Learning Overcomplete Latent Variable Models through Tensor Methods

Majid Janzamin

UC Irvine

Joint work with

Anima Anandkumar — UC Irvine

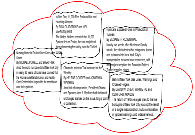
Rong Ge — Microsoft Research

Latent Variable Modeling

Goal: Discover hidden effects from observed measurements

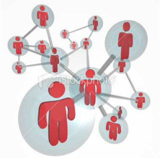
Document modeling

- Observed: words.
- Hidden: topics.



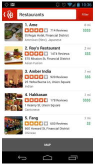
Social Network Modeling

- Observed: social interactions.
- Hidden: communities, relationships.



Recommendation Systems

- Observed: recommendations (e.g., reviews).
- Hidden: User and business attributes



Applications in Speech, Vision, . . .

Latent Variable Modeling

Feature Learning

- Learn good features/representations for classification tasks, e.g., image and speech recognition.

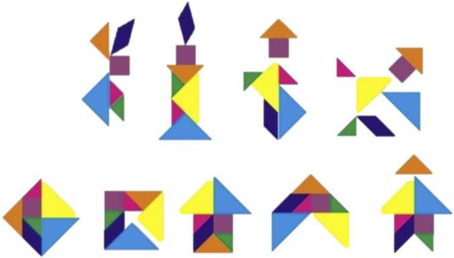
Latent Variable Modeling

Feature Learning

- Learn good features/representations for classification tasks, e.g., image and speech recognition.

Sparse Coding, Dictionary Learning

- Sparse representations, low dimensional hidden structures.
- A few dictionary elements make complicated shapes.



(Image from Sanjeev Arora's slides.)

Learning Latent Variable Models

Goal: Discover hidden effects from observed measurements.

- Unsupervised learning: no labeled samples.

Learning Latent Variable Models

Goal: Discover hidden effects from observed measurements.

- Unsupervised learning: no labeled samples.
- Semi-supervised learning: few labeled samples.

Learning Latent Variable Models

Goal: Discover hidden effects from observed measurements.

- Unsupervised learning: no labeled samples.
- Semi-supervised learning: few labeled samples.

Challenge: Conditions for Identifiability

- When can model be identified (given infinite computation and data)?
- Does identifiability also lead to tractable algorithms?

Learning Latent Variable Models

Goal: Discover hidden effects from observed measurements.

- Unsupervised learning: no labeled samples.
- Semi-supervised learning: few labeled samples.

Challenge: Conditions for Identifiability

- When can model be identified (given infinite computation and data)?
- Does identifiability also lead to tractable algorithms?

Challenge: Efficient Learning of Latent Variable Models

- Maximum likelihood is NP-hard in most cases.
- Practice: EM, Variational Bayes, but have no consistency guarantees.
- Scalable guaranteed learning algorithms?
  - ⋆ Low computational and statistical complexity

Learning Latent Variable Models

Goal: Discover hidden effects from observed measurements.

- Unsupervised learning: no labeled samples.
- Semi-supervised learning: few labeled samples.

Challenge: Conditions for Identifiability

- When can model be identified (given infinite computation and data)?
- Does identifiability also lead to tractable algorithms?

Challenge: Efficient Learning of Latent Variable Models

- Maximum likelihood is NP-hard in most cases.
- Practice: EM, Variational Bayes, but have no consistency guarantees.
- Scalable guaranteed learning algorithms?
  - ⋆ Low computational and statistical complexity

This talk: guaranteed and efficient learning through spectral methods.

LVMs as Probabilistic Models

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

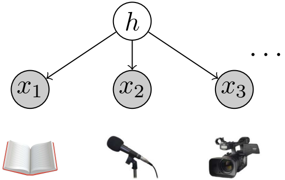
LVMs as Probabilistic Models

- 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:

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



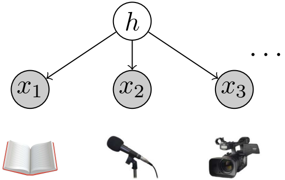
LVMs as Probabilistic Models

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

Multiview linear mixture models

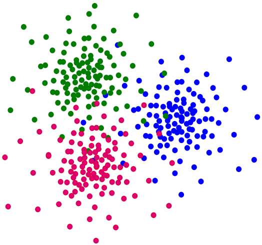
- Categorical hidden variable h.
- Views: conditionally indep. given h.
- Linear model:

