{i_2} w_{i_3}$.

# Decomposition of Orthogonal Tensors

- $A$ has orthogonal columns.

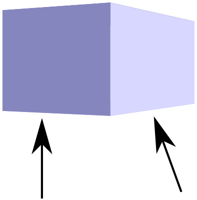
$$M_3 = \sum_i w_i a_i \otimes a_i \otimes a_i.$$

# Decomposition of Orthogonal Tensors

- $A$ has orthogonal columns.

$$M_3 = \sum_i w_i a_i \otimes a_i \otimes a_i.$$

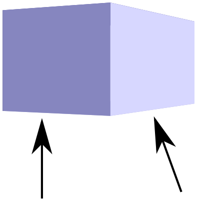
- $M_3(I, a_1, a_1) = \sum_i w_i \langle a_i, a_1 \rangle^2 a_i = w_1 a_1.$



# Decomposition of Orthogonal Tensors

- $A$ has orthogonal columns.

$$M_3 = \sum_i w_i a_i \otimes a_i \otimes a_i.$$

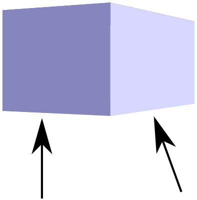


- $M_3(I, a_1, a_1) = \sum_i w_i \langle a_i, a_1 \rangle^2 a_i = w_1 a_1.$
- $a_i$ are eigenvectors of tensor $M_3$.
- Analogous to matrix eigenvectors: $Mv = M(I, v) = \lambda v.$

# Decomposition of Orthogonal Tensors

- $A$ has orthogonal columns.

$$M_3 = \sum_i w_i a_i \otimes a_i \otimes a_i.$$



- $M_3(I, a_1, a_1) = \sum_i w_i \langle a_i, a_1 \rangle^2 a_i = w_1 a_1.$
- $a_i$ are eigenvectors of tensor $M_3$.
- Analogous to matrix eigenvectors: $Mv = M(I, v) = \lambda v$.

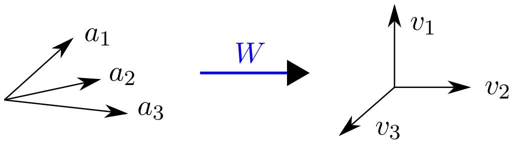
## Two Problems

- How to find eigenvectors of a tensor?
- $A$ is not orthogonal in general.

# Whitening

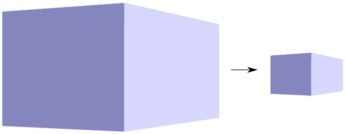
$$M_3 = \sum_i w_i a_i \otimes a_i \otimes a_i, \quad M_2 = \sum_i w_i a_i \otimes a_i.$$

- Find whitening matrix $W$ s.t. $W^\top A = V$ is an orthogonal matrix.
- When $A \in \mathbb{R}^{d \times k}$ has full column rank, it is an invertible transformation.




- Use pairwise moments $M_2$ to find $W$ s.t. $W^\top M_2 W = I$.
- Eigen-decomposition of $M_2 = U\mathrm{Diag}(\tilde{\lambda})U^\top$, then $W = U\mathrm{Diag}(\tilde{\lambda}^{-1/2})$.

# Using Whitening to Obtain Orthogonal Tensor



Tensor $M_3$ Tensor $T$

## Multi-linear transform

- $M_3 \in \mathbb{R}^{d\times d\times d}$ and $T \in \mathbb{R}^{k\times k\times k}$.
- $T = M_3(W, W, W) = \sum_i w_i (W^\top a_i)^{\otimes 3}$.
- $T = \sum_{i\in[k]} \lambda_i \cdot v_i \otimes v_i \otimes v_i$ is orthogonal.
- Dimensionality reduction when $k \ll d$.

# Putting it together

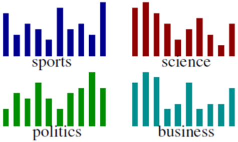
$$M_2 = \sum_i w_i a_i \otimes a_i, \quad M_3 = \sum_i w_i a_i \otimes a_i \otimes a_i.$$

- Obtain whitening matrix $W$ from SVD of $M_2$.
- Use $W$ for multi-linear transform: $T = M_3(W, W, W)$.
- Find eigenvectors of $T$ through power method and deflation.

For what models can we obtain $M_2$ and $M_3$ forms?

