# Learning Overcomplete Latent Variable Models through Tensor Methods


**Anima Anandkumar**

UC Irvine

Joint work with

Majid Janzamin — UC Irvine

Rong Ge — Microsoft Research

# Latent Variable Probabilistic Models

- Latent (hidden) variable $h \in \mathbb{R}^k$, observed variable $x \in \mathbb{R}^d$.

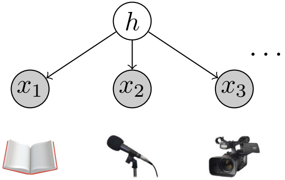
# Latent Variable Probabilistic Models

- Latent (hidden) variable $h \in \mathbb{R}^k$, observed variable $x \in \mathbb{R}^d$.

## Multiview linear mixture models

- Categorical hidden variable $h$.
- Views: conditionally indep. given $h$.
- Linear model:

  $\mathbb{E}[x_1|h] = a_h$, $\mathbb{E}[x_2|h] = b_h$, $\mathbb{E}[x_3|h] = c_h$.



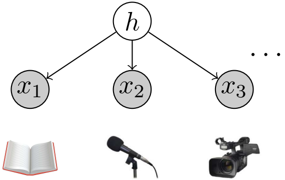
# Latent Variable Probabilistic Models

- Latent (hidden) variable $h \in \mathbb{R}^k$, observed variable $x \in \mathbb{R}^d$.

## Multiview linear mixture models

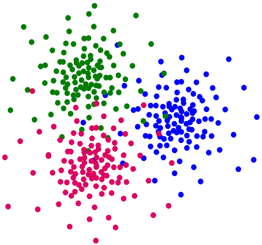
- Categorical hidden variable $h$.
- Views: conditionally indep. given $h$.
- Linear model:

  $\mathbb{E}[x_1|h] = a_h$, $\mathbb{E}[x_2|h] = b_h$, $\mathbb{E}[x_3|h] = c_h$.




## Gaussian Mixture

- Categorical hidden variable $h$.
- $x|h \sim \mathcal{N}(\mu_h, \Sigma_h)$.



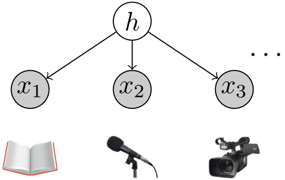
# Latent Variable Probabilistic Models

- Latent (hidden) variable $h \in \mathbb{R}^k$, observed variable $x \in \mathbb{R}^d$.

## Multiview linear mixture models

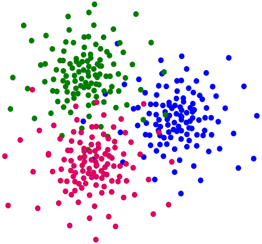
- Categorical hidden variable $h$.
- Views: conditionally indep. given $h$.
- Linear model:

  $\mathbb{E}[x_1|h] = a_h$, $\mathbb{E}[x_2|h] = b_h$, $\mathbb{E}[x_3|h] = c_h$.




## Gaussian Mixture

- Categorical hidden variable $h$.
- $x|h \sim \mathcal{N}(\mu_h, \Sigma_h)$.



ICA, Sparse Coding, HMM, Topic modeling, ...

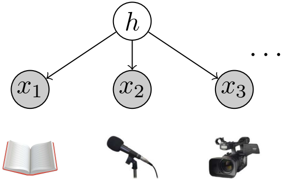
# Latent Variable Probabilistic Models

- Latent (hidden) variable $h \in \mathbb{R}^k$, observed variable $x \in \mathbb{R}^d$.

## Multiview linear mixture models

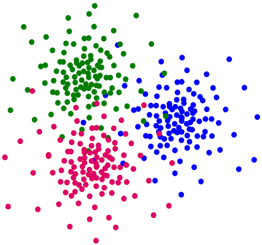
- Categorical hidden variable $h$.
- Views: conditionally indep. given $h$.
- Linear model:

  $\mathbb{E}[x_1|h] = a_h$, $\mathbb{E}[x_2|h] = b_h$, $\mathbb{E}[x_3|h] = c_h$.




## Gaussian Mixture

- Categorical hidden variable $h$.
- $x|h \sim \mathcal{N}(\mu_h, \Sigma_h)$.



ICA, Sparse Coding, HMM, Topic modeling, . . .

---

Efficient Learning of the parameters $a_h, \mu_h, \dots$?