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



Gaussian Mixture

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



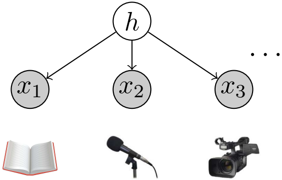
LVMs as Probabilistic Models

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

Multiview linear mixture models

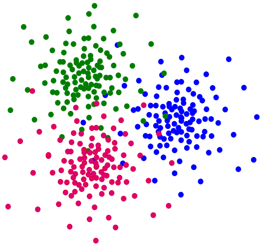
- Categorical hidden variable h.
- Views: conditionally indep. given h.
- Linear model:

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



Gaussian Mixture

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



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

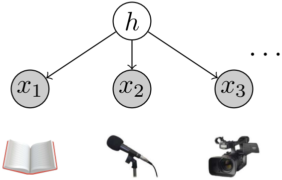
LVMs as Probabilistic Models

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

Multiview linear mixture models

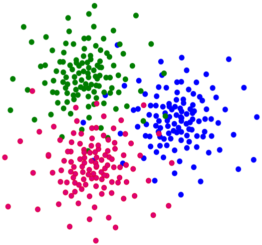
- Categorical hidden variable h.
- Views: conditionally indep. given h.
- Linear model:

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



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 $a_h, \mu_h, \ldots$?

Method-of-Moments (Spectral methods)

Multi-variate observed moments

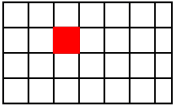
$$M_1 := \mathbb{E}[x], \quad M_2 := \mathbb{E}[x \otimes x], \quad M_3 := \mathbb{E}[x \otimes x \otimes x].$$

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: $\mathbb{E}[x \otimes x] = \mathbb{E}[x x^\top]$.

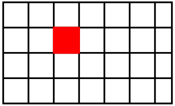
Method-of-Moments (Spectral methods)

Multi-variate observed moments

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

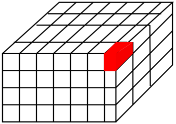
Matrix

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



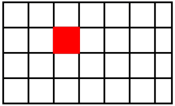
Method-of-Moments (Spectral methods)

Multi-variate observed moments

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

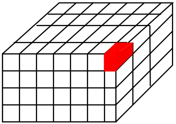
Matrix

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



Tensor

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



Information in moments for learning LVMs?

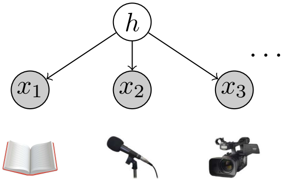
Multiview Mixture Model

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

Multiview linear mixture models

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

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



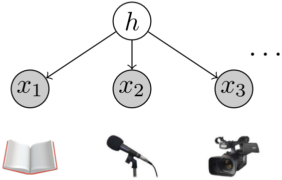
Multiview Mixture Model

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

Multiview linear mixture models

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

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



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

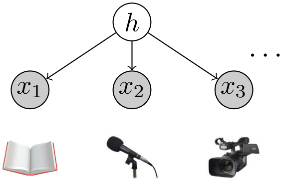
Multiview Mixture Model

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

Multiview linear mixture models

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

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



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

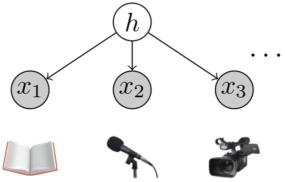
Multiview Mixture Model

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

Multiview linear mixture models

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

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



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

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

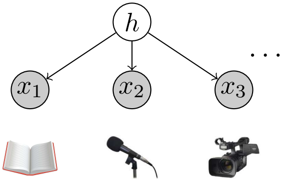
Multiview Mixture Model

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

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.

Matrix vs. Tensor Decomposition

Uniqueness of decomposition.

Matrix Decomposition

- Distinct weights.
- Orthogonal components, i.e., $\langle a_i, a_j \rangle = 0$, $i \neq j$.
- Too limiting.
- Otherwise, only learning up to subspace is possible.

Matrix vs. Tensor Decomposition

Uniqueness of decomposition.

Matrix Decomposition

- Distinct weights.
- Orthogonal components, i.e., $\langle a_i, a_j \rangle = 0$, $i \neq j$.
- Too limiting.
- Otherwise, only learning up to subspace is possible.

Tensor Decomposition

- Same weights.
- Non-Orthogonal components $\Rightarrow$ Overcomplete models.
- More general models.

Matrix vs. Tensor Decomposition

Uniqueness of decomposition.

Matrix Decomposition

- Distinct weights.
- Orthogonal components, i.e., $\langle a_i, a_j \rangle = 0$, $i \neq j$.
- Too limiting.
- Otherwise, only learning up to subspace is possible.

Tensor Decomposition

- Same weights.
- Non-Orthogonal components $\Rightarrow$ Overcomplete models.
- More general models.