# Outline

# Topic Modeling

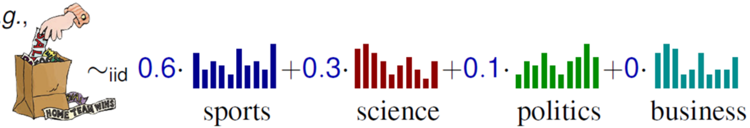


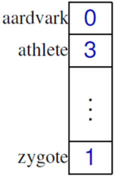

$k$ topics (distributions over vocab words).

Each document $\leftrightarrow$ mixture of topics.

Words in document $\sim_{\text{iid}}$ mixture dist.

*E.g.*,




$\Pr_\theta[\text{"play"} \mid \text{sports}] = 0.0002$

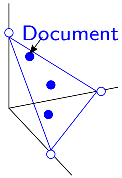
$\Pr_\theta[\text{"game"} \mid \text{sports}] = 0.0003$

$\Pr_\theta[\text{"season"} \mid \text{sports}] = 0.0001$

$\vdots$

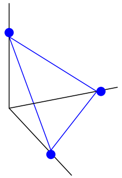
# Geometric Picture for Topic Models

Topic proportions vector $(h)$



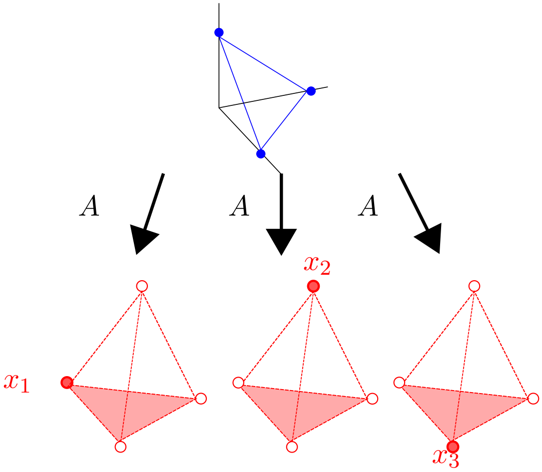
# Geometric Picture for Topic Models

Single topic $(h)$



# Geometric Picture for Topic Models

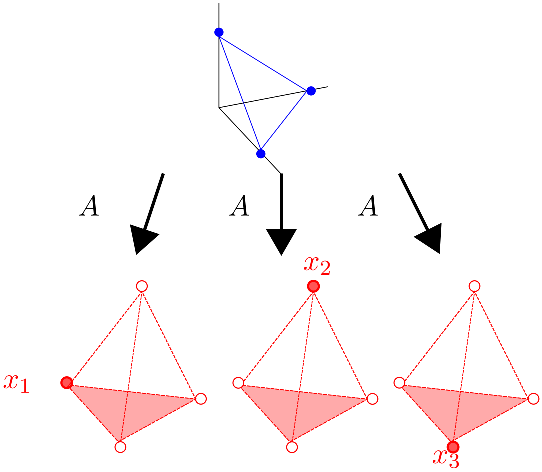
Single topic $(h)$




Word generation $(x_1, x_2, \ldots)$

# Geometric Picture for Topic Models

Single topic $(h)$

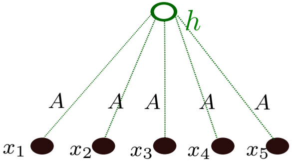



Word generation $(x_1, x_2, \ldots)$

- Linear model: $\boxed{\mathbb{E}[x_i|h] = Ah}$.

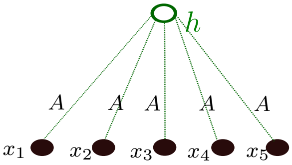
# Moments for Single Topic Models

- $\boxed{\mathbb{E}[x_i|h] = Ah.}$
- $\boxed{w := \mathbb{E}[h].}$
- Learn topic-word matrix $A$, vector $w$



# Moments for Single Topic Models

- $\boxed{\mathbb{E}[x_i|h] = Ah.}$
- $\boxed{w := \mathbb{E}[h].}$
- Learn topic-word matrix $A$, vector $w$




## Pairwise Co-occurence Matrix $M_x$

$$M_2 := \mathbb{E}[x_1 \otimes x_2] = \mathbb{E}[\mathbb{E}[x_1 \otimes x_2|h]] = \sum_{i=1}^{k} w_i a_i \otimes a_i$$

## Triples Tensor $M_3$

$$M_3 := \mathbb{E}[x_1 \otimes x_2 \otimes x_3] = \mathbb{E}[\mathbb{E}[x_1 \otimes x_2 \otimes x_3|h]] = \sum_{i=1}^{k} w_i a_i \otimes a_i \otimes a_i$$