# Method-of-Moments (Spectral methods)

Multi-variate observed moments

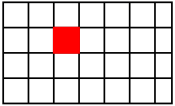
$$M_1 := \mathbb{E}[x], \quad M_2 := \mathbb{E}[x \otimes x], \quad M_3 := \mathbb{E}[x \otimes x \otimes x].$$

# Method-of-Moments (Spectral methods)

## Multi-variate observed 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: $\mathbb{E}[x \otimes x] = \mathbb{E}[x x^\top]$.

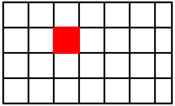
# Method-of-Moments (Spectral methods)

## Multi-variate observed moments

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

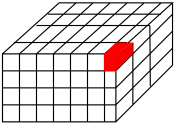
## Matrix

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



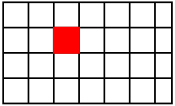
# Method-of-Moments (Spectral methods)

## Multi-variate observed moments

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

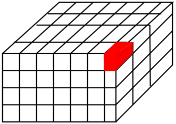
## Matrix

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



---

Information in moments for learning LVMs?

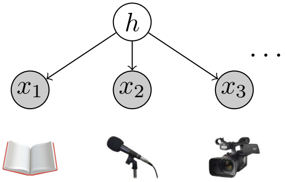
# Multiview Mixture Model

- $[k] := \{1, \ldots, k\}$.

Multiview linear mixture models

- Categorical hidden variable $h \in [k]$.
- $w_j := \Pr[h = j]$
- Views: conditionally indep. given $h$.
- Linear model:

  $\mathbb{E}[x_1|h] = a_h$, $\mathbb{E}[x_2|h] = b_h$, $\mathbb{E}[x_3|h] = c_h$.



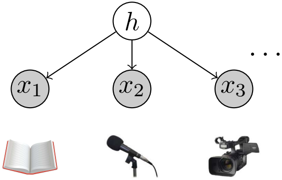
# Multiview Mixture Model

- $[k] := \{1, \ldots, k\}$.

## Multiview linear mixture models

- Categorical hidden variable $h \in [k]$.
- $w_j := \Pr[h = j]$
- Views: conditionally indep. given $h$.
- Linear model:

  $\mathbb{E}[x_1|h] = a_h$, $\mathbb{E}[x_2|h] = b_h$, $\mathbb{E}[x_3|h] = c_h$.




---

$$\mathbb{E}_x[\overbrace{x_1 \otimes x_2}^{x_1 x_2^\top}] = \mathbb{E}_h[\mathbb{E}_x[x_1 \otimes x_2|h]]$$
$$= \mathbb{E}_h[a_h \otimes b_h]$$
$$= \sum_{j\in[k]} w_j a_j \otimes b_j.$$

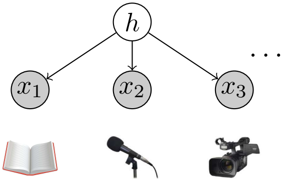
# Multiview Mixture Model

- $[k] := \{1, \ldots, k\}$.

Multiview linear mixture models

- Categorical hidden variable $h \in [k]$.
- $w_j := \Pr[h = j]$
- Views: conditionally indep. given $h$.
- Linear model:

  $\mathbb{E}[x_1|h] = a_h$, $\mathbb{E}[x_2|h] = b_h$, $\mathbb{E}[x_3|h] = c_h$.




---

$$\mathbb{E}[x_1 \otimes x_2] = \sum_{j \in [k]} w_j a_j \otimes b_j,$$

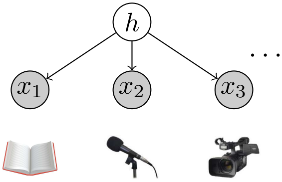
# Multiview Mixture Model

- $[k] := \{1, \dots, k\}$.

Multiview linear mixture models

- Categorical hidden variable $h \in [k]$.
- $w_j := \Pr[h = j]$
- Views: conditionally indep. given $h$.
- Linear model:

  $\mathbb{E}[x_1 | h] = a_h$, $\mathbb{E}[x_2 | h] = b_h$, $\mathbb{E}[x_3 | h] = c_h$.




---

$$\mathbb{E}[x_1 \otimes x_2] = \sum_{j \in [k]} w_j a_j \otimes b_j,$$

$$\boxed{\mathbb{E}[x_1 \otimes x_2 \otimes x_3] = \sum_{j \in [k]} w_j a_j \otimes b_j \otimes c_j.}$$

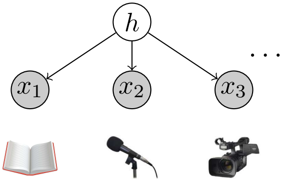
# Multiview Mixture Model

- $[k] := \{1, \ldots, k\}$.

