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: \( \mathbb{E}[x \otimes x] = \mathbb{E}[xx^\top] \).

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

1. Introduction

2. Spectral Methods: Matrices to Tensors

3. Tensor Forms for Different Models

4. Experimental Results

5. Overcomplete Tensors

6. Conclusion
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

\[
M_2 = \sum_i \lambda_i u_i \otimes v_i
\]

Matrix \( M_2 \)

\( \lambda_1 u_1 \otimes v_1 \)

\( \lambda_2 u_2 \otimes v_2 \)

\ldots \ldots
Spectral Decomposition

\[ M_2 = \sum_i \lambda_i u_i \otimes v_i \]

\[ M_3 = \sum_i \lambda_i u_i \otimes v_i \otimes w_i \]

\( u \otimes v \otimes w \) is a rank-1 tensor since its \((i_1, i_2, i_3)^{th}\) 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: $M v = M(I, v) = \lambda v.$

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. \( W^T A = V \) 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. \( W^T M_2 W = I \).

Eigen-decomposition of \( M_2 = U \text{Diag}(\tilde{\lambda}) U^T \), then \( W = U \text{Diag}(\tilde{\lambda}^{-1/2}) \).
Using Whitening to Obtain Orthogonal Tensor

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 $k \ll d$. 

Tensor $M_3$ → Tensor $T$
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

1. Introduction

2. Spectral Methods: Matrices to Tensors

3. Tensor Forms for Different Models

4. Experimental Results

5. Overcomplete Tensors

6. Conclusion
Topic Modeling

$k$ topics (distributions over vocab words). Each document \( \leftrightarrow \) mixture of topics. Words in document \( \sim_{iid} \) mixture dist.

E.g.,

\[
\begin{align*}
0.6 \cdot \text{sports} & \quad +0.3 \cdot \text{science} & \quad +0.1 \cdot \text{politics} & \quad +0.1 \cdot \text{business} \\
\end{align*}
\]

\[
\begin{array}{c}
\text{aardvark} \\
\text{athlete} \\
\vdots \\
\text{zygote}
\end{array}
\begin{array}{c}
0 \\
3 \\
\vdots \\
1
\end{array}
\]

\[
\begin{align*}
\Pr_{\theta}[\text{“play” | sports}] &= 0.0002 \\
\Pr_{\theta}[\text{“game” | sports}] &= 0.0003 \\
\Pr_{\theta}[\text{“season” | sports}] &= 0.0001 \\
\vdots
\end{align*}
\]
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 \((h)\)

\[ A \]

\[ A \]

\[ A \]

\[ A \]

\[ A \]

\[ A \]

\[ A \]

\[ A \]

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

- Linear model: \[ \mathbb{E}[x_i|h] = Ah. \]
Moments for Single Topic Models

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

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

**Pairwise Co-occurrence 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

\[ M_2 := \mathbb{E}[x_1 \otimes x_2] - \frac{\alpha_0}{\alpha_0 + 1} \mathbb{E}[x_1] \otimes \mathbb{E}[x_1] \]

\[ M_3 := \mathbb{E}[x_1 \otimes x_2 \otimes x_3] - \frac{\alpha_0}{\alpha_0 + 2} \mathbb{E}[x_1 \otimes x_2 \otimes \mathbb{E}[x_1]] - \text{more stuff...} \]

Then

\[ M_2 = \sum \tilde{\omega}_i a_i \otimes a_i \]

\[ M_3 = \sum \tilde{\omega}_i a_i \otimes a_i \otimes a_i. \]

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

Diagram showing network community models with various icons and numbers. The diagram is not described in detail here.
Network Community Models

![Diagram of network community models with images and numbers representing connections and probabilities.]
Network Community Models

![Diagram of network community models with various icons and numbers representing connections and communities.](image-url)
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
3-star counts sufficient for identifiability and learning of MMSB

3-Star Count Tensor

\[ \tilde{M}_3(a, b, c) = \frac{1}{|X|} \text{# of common neighbors in } X \]

\[ = \frac{1}{|X|} \sum_{x \in X} G(x, a) G(x, b) G(x, c). \]

\[ \tilde{M}_3 = \frac{1}{|X|} \sum_{x \in X} [G_{x, A}^\top \otimes G_{x, B}^\top \otimes G_{x, C}^\top] \]
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