# Moments under LDA

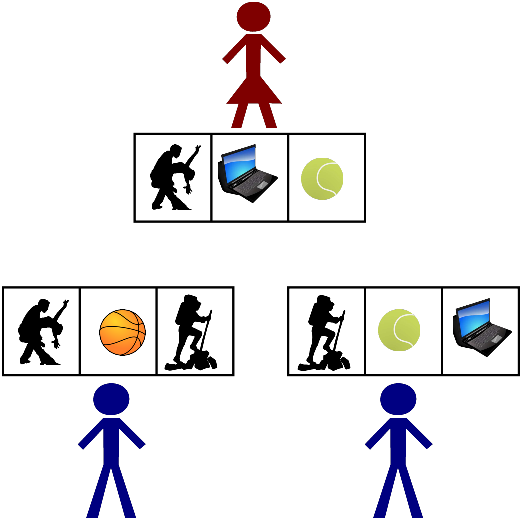
$$
\begin{aligned}
M_2 &:= \mathbb{E}[x_1 \otimes x_2] - \frac{\alpha_0}{\alpha_0 + 1}\mathbb{E}[x_1] \otimes \mathbb{E}[x_1] \\
M_3 &:= \mathbb{E}[x_1 \otimes x_2 \otimes x_3] - \frac{\alpha_0}{\alpha_0 + 2}\mathbb{E}[x_1 \otimes x_2 \otimes \mathbb{E}[x_1]] - \text{more stuff...}
\end{aligned}
$$

Then

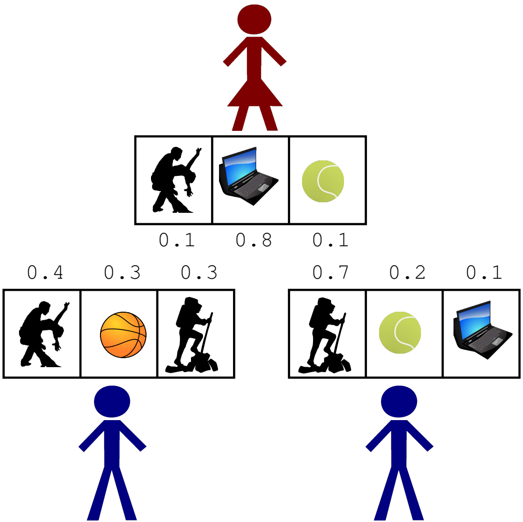
$$
\begin{aligned}
M_2 &= \sum \tilde{w}_i \; a_i \otimes a_i \\
M_3 &= \sum \tilde{w}_i \; a_i \otimes a_i \otimes a_i .
\end{aligned}
$$

- Three words per document suffice for learning LDA.
- Similar forms for HMM, ICA, etc.