Focus on tensor decomposition for learning LVMs.

Overcomplete Latent Variable Models

Overcomplete Latent Representations

- Latent dimensionality $>$ observed dimensionality, i.e., $k > d$.
- Flexible modeling, robust to noise.
- Applicable in speech and image modeling.
- Large amount of unlabeled samples.

Overcomplete Latent Variable Models

Overcomplete Latent Representations

- Latent dimensionality $>$ observed dimensionality, i.e., $k > d$.
- Flexible modeling, robust to noise.
- Applicable in speech and image modeling.
- Large amount of unlabeled samples.
- Possible to learn when using higher (e.g., 3rd) order tensor moment.

Overcomplete Latent Variable Models

Overcomplete Latent Representations

- Latent dimensionality $>$ observed dimensionality, i.e., $k > d$.
- Flexible modeling, robust to noise.
- Applicable in speech and image modeling.
- Large amount of unlabeled samples.
- Possible to learn when using higher (e.g., 3rd) order tensor moment.

Example: $T \in \mathbb{R}^{2\times 2\times 2}$ with rank 3 ($d = 2, k = 3$)

$$T(:,:,1) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad T(:,:,2) = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}.$$

$$T = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \otimes \begin{bmatrix} 0 \\ 1 \end{bmatrix} \otimes \begin{bmatrix} 0 \\ 1 \end{bmatrix} + \begin{bmatrix} 1 \\ -1 \end{bmatrix} \otimes \begin{bmatrix} 1 \\ 0 \end{bmatrix} \otimes \begin{bmatrix} 1 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \otimes \begin{bmatrix} 1 \\ 1 \end{bmatrix} \otimes \begin{bmatrix} 1 \\ -1 \end{bmatrix}.$$

Overcomplete Latent Variable Models

Overcomplete Latent Representations

- Latent dimensionality $>$ observed dimensionality, i.e., $k > d$.
- Flexible modeling, robust to noise.
- Applicable in speech and image modeling.
- Large amount of unlabeled samples.
- Possible to learn when using higher (e.g., 3rd) order tensor moment.

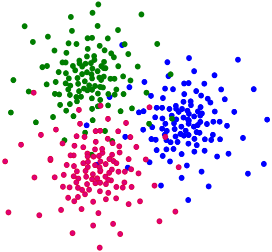
So far

- Learning LVMs.
- Spectral methods (method-of-moments).
- Overcomplete LVMs.

This work: theoretical guarantees for above.

Outline

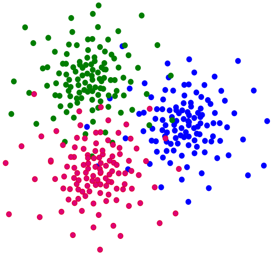
Spherical Gaussian Mixtures



Assumptions

- k components, d: observed dimension.
- Component means a_i incoherent: randomly drawn from the sphere.
- Spherical variance $\frac{\sigma^2}{d}I$ (assume known).

Spherical Gaussian Mixtures



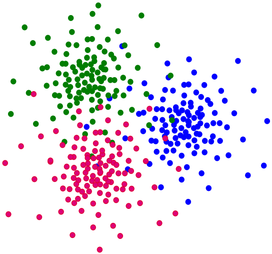
Assumptions

- k components, d: observed dimension.
- Component means a_i incoherent: randomly drawn from the sphere.
- Spherical variance $\frac{\sigma^2}{d} I$ (assume known).

In this talk: special case

- Noise norm $\sigma^2 = 1$: same as signal.
- Uniform probability of components.

Spherical Gaussian Mixtures



Assumptions

- k components, d: observed dimension.
- Component means a_i incoherent: randomly drawn from the sphere.
- Spherical variance $\frac{\sigma^2}{d} I$ (assume known).

In this talk: special case

- Noise norm $\sigma^2 = 1$: same as signal.
- Uniform probability of components.

Tensor For Learning (Hsu, Kakade 2012)

$$M_3 := \mathbb{E}[x^{\otimes 3}] - \sigma^2 \sum_{i \in [d]} (\mathbb{E}[x] \otimes e_i \otimes e_i + \cdots) \Rightarrow M_3 = \sum_{j \in [k]} w_j a_j \otimes a_j \otimes a_j.$$

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

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

- 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!
- Approximation error: decaying in high dimensions.