## Multiview linear mixture models

- Categorical hidden variable $h \in [k]$.
- $w_j := \Pr[h = j]$
- Views: conditionally indep. given $h$.
- Linear model:

  $\mathbb{E}[x_1|h] = a_h$, $\mathbb{E}[x_2|h] = b_h$, $\mathbb{E}[x_3|h] = c_h$.




---

$$\mathbb{E}[x_1 \otimes x_2] = \sum_{j \in [k]} w_j a_j \otimes b_j,$$

$$\boxed{\mathbb{E}[x_1 \otimes x_2 \otimes x_3] = \sum_{j \in [k]} w_j a_j \otimes b_j \otimes c_j.}$$

---

Tensor (matrix) factorization for learning LVMs.

# Tensor Rank and Tensor Decomposition

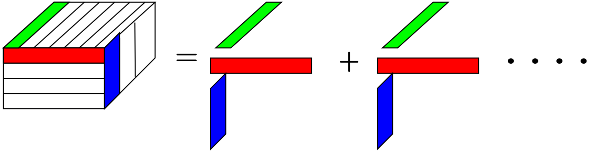
Rank-1 tensor: $T = w \cdot a \otimes b \otimes c \Leftrightarrow T(i, j, l) = w \cdot a(i) \cdot b(j) \cdot c(l)$.

# Tensor Rank and Tensor Decomposition

Rank-1 tensor: $T = w \cdot a \otimes b \otimes c \Leftrightarrow T(i,j,l) = w \cdot a(i) \cdot b(j) \cdot c(l)$.

## CANDECOMP/PARAFAC (CP) Decomposition

$$T = \sum_{j \in [k]} w_j a_j \otimes b_j \otimes c_j \in \mathbb{R}^{d \times d \times d}, \quad a_j, b_j, c_j \in \mathcal{S}^{d-1}.$$



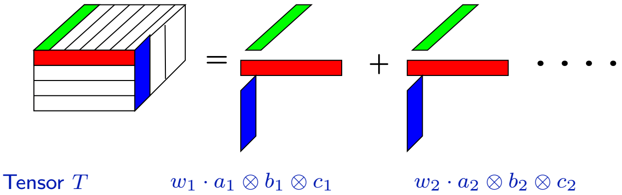
Tensor $T$ $\quad$ $w_1 \cdot a_1 \otimes b_1 \otimes c_1$ $\quad$ $w_2 \cdot a_2 \otimes b_2 \otimes c_2$

# Tensor Rank and Tensor Decomposition

Rank-1 tensor: $T = w \cdot a \otimes b \otimes c \Leftrightarrow T(i,j,l) = w \cdot a(i) \cdot b(j) \cdot c(l)$.

## CANDECOMP/PARAFAC (CP) Decomposition

$$T = \sum_{j \in [k]} w_j a_j \otimes b_j \otimes c_j \in \mathbb{R}^{d \times d \times d}, \quad a_j, b_j, c_j \in \mathcal{S}^{d-1}.$$



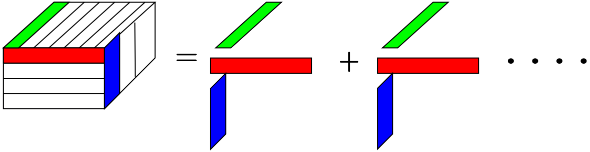

- $k$: tensor rank, $d$: ambient dimension.
- $k \le d$: undercomplete and $k > d$: overcomplete.

# Tensor Rank and Tensor Decomposition

Rank-1 tensor: $T = w \cdot a \otimes b \otimes c \Leftrightarrow T(i,j,l) = w \cdot a(i) \cdot b(j) \cdot c(l)$.

## CANDECOMP/PARAFAC (CP) Decomposition

$$T = \sum_{j \in [k]} w_j a_j \otimes b_j \otimes c_j \in \mathbb{R}^{d \times d \times d}, \quad a_j, b_j, c_j \in \mathcal{S}^{d-1}.$$



Tensor $T$ $\quad w_1 \cdot a_1 \otimes b_1 \otimes c_1 \quad w_2 \cdot a_2 \otimes b_2 \otimes c_2$

- $k$: tensor rank, $d$: ambient dimension.
- $k \le d$: undercomplete and $k > d$: overcomplete.