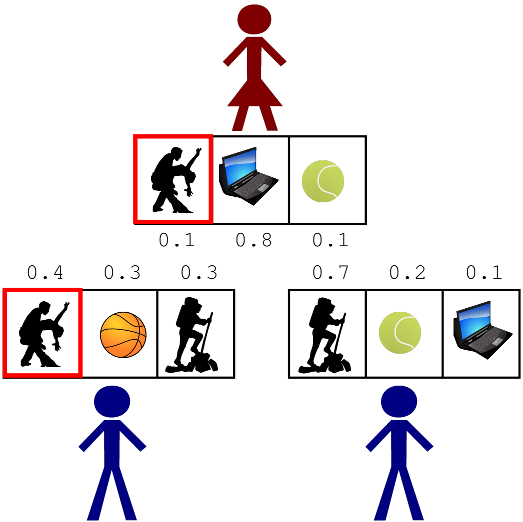
# Network Community Models



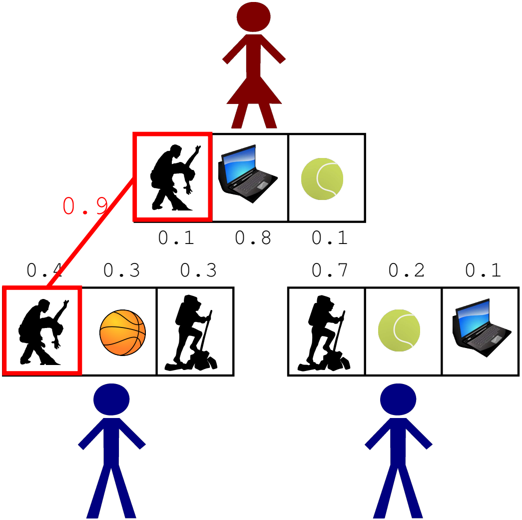
# Network Community Models



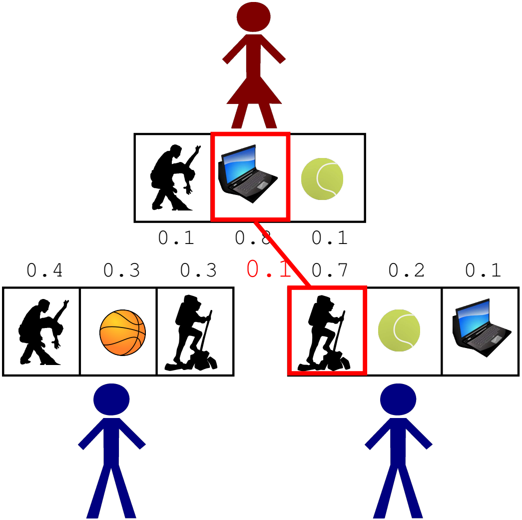
# Network Community Models



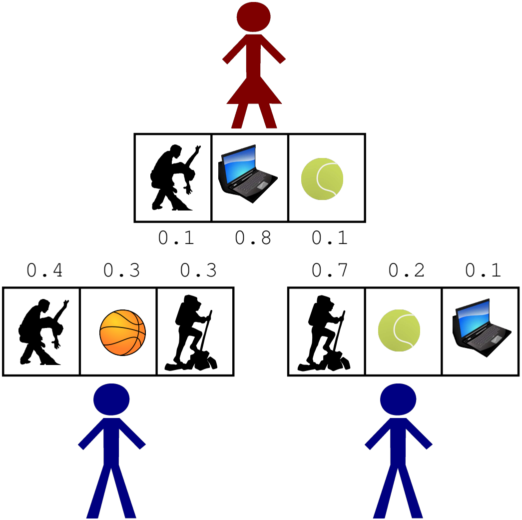
# Network Community Models



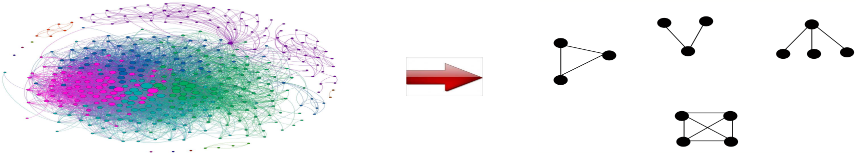
# Network Community Models



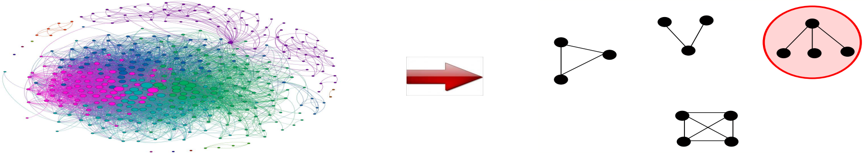
# Network Community Models



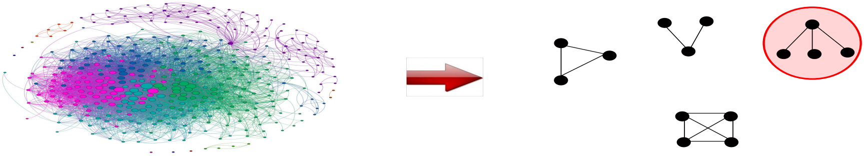
# Subgraph Counts as Graph Moments



# Subgraph Counts as Graph Moments

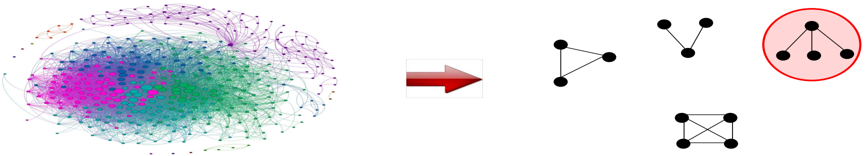


# Subgraph Counts as Graph Moments



$3$-star counts sufficient for identifiability and learning of MMSB

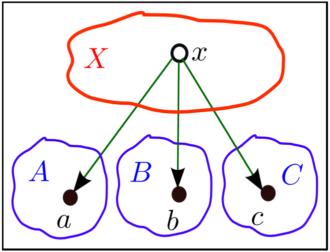
# Subgraph Counts as Graph Moments



3-star counts sufficient for identifiability and learning of MMSB

## 3-Star Count Tensor

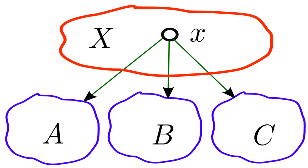
$$\tilde{M}_3(a,b,c) = \frac{1}{|X|}\#\text{ of common neighbors in } X$$
$$= \frac{1}{|X|}\sum_{x\in X} G(x,a)G(x,b)G(x,c).$$
$$\tilde{M}_3 = \frac{1}{|X|}\sum_{x\in X}\left[G_{x,A}^\top \otimes G_{x,B}^\top \otimes G_{x,C}^\top\right]$$



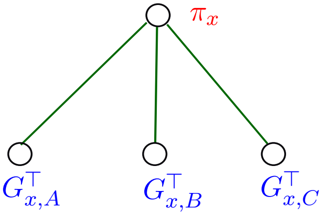
# Multi-view Representation

- Conditional independence of the three views
- $\pi_x$: community membership vector of node $x$.

3-stars




Graphical model




Similar form as $M_2$ and $M_3$ for topic models

# Main Results

- $k$ communities, $n$ nodes. Uniform communities.
- $\alpha_0$: Sparsity level of community memberships (Dirichlet parameter).
- $p, q$: intra/inter-community edge density.

## Scaling Requirements

$$\boxed{n = \tilde{\Omega}(k^2(\alpha_0 + 1)^3), \qquad \frac{p-q}{\sqrt{p}} = \tilde{\Omega}\left(\frac{(\alpha_0+1)^{1.5}k}{\sqrt{n}}\right).}$$

---

"A Tensor Spectral Approach to Learning Mixed Membership Community Models" by A. Anandkumar, R. Ge, D. Hsu, and S.M. Kakade. COLT 2013.

# Main Results

- $k$ communities, $n$ nodes. Uniform communities.