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

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

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

- 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 $\alpha_0$

Main Results (Contd)

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

Norm Guarantees

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

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

Norm Guarantees

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

Support Recovery

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

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

Zero-error Support Recovery of Significant Memberships of All Nodes
Outline

1. Introduction
2. Spectral Methods: Matrices to Tensors
3. Tensor Forms for Different Models
4. Experimental Results
5. Overcomplete Tensors
6. Conclusion
**Computational Complexity** ($k \ll n$)

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

<table>
<thead>
<tr>
<th></th>
<th>Whiten</th>
<th>STGD</th>
<th>Unwhiten</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Space</strong></td>
<td>$O(nk)$</td>
<td>$O(k^2)$</td>
<td>$O(nk)$</td>
</tr>
<tr>
<td><strong>Time</strong></td>
<td>$O(nsk/c + k^3)$</td>
<td>$O(Nk^3/c)$</td>
<td>$O(nsk/c)$</td>
</tr>
</tbody>
</table>

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

**Our approach**: $O\left(\frac{nsk}{c} + k^3\right)$

Embarrassingly Parallel and fast!
Scaling Of The Stochastic Iterations

Number of communities $k$ vs. Running time (secs)

- MATLAB Tensor Toolbox (CPU)
- CULA Standard Interface (GPU)
- CULA Device Interface (GPU)
- Eigen Sparse (CPU)
### Summary of Results

<table>
<thead>
<tr>
<th>Dataset</th>
<th>( \hat{k} )</th>
<th>Method</th>
<th>Running Time</th>
<th>( \mathcal{E} )</th>
<th>( \mathcal{R} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Facebook (k=360)</td>
<td>500</td>
<td>ours</td>
<td>468</td>
<td>0.0175</td>
<td>100%</td>
</tr>
<tr>
<td>Facebook (k=360)</td>
<td>500</td>
<td>variational</td>
<td>86,808</td>
<td>0.0308</td>
<td>100%</td>
</tr>
<tr>
<td>Yelp (k=159)</td>
<td>100</td>
<td>ours</td>
<td>287</td>
<td>0.046</td>
<td>86%</td>
</tr>
<tr>
<td>Yelp (k=159)</td>
<td>100</td>
<td>variational</td>
<td>N.A.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DBLP sub (k=250)</td>
<td>500</td>
<td>ours</td>
<td>10,157</td>
<td>0.139</td>
<td>89%</td>
</tr>
<tr>
<td>DBLP sub (k=250)</td>
<td>500</td>
<td>variational</td>
<td>558,723</td>
<td>16.38</td>
<td>99%</td>
</tr>
<tr>
<td>DBLP (k=6000)</td>
<td>100</td>
<td>ours</td>
<td>5407</td>
<td>0.105</td>
<td>95%</td>
</tr>
</tbody>
</table>

Thanks to Prem Gopalan and David Mimno for providing variational code.
Experimental Results on Yelp

Lowest error business categories & largest weight businesses

<table>
<thead>
<tr>
<th>Rank</th>
<th>Category</th>
<th>Business</th>
<th>Stars</th>
<th>Review Counts</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Latin American</td>
<td>Salvadoreno Restaurant</td>
<td>4.0</td>
<td>36</td>
</tr>
<tr>
<td>2</td>
<td>Gluten Free</td>
<td>P.F. Chang's China Bistro</td>
<td>3.5</td>
<td>55</td>
</tr>
<tr>
<td>3</td>
<td>Hobby Shops</td>
<td>Make Meaning</td>
<td>4.5</td>
<td>14</td>
</tr>
<tr>
<td>4</td>
<td>Mass Media</td>
<td>KJZZ 91.5FM</td>
<td>4.0</td>
<td>13</td>
</tr>
<tr>
<td>5</td>
<td>Yoga</td>
<td>Sutra Midtown</td>
<td>4.5</td>
<td>31</td>
</tr>
</tbody>
</table>
Experimental Results on Yelp