---

This talk: guarantees for overcomplete tensor decomposition

# Challenges in Tensor Decomposition

Symmetric tensor $T \in \mathbb{R}^{d \times d \times d}$: $T = \sum_{i \in [k]} \lambda_i v_i \otimes v_i \otimes v_i$.

## Challenges in tensors

- Decomposition may not always exist for general tensors.
- Finding the decomposition is NP-hard in general.

# Challenges in Tensor Decomposition

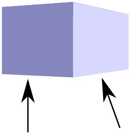
Symmetric tensor $T \in \mathbb{R}^{d\times d\times d}$: $T = \sum_{i\in[k]} \lambda_i v_i \otimes v_i \otimes v_i$.

## Challenges in tensors

- Decomposition may not always exist for general tensors.
- Finding the decomposition is NP-hard in general.

## Tractable case: orthogonal tensor decomposition ($\langle v_i, v_j\rangle = 0$, $i \neq j$)

Algorithm: tensor power method: $v \mapsto \frac{T(I,v,v)}{\|T(I,v,v)\|}$.



---

- $\{v_i\}$'s are the only robust fixed points.

# Challenges in Tensor Decomposition

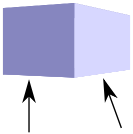
Symmetric tensor $T \in \mathbb{R}^{d\times d\times d}$: $T = \sum_{i\in[k]} \lambda_i v_i \otimes v_i \otimes v_i$.

## Challenges in tensors

- Decomposition may not always exist for general tensors.
- Finding the decomposition is NP-hard in general.

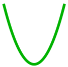
Tractable case: orthogonal tensor decomposition ($\langle v_i, v_j\rangle = 0$, $i \neq j$)

Algorithm: tensor power method: $v \mapsto \dfrac{T(I,v,v)}{\|T(I,v,v)\|}$.



---

- $\{v_i\}$'s are the only robust fixed points.



- All other eigenvectors are saddle points.

# Challenges in Tensor Decomposition

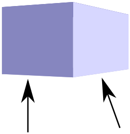
Symmetric tensor $T \in \mathbb{R}^{d\times d\times d}$: $T = \sum_{i\in[k]} \lambda_i v_i \otimes v_i \otimes v_i$.

## Challenges in tensors

- Decomposition may not always exist for general tensors.
- Finding the decomposition is NP-hard in general.

## Tractable case: orthogonal tensor decomposition ($\langle v_i, v_j\rangle = 0$, $i \neq j$)

Algorithm: tensor power method: $v \mapsto \frac{T(I,v,v)}{\|T(I,v,v)\|}$.



---

- $\{v_i\}$'s are the only robust fixed points.

- All other eigenvectors are saddle points.

---

For an orthogonal tensor, no spurious local optima!

# Beyond Orthogonal Tensor Decomposition

## Limitations

- Not ALL tensors have orthogonal decomposition (unlike matrices).

# Beyond Orthogonal Tensor Decomposition

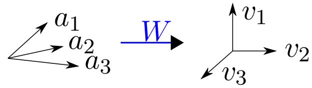
## Limitations

- Not ALL tensors have orthogonal decomposition (unlike matrices).

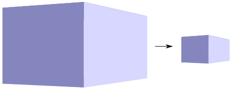
## Undercomplete tensors $(k \leq d)$ with full rank components

Non-orthogonal decomposition $T_1 = \sum_i w_i a_i \otimes a_i \otimes a_i$.

- Whitening matrix $W$:




- Multilinear transform: $T_2 = T_1(W, W, W)$



Tensor $T_1$ Tensor $T_2$

# Beyond Orthogonal Tensor Decomposition

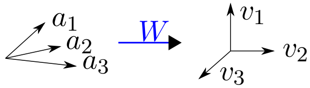
## Limitations

- Not ALL tensors have orthogonal decomposition (unlike matrices).

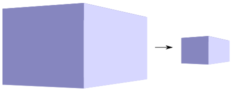
## Undercomplete tensors $(k \le d)$ with full rank components

Non-orthogonal decomposition $T_1 = \sum_i w_i a_i \otimes a_i \otimes a_i$.

- Whitening matrix $W$:




- Multilinear transform: $T_2 = T_1(W, W, W)$



Tensor $T_1$ Tensor $T_2$

---

This talk: guarantees for overcomplete tensor decomposition

# Outline

# Our Setup

## So far

- General tensor decomposition: NP-hard.
- Orthogonal tensors: too limiting.