- $\alpha_0$: Sparsity level of community memberships (Dirichlet parameter).
- $p, q$: intra/inter-community edge density.

## Scaling Requirements

$$\boxed{n = \tilde{\Omega}(k^2(\alpha_0+1)^3), \qquad \frac{p-q}{\sqrt{p}} = \tilde{\Omega}\left(\frac{(\alpha_0+1)^{1.5}k}{\sqrt{n}}\right).}$$

- For stochastic block model $(\alpha_0 = 0)$, tight results
- Tight guarantees for sparse graphs (scaling of $p, q$)
- Tight guarantees on community size: require at least $\sqrt{n}$ sized communities
- Efficient scaling w.r.t. sparsity level of memberships $\alpha_0$

---

"A Tensor Spectral Approach to Learning Mixed Membership Community Models" by A. Anandkumar, R. Ge, D. Hsu, and S.M. Kakade. COLT 2013.

# Main Results (Contd)

- $\alpha_0$: Sparsity level of community memberships (Dirichlet parameter).
- $\Pi$: Community membership matrix, $\Pi^{(i)}$: $i^{\text{th}}$ community
- $\widehat{S}$: Estimated supports, $\widehat{S}(i,j)$: Support for node $j$ in community $i$.

## Norm Guarantees

$$\boxed{\frac{1}{n}\max_i \|\widehat{\Pi}^i - \Pi^i\|_1 = \tilde{O}\left(\frac{(\alpha_0+1)^{3/2}\sqrt{p}}{(p-q)\sqrt{n}}\right)}$$

# Main Results (Contd)

- $\alpha_0$: Sparsity level of community memberships (Dirichlet parameter).
- $\Pi$: Community membership matrix, $\Pi^{(i)}$: $i^{\text{th}}$ community
- $\widehat{S}$: Estimated supports, $\widehat{S}(i,j)$: Support for node $j$ in community $i$.

## Norm Guarantees

$$\boxed{\frac{1}{n}\max_i \|\widehat{\Pi}^i - \Pi^i\|_1 = \tilde{O}\left(\frac{(\alpha_0+1)^{3/2}\sqrt{p}}{(p-q)\sqrt{n}}\right)}$$

## Support Recovery

$\exists\, \xi$ s.t. for all nodes $j \in [n]$ and all communities $i \in [k]$, w.h.p

$$\boxed{\Pi(i,j) \geq \xi \Rightarrow \widehat{S}(i,j) = 1 \quad \text{and} \quad \Pi(i,j) \leq \frac{\xi}{2} \Rightarrow \widehat{S}(i,j) = 0.}$$

Zero-error Support Recovery of Significant Memberships of All Nodes

# Outline

# Computational Complexity ($k \ll n$)

- $n = \#$ of nodes
- $N = \#$ of iterations
- $k = \#$ of communities.
- $c = \#$ of cores.