Lowest error business categories & largest weight businesses

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

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

Top-5 bridging nodes (businesses)

<table>
<thead>
<tr>
<th>Business</th>
<th>Categories</th>
</tr>
</thead>
<tbody>
<tr>
<td>Four Peaks Brewing</td>
<td>Restaurants, Bars, American, Nightlife, Food, Pubs, Tempe</td>
</tr>
<tr>
<td>Pizzeria Bianco</td>
<td>Restaurants, Pizza, Phoenix</td>
</tr>
<tr>
<td>FEZ</td>
<td>Restaurants, Bars, American, Nightlife, Mediterranean, Lounges, Phoenix</td>
</tr>
<tr>
<td>Matt’s Big Breakfast</td>
<td>Restaurants, Phoenix, Breakfast&amp; Brunch</td>
</tr>
<tr>
<td>Cornish Pasty Co</td>
<td>Restaurants, Bars, Nightlife, Pubs, Tempe</td>
</tr>
</tbody>
</table>
Outline

1. Introduction
2. Spectral Methods: Matrices to Tensors
3. Tensor Forms for Different Models
4. Experimental Results
5. Overcomplete Tensors
6. Conclusion
Beyond Orthogonal Tensor Decomposition

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

- \(k\): tensor rank, \(d\): ambient dimension. \(k > d\): overcomplete.
Beyond Orthogonal Tensor Decomposition

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

- \( k \): 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

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

- \( k \): 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 \( k = o(d^{1.5}) \).
Beyond Orthogonal Tensor Decomposition

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

- \( k \): 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 \( k = o(d^{1.5}) \).
- Tight sample complexity bounds.


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: \( \hat{T} = \frac{1}{n} \sum_i x_i \otimes x_i \otimes x_i \).

Naive Idea: \( \| \hat{T} - T \| \leq \| \text{mat}(\hat{T}) - \text{mat}(T) \| \), 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 \( \frac{1}{n} \sum_i \langle z_i, u \rangle^3 \), for all \( u \in S^{d-1} \).
- 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 $T = \sum_i w_ia_i \otimes a_i \otimes a_i$.
- Sample moment: $\hat{T} = \frac{1}{n} \sum x_i \otimes x_i \otimes x_i - \ldots$

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

- Our idea: Careful $\epsilon$-net covering for $\hat{T} - T$.
- $\hat{T} - T$ has many terms, e.g. $\frac{1}{n} \sum z_i \otimes z_i \otimes z_i$.
- Need to bound $\frac{1}{n} \sum_i \langle z_i, u \rangle^3$, for all $u \in S^{d-1}$.
- 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 \( \epsilon_0 < \text{const.} \).
- **Noise:** \( \hat{T} := T + E \), and \( \|E\| \leq 1/\text{polylog}(d) \).
- **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,

\[
\|a_1 - a^{(t)}\| = O(\epsilon_T).
\]

- **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: \( N = \Theta(\log(1/\epsilon_T)) \). 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: \( N = \Theta(\log(1/\epsilon_T)) \). Recall \( \epsilon_T \): approx. error.

Latest Improvement (Assuming Gaussian \( a_j \)'s)
- Improved initialization requirements for convergence.

\[ |\langle x^{(0)}, a_j \rangle| \geq d^\beta \frac{\sqrt{k}}{d} \]
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: $N = \Theta(\log(1/\epsilon_T))$. Recall $\epsilon_T$: approx. error.

Latest Improvement (Assuming Gaussian $a_j$'s)
- Improved initialization requirements for convergence.
  \[ |\langle x^{(0)}, a_j \rangle| \geq d^\beta \frac{\sqrt{k}}{d} \]
- Initialize with samples with noise variance $d\sigma^2$ s.t.
  \[ \sigma = o\left(\frac{\sqrt{d}}{\sqrt{k}}\right) \]
Outline

1. Introduction
2. Spectral Methods: Matrices to Tensors
3. Tensor Forms for Different Models
4. Experimental Results
5. Overcomplete Tensors
6. Conclusion
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-Overlapping-Communities

Youtube videos and slides from ML summer school