  Tractable cases? Covers overcomplete tensors?

# Our Setup

## So far

- General tensor decomposition: NP-hard.
- Orthogonal tensors: too limiting.

  Tractable cases? Covers overcomplete tensors?

## Our framework: Incoherent Components

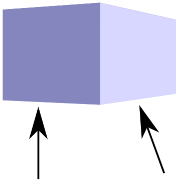
- $|\langle a_i, a_j \rangle| = O\left(1/\sqrt{d}\right)$ for $i \neq j$. Similarly for $b, c$.
- Can handle overcomplete tensors. Satisfied by random vectors.

Guaranteed recovery for alternating minimization?

# Alternating minimization

$$\min_{a,b,c \in \mathcal{S}^{d-1}, w \in \mathbb{R}} \|T - w \cdot a \otimes b \otimes c\|_F.$$

Rank-1 ALS iteration (power iteration)



- Initialization: $a^{(0)}, b^{(0)}, c^{(0)}$.
- Update in $t^{\text{th}}$ step: fix $a^{(t)}, b^{(t)}$ and

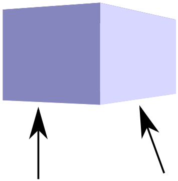
$$c^{(t+1)} \propto T(a^{(t)}, b^{(t)}, I).$$

- After (approx.) convergence, restart.

# Alternating minimization

$$\boxed{\min_{a,b,c \in \mathcal{S}^{d-1}, w \in \mathbb{R}} \|T - w \cdot a \otimes b \otimes c\|_F.}$$

## Rank-1 ALS iteration (power iteration)



- Initialization: $a^{(0)}, b^{(0)}, c^{(0)}$.
- Update in $t^{\text{th}}$ step: fix $a^{(t)}, b^{(t)}$ and

$$c^{(t+1)} \propto T(a^{(t)}, b^{(t)}, I).$$

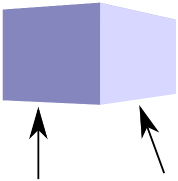
- After (approx.) convergence, restart.

- Simple update: trivially parallelizable and hence scalable.
- Linear computation in dimension, rank, number of different runs.

# Alternating minimization

$$\boxed{\min_{a,b,c\in\mathcal{S}^{d-1},w\in\mathbb{R}} \|T - w\cdot a\otimes b\otimes c\|_F.}$$

## Rank-1 ALS iteration (power iteration)



- Initialization: $a^{(0)}, b^{(0)}, c^{(0)}$.
- Update in $t^{\text{th}}$ step: fix $a^{(t)}, b^{(t)}$ and

$$c^{(t+1)} \propto T(a^{(t)}, b^{(t)}, I).$$

- After (approx.) convergence, restart.
- Simple update: trivially parallelizable and hence scalable.
- Linear computation in dimension, rank, number of different runs.

**Rank-1 ALS iteration ≡ asymmetric power iteration**

# Main Result: Local Convergence

- Initialization: $\max\{\|a_1 - \hat{a}^{(0)}\|, \|b_1 - \hat{b}^{(0)}\|\} \leq \epsilon_0$, and $\epsilon_0 <$ constant.
- Noise: $\hat{T} := T + E$, and $\|E\| \leq 1/\operatorname{polylog}(d)$.
- Rank: $k = o(d^{1.5})$.

---

# Main Result: Local Convergence

- Initialization: $\max\{\|a_1 - \hat{a}^{(0)}\|, \|b_1 - \hat{b}^{(0)}\|\} \leq \epsilon_0$, and $\epsilon_0 <$ constant.
- Noise: $\hat{T} := T + E$, and $\|E\| \leq 1/\operatorname{polylog}(d)$.
- Rank: $k = o(d^{1.5})$.

---

Theorem (Local Convergence)[AGJ2014]

After $N = O(\log(1/\|E\|))$ steps of alternating rank-$1$ updates,

$$\boxed{\|a_1 - \hat{a}^{(N)}\| = O\left(\|E\|\right).}$$

# Main Result: Local Convergence

- Initialization: $\max\{\|a_1 - \hat{a}^{(0)}\|, \|b_1 - \hat{b}^{(0)}\|\} \le \epsilon_0$, and $\epsilon_0 <$ constant.
- Noise: $\hat{T} := T + E$, and $\|E\| \le 1/\operatorname{polylog}(d)$.
- Rank: $k = o(d^{1.5})$.