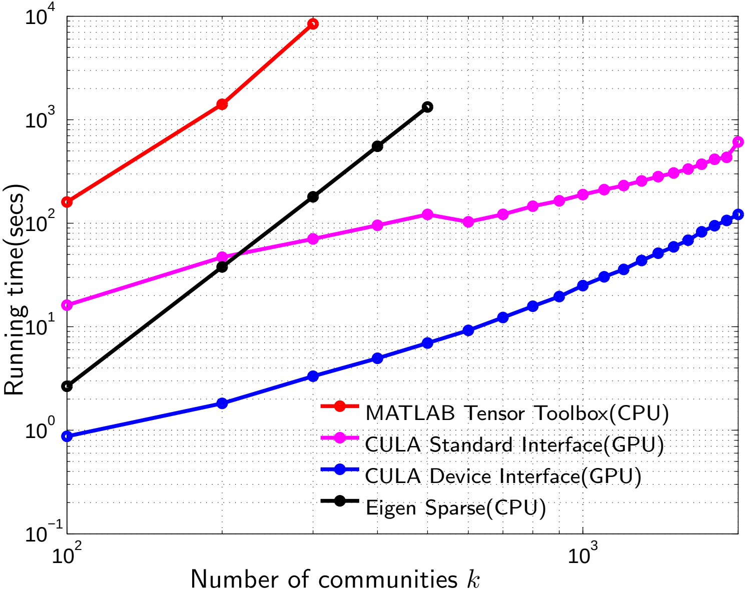
| | Whiten | STGD | Unwhiten |
|---|---|---|---|
| Space | $O(nk)$ | $O(k^2)$ | $O(nk)$ |
| Time | $O(nsk/c + k^3)$ | $O(Nk^3/c)$ | $O(nsk/c)$ |

- Whiten: matrix/vector products and SVD.
- STGD: Stochastic Tensor Gradient Descent
- Unwhiten: matrix/vector products

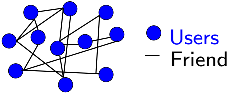
$$\boxed{\text{Our approach: } O(\frac{nsk}{c} + k^3)}$$

Embarrassingly Parallel and fast!

# Scaling Of The Stochastic Iterations

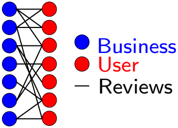


# Summary of Results



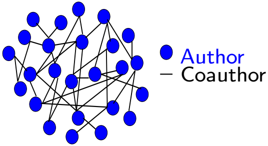

Facebook

$n \sim 20k$




Yelp

$n \sim 40k$




DBLP(sub)

$n \sim 1$ million($\sim 100k$)

Error ($\mathcal{E}$) and Recovery ratio ($\mathcal{R}$)

| Dataset | $\hat{k}$ | Method | **Running Time** | $\mathcal{E}$ | $\mathcal{R}$ |
|---|---|---|---|---|---|
| Facebook(k=360) | 500 | ours | 468 | 0.0175 | 100% |
| Facebook(k=360) | 500 | variational | 86,808 | 0.0308 | 100% |
| Yelp(k=159) | 100 | ours | 287 | 0.046 | 86% |
| Yelp(k=159) | 100 | variational | N.A. | | |
| DBLP sub(k=250) | 500 | ours | 10,157 | 0.139 | 89% |
| DBLP sub(k=250) | 500 | variational | 558,723 | 16.38 | 99% |
| DBLP(k=6000) | 100 | ours | 5407 | 0.105 | 95% |

Thanks to Prem Gopalan and David Mimno for providing variational code.