Unsupervised Learning of Gaussian Mixtures

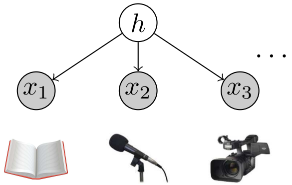
- No. of mixture components: $k = C \cdot d$
- No. of unlabeled samples: $n = \tilde{\Omega}(k \cdot d)$.
- Computational complexity: $\tilde{O}\left(k^{C^2}\right)$

Our result: achieved error with n unlabeled samples

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

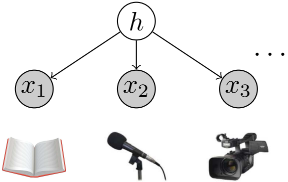
- Linear convergence.
- Error: same as before, for semi-supervised setting.
- Sample complexity: worse than semi-supervised, but better than previous works (no dependence on condition number of A).
- Computational complexity: polynomial when $k = \Theta(d)$.

Multi-view Mixture Models



- $A = [a_1 \; a_2 \; \cdots \; a_k] \in \mathbb{R}^{d \times k}$, similarly B and C.
- Linear model: $x_1 = Ah + z_1$, $x_2 = Bh + z_2$, $x_3 = Ch + z_3$.

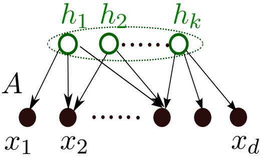
Multi-view Mixture Models



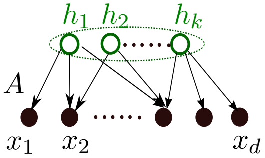
- $A = [a_1 \ a_2 \ \cdots \ a_k] \in \mathbb{R}^{d \times k}$, similarly B and C.
- Linear model: $x_1 = Ah + z_1$, $x_2 = Bh + z_2$, $x_3 = Ch + z_3$.
- Incoherence: Component means a_i's are incoherent (randomly drawn from unit sphere). Similarly b_i's and c_i's.
- The zero-mean noise z_l's satisfy RIP, e.g., Gaussian, Bernoulli.
- Same results as Gaussian mixtures.

Independent Component Analysis

- $x = Ah$, independent sources, unknown mixing.
- Blind source separation of speech, image, video.



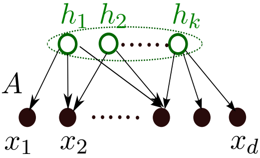
Independent Component Analysis



- $x = Ah$, independent sources, unknown mixing.
- Blind source separation of speech, image, video.

- Sources h are sub-Gaussian (but not Gaussian).
- Columns of A are incoherent.
- Form cumulant tensor $M_4 := \mathbb{E}[x^{\otimes 4}] - \cdots$
- n samples. k sources. d dimensions.

Independent Component Analysis



- $x = Ah$, independent sources, unknown mixing.
- Blind source separation of speech, image, video.

- Sources h are sub-Gaussian (but not Gaussian).
- Columns of A are incoherent.
- Form cumulant tensor $M_4 := \mathbb{E}[x^{\otimes 4}] - \cdots$
- n samples. k sources. d dimensions.

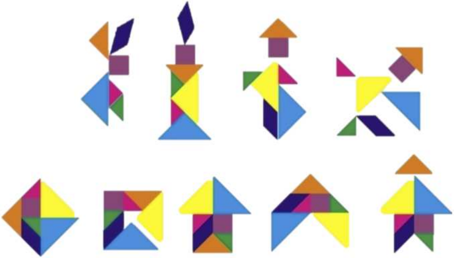
Learning Result

- Semi-supervised: $k = o(d^2)$, $n \geq \tilde{\Omega}(\max(k^2, k^4/d^3))$.
- Unsupervised: $k = O(d)$, $n \geq \tilde{\Omega}(k^3)$.

$$\boxed{\max_j \min_{f \in \{-1,1\}} \|f\widehat{a}_j - a_j\| = \tilde{O}\left(\frac{k^2}{\min\left(n, \sqrt{d^3 n}\right)}\right) + \tilde{O}\left(\frac{\sqrt{k}}{d^{1.5}}\right)}$$

Sparse Coding

- Sparse representations, low dimensional hidden structures.
- A few dictionary elements make complicated shapes.



Sparse Coding

- $x = Ah$, sparse coefficients, unknown dictionary.
- Image compression, feature learning, ...

Sparse Coding

- $x = Ah$, sparse coefficients, unknown dictionary.
- Image compression, feature learning, ...

- Coefficients h are independent Bernoulli Gaussian: Sparse ICA.
- Columns of A are incoherent.
- Form cumulant tensor $M_4 := \mathbb{E}[x^{\otimes 4}] - \cdots$
- n samples. k dictionary elements. d dimensions. s avg. sparsity.