---

Theorem (Local Convergence)[AGJ2014]

After $N = O(\log(1/\|E\|))$ steps of alternating rank-1 updates,

$$\boxed{\|a_1 - \hat{a}^{(N)}\| = O\left(\|E\|\right).}$$

- 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 vectors of $T(I, I, \theta)$ for $\theta \sim \mathcal{N}(0, I)$.
- Use them for initialization. $L$ trials.

# Global Convergence $k = O(d)$

## SVD Initialization

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

## Assumptions

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

# Global Convergence $k = O(d)$

## SVD Initialization

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

## Assumptions

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

Theorem (Global Convergence)[AGJ2014]: $\boxed{\|a_1 - \hat{a}^{(N)}\| \leq O(\epsilon_R).}$

# Outline

# High-level Intuition for Sample Bounds

- Multi-view Model: $x_1 = Ah + z_1$, where $z_1$ is noise.
- Exact moment $T = \sum_i w_i a_i \otimes b_i \otimes c_i$.
- Sample moment: $\hat{T} = \frac{1}{n} \sum_i x_1^i \otimes x_2^i \otimes x_3^i$.

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

# High-level Intuition for Sample Bounds

- Multi-view Model: $x_1 = Ah + z_1$, where $z_1$ is noise.
- Exact moment $T = \sum_i w_i a_i \otimes b_i \otimes c_i$.
- Sample moment: $\hat{T} = \frac{1}{n} \sum_i x_1^i \otimes x_2^i \otimes x_3^i$.

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

- Our idea: Careful $\epsilon$-net covering for $\hat{T} - T$.
- $\hat{T} - T$ has many terms, e.g., all-noise term: $\frac{1}{n} \sum_i z_1^i \otimes z_2^i \otimes z_3^i$ and signal-noise terms.
- Need to bound $\frac{1}{n} \sum_i \langle z_1^i, u \rangle \langle z_2^i, v \rangle \langle z_3^i, w \rangle$, for all $u, v, w \in \mathcal{S}^{d-1}$.
- Classify inner products into buckets and bound them separately.

Tight sample bounds for a range of latent variable models

# Unsupervised Learning of Gaussian Mixtures

- No. of mixture components: $k = C \cdot d$
- No. of unlabeled samples: $n = \tilde{\Omega}(k \cdot d)$.
- Computational complexity: $\tilde{O}\left(k^{C^2}\right)$

Our result: achieved error with $n$ unlabeled samples

$$\boxed{\max_j \|\widehat{a}_j - a_j\| = \tilde{O}\left(\sqrt{\frac{k}{n}}\right)}$$

- Linear convergence.
- Error: same as before, for semi-supervised setting.
- Computational complexity: polynomial when $k = \Theta(d)$.

# Semi-supervised Learning of Gaussian Mixtures

- $n$ unlabeled samples, $m_j$: samples for component $j$.
- No. of mixture components: $k = o(d^{1.5})$
- No. of labeled samples: $m_j = \tilde{\Omega}(1)$.
- No. of unlabeled samples: $n = \tilde{\Omega}(k)$.

Our result: achieved error with $n$ unlabeled samples

$$\boxed{\max_j \|\widehat{a}_j - a_j\| = \tilde{O}\left(\sqrt{\frac{k}{n}}\right)}$$

- Linear convergence.
- Can handle (polynomially) overcomplete mixtures.
- Extremely small number of labeled samples: $\mathrm{polylog}(d)$.
- Sample complexity is tight: need $\tilde{\Omega}(k)$ samples!

# Outline

# Conclusion

- Learning overcomplete Latent variable models.
  - ⋆ Method-of-moments.
  - ⋆ Tensor power iteration.
- Robustness to noise.
- Sample complexity bounds for a range of LVMs.
  - ⋆ Unsupervised setting.
  - ⋆ Semi-supervised setting.

# Conclusion

- Learning overcomplete Latent variable models.
  - ⋆ Method-of-moments.
  - ⋆ Tensor power iteration.
- Robustness to noise.
- Sample complexity bounds for a range of LVMs.
  - ⋆ Unsupervised setting.
  - ⋆ Semi-supervised setting.
- Latest result: improved initialization for tensor with Gaussian components.

# Conclusion

- Learning overcomplete Latent variable models.
  - ⋆ Method-of-moments.
  - ⋆ Tensor power iteration.
- Robustness to noise.
- Sample complexity bounds for a range of LVMs.
  - ⋆ Unsupervised setting.
  - ⋆ Semi-supervised setting.
- Latest result: improved initialization for tensor with Gaussian components.

Thank you!