# Experimental Results on Yelp

Lowest error business categories & largest weight businesses

| Rank | Category | Business | Stars | Review Counts |
|---|---|---|---|---|
| 1 | Latin American | Salvadoreno Restaurant | 4.0 | 36 |
| 2 | Gluten Free | P.F. Chang's China Bistro | 3.5 | 55 |
| 3 | Hobby Shops | Make Meaning | 4.5 | 14 |
| 4 | Mass Media | KJZZ 91.5FM | 4.0 | 13 |
| 5 | Yoga | Sutra Midtown | 4.5 | 31 |

# Experimental Results on Yelp

Lowest error business categories & largest weight businesses

| Rank | Category | Business | Stars | Review Counts |
|---|---|---|---|---|
| 1 | Latin American | Salvadoreno Restaurant | 4.0 | 36 |
| 2 | Gluten Free | P.F. Chang's China Bistro | 3.5 | 55 |
| 3 | Hobby Shops | Make Meaning | 4.5 | 14 |
| 4 | Mass Media | KJZZ 91.5FM | 4.0 | 13 |
| 5 | Yoga | Sutra Midtown | 4.5 | 31 |

Bridgeness: Distance from vector $[1/\hat{k}, \ldots, 1/\hat{k}]^\top$

Top-5 bridging nodes (businesses)

| Business | Categories |
|---|---|
| Four Peaks Brewing | Restaurants, Bars, American, Nightlife, Food, Pubs, Tempe |
| Pizzeria Bianco | Restaurants, Pizza, Phoenix |
| FEZ | Restaurants, Bars, American, Nightlife, Mediterranean, Lounges, Phoenix |
| Matt's Big Breakfast | Restaurants, Phoenix, Breakfast& Brunch |
| Cornish Pasty Co | Restaurants, Bars, Nightlife, Pubs, Tempe |

# Outline

# Beyond Orthogonal Tensor Decomposition

$$\boxed{T = \sum_{j \in [k]} w_j a_j \otimes a_j \otimes a_j.}$$

- $k$: tensor rank, $d$: ambient dimension. $k > d$: overcomplete.

# Beyond Orthogonal Tensor Decomposition

$$\boxed{T = \sum_{j \in [k]} w_j a_j \otimes a_j \otimes a_j.}$$

- $k$: tensor rank, $d$: ambient dimension. $k > d$: overcomplete.
- $A$ is incoherent: $\langle a_i, a_j \rangle \sim \frac{1}{\sqrt{d}}$ for $i \neq j$.

# Beyond Orthogonal Tensor Decomposition

$$\boxed{T = \sum_{j \in [k]} w_j a_j \otimes a_j \otimes a_j.}$$

- $k$: tensor rank, $d$: ambient dimension. $k > d$: overcomplete.
- $A$ is incoherent: $\langle a_i, a_j \rangle \sim \frac{1}{\sqrt{d}}$ for $i \neq j$.
- Guaranteed Recovery when $\boxed{k = o(d^{1.5})}$.

# Beyond Orthogonal Tensor Decomposition

$$\boxed{T = \sum_{j \in [k]} w_j a_j \otimes a_j \otimes a_j.}$$

- $k$: tensor rank, $d$: ambient dimension. $k > d$: overcomplete.
- $A$ is incoherent: $\langle a_i, a_j \rangle \sim \frac{1}{\sqrt{d}}$ for $i \neq j$.
- Guaranteed Recovery when $\boxed{k = o(d^{1.5})}$.
- Tight sample complexity bounds.

---

"Guaranteed Non-Orthogonal Tensor Decomposition via Alternating Rank-1 Updates" by A., R. Ge, M. Janzamin. Preprint, Feb. 2014.

"Provable Learning of Overcomplete Latent Variable Models: Semi-supervised & Unsupervised".

# High-level Intuition for Sample Bounds

- Gaussian mixture model: $x = Ah + z$, where $z$ is noise.
- Exact moment $T = \sum_i w_i a_i \otimes a_i \otimes a_i$.
- Sample moment: $\hat{T} = \frac{1}{n} \sum_i x^i \otimes x^i \otimes x^i - \dots$.

Naive Idea: $\|\hat{T} - T\| \leq \| \mathrm{mat}(\hat{T}) - \mathrm{mat}(T)\|$, apply matrix Bernstein's.

- Our idea: Careful $\epsilon$-net covering for $\hat{T} - T$.
- $\hat{T} - T$ has many terms, e.g. $\frac{1}{n} \sum_i z^i \otimes z^i \otimes z^i$.
- Need to bound $\dfrac{1}{n} \sum_i \langle z^i, u \rangle^3$, for all $u \in \mathcal{S}^{d-1}$.
- Classify inner products into buckets and bound them separately.

# High-level Intuition for Sample Bounds

- Gaussian mixture model: $x = Ah + z$, where $z$ is noise.
- Exact moment $T = \sum_i w_i a_i \otimes a_i \otimes a_i$.
- Sample moment: $\hat{T} = \frac{1}{n} \sum_i x^i \otimes x^i \otimes x^i - \dots$.