Sparse Coding

- $x = Ah$, sparse coefficients, unknown dictionary.
- Image compression, feature learning, ...

- Coefficients h are independent Bernoulli Gaussian: Sparse ICA.
- Columns of A are incoherent.
- Form cumulant tensor $M_4 := \mathbb{E}[x^{\otimes 4}] - \cdots$
- n samples. k dictionary elements. d dimensions. s avg. sparsity.

Learning Result

- Semi-supervised: $k = o(d^2)$, $n \geq \tilde{\Omega}(\max(sk, s^2k^2/d^3))$.
- Unsupervised: $k = O(d)$, $n \geq \tilde{\Omega}(sk^2)$.

$$\boxed{\max_j \min_{f \in \{-1,1\}} \|f\widehat{a}_j - a_j\| = \tilde{O}\left(\frac{sk}{\min\left(n, \sqrt{d^3 n}\right)}\right) + \tilde{O}\left(\frac{\sqrt{k}}{d^{1.5}}\right)}$$

Outline

Recap of Orthogonal Matrix Eigen Analysis

Symmetric $M \in \mathbb{R}^{d \times d}$

- Eigen-vectors are fixed points: $Mv = \lambda v$.
- Eigen decomposition: $M = \sum_i \lambda_i v_i v_i^\top$. Orthogonal: $\langle v_i, v_j \rangle = 0, i \neq j$.

Recap of Orthogonal Matrix Eigen Analysis

Symmetric $M \in \mathbb{R}^{d \times d}$

- Eigen-vectors are fixed points: $Mv = \lambda v$.
- Eigen decomposition: $M = \sum_i \lambda_i v_i v_i^\top$. Orthogonal: $\langle v_i, v_j \rangle = 0, i \neq j$.

Uniqueness (Identifiability): Iff. λ_i's are distinct.

Recap of Orthogonal Matrix Eigen Analysis

Symmetric $M \in \mathbb{R}^{d \times d}$

- Eigen-vectors are fixed points: $Mv = \lambda v$.
- Eigen decomposition: $M = \sum_i \lambda_i v_i v_i^\top$. Orthogonal: $\langle v_i, v_j \rangle = 0, i \neq j$.

Uniqueness (Identifiability): Iff. λ_i's are distinct.

Algorithm: Power method: $\boxed{v \mapsto \frac{Mv}{\|Mv\|}.}$

Recap of Orthogonal Matrix Eigen Analysis

Symmetric $M \in \mathbb{R}^{d\times d}$

- Eigen-vectors are fixed points: $Mv = \lambda v$.
- Eigen decomposition: $M = \sum_i \lambda_i v_i v_i^\top$. Orthogonal: $\langle v_i, v_j \rangle = 0, i \neq j$.

Uniqueness (Identifiability): Iff. λ_i's are distinct.

Algorithm: Power method: $\boxed{v \mapsto \frac{Mv}{\|Mv\|}.}$

Convergence properties

- Let $\lambda_1 > \lambda_2 > \cdots > \lambda_d$.
- Only v_i's are fixed points of power iteration. $Mv_i = \lambda_i v_i$.

Recap of Orthogonal Matrix Eigen Analysis

Symmetric $M \in \mathbb{R}^{d \times d}$

- Eigen-vectors are fixed points: $Mv = \lambda v$.
- Eigen decomposition: $M = \sum_i \lambda_i v_i v_i^\top$. Orthogonal: $\langle v_i, v_j \rangle = 0, i \neq j$.

Uniqueness (Identifiability): Iff. λ_i's are distinct.

Algorithm: Power method: $\boxed{v \mapsto \frac{Mv}{\|Mv\|}.}$

Convergence properties

- Let $\lambda_1 > \lambda_2 > \cdots > \lambda_d$.
- Only v_i's are fixed points of power iteration. $Mv_i = \lambda_i v_i$.
  - v_1 is the only robust fixed point.

Recap of Orthogonal Matrix Eigen Analysis

Symmetric $M \in \mathbb{R}^{d \times d}$

- Eigen-vectors are fixed points: $Mv = \lambda v$.
- Eigen decomposition: $M = \sum_i \lambda_i v_i v_i^\top$. Orthogonal: $\langle v_i, v_j \rangle = 0, i \neq j$.

Uniqueness (Identifiability): Iff. λ_i's are distinct.

Algorithm: Power method: $\boxed{v \mapsto \dfrac{Mv}{\|Mv\|}.}$

Convergence properties

- Let $\lambda_1 > \lambda_2 > \cdots > \lambda_d$.
- Only v_i's are fixed points of power iteration. $Mv_i = \lambda_i v_i$.

- v_1