Naive Idea: $\|\hat{T} - T\| \le \|\operatorname{mat}(\hat{T}) - \operatorname{mat}(T)\|$, apply matrix Bernstein's.

- Our idea: Careful $\epsilon$-net covering for $\hat{T} - T$.
- $\hat{T} - T$ has many terms, e.g. $\frac{1}{n} \sum_i z^i \otimes z^i \otimes z^i$.
- Need to bound $\displaystyle\frac{1}{n} \sum_i \langle z^i, u \rangle^3$, for all $u \in \mathcal{S}^{d-1}$.
- Classify inner products into buckets and bound them separately.

- Tight sample bounds for a range of latent variable models.
- E.g. Require $\tilde{\Omega}(k)$ samples for $k$-Gaussian mixtures in low-noise regime.

# Main Result: Local Convergence

- Initialization: $\|a_1 - a^{(0)}\| \leq \epsilon_0$, and $\epsilon_0 <$ const.
- Noise: $\hat{T} := T + E$, and $\|E\| \leq 1/\operatorname{polylog}(d)$.
- Error: $\epsilon_T := \|E\| + \tilde{O}\left(\frac{\sqrt{k}}{d}\right)$

Theorem (Local Convergence)

After $O(\log(1/\epsilon_T))$ steps of alternating rank-1 updates,

$$\boxed{\|a_1 - a^{(t)}\| = O(\epsilon_T).}$$

- Linear convergence: up to approximation error.
- Guarantees for overcomplete tensors: $k = o(d^{1.5})$ and for $p^{\text{th}}$-order tensors $k = o(d^{p/2})$.
- Requires good initialization. What about global convergence?

# Global Convergence $k = O(d)$

## SVD Initialization

- Find the top singular vector of $T(I, I, \theta)$ for $\theta \sim \mathcal{N}(0, I)$.
- Use them for initialization. $L$ trials.

## Conditions for global convergence

- Number of initializations: $L \geq k^{\Omega(k/d)^2}$, Tensor Rank: $k = O(d)$
- No. of Iterations: $N = \Theta\left(\log(1/\epsilon_T)\right)$. Recall $\epsilon_T$: approx. error.

# Global Convergence $k = O(d)$

## SVD Initialization

- Find the top singular vector of $T(I, I, \theta)$ for $\theta \sim \mathcal{N}(0, I)$.
- Use them for initialization. $L$ trials.

## Conditions for global convergence

- Number of initializations: $L \geq k^{\Omega(k/d)^2}$, Tensor Rank: $k = O(d)$
- No. of Iterations: $N = \Theta\left(\log(1/\epsilon_T)\right)$. Recall $\epsilon_T$: approx. error.

## Latest Improvement (Assuming Gaussian $a_j$'s)

- Improved initialization requirements for convergence.

$$\boxed{|\langle x^{(0)}, a_j \rangle| \geq d^{\beta} \frac{\sqrt{k}}{d}}$$

.

# Global Convergence $k = O(d)$

## SVD Initialization

- Find the top singular vector of $T(I, I, \theta)$ for $\theta \sim \mathcal{N}(0, I)$.
- Use them for initialization. $L$ trials.

## Conditions for global convergence

- Number of initializations: $L \geq k^{\Omega(k/d)^2}$, Tensor Rank: $k = O(d)$
- No. of Iterations: $N = \Theta\left(\log(1/\epsilon_T)\right)$. Recall $\epsilon_T$: approx. error.

## Latest Improvement (Assuming Gaussian $a_j$'s)

- Improved initialization requirements for convergence.

$$\boxed{|\langle x^{(0)}, a_j \rangle| \geq d^{\beta} \frac{\sqrt{k}}{d}}$$

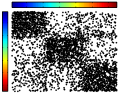
.

- Initialize with samples with noise variance $d\sigma^2$ s.t. $\boxed{\sigma = o\left(\frac{\sqrt{d}}{\sqrt{k}}\right)}$

# Outline

# Conclusion

## Guaranteed Learning of Latent Variable Models



- Efficient sample and computational complexities
- Better performance compared to EM, Variational Bayes etc.

## In practice

- Scalable and embarrassingly parallel: handle large datasets.
- Efficient performance: perplexity or ground truth validation.

## Software Code

- Topic modeling
  `https://github.com/FurongHuang/TopicModeling`
- Community detection
  `https://github.com/FurongHuang/Fast-Detection-of-Overlapp`

---

Youtube videos and slides from ML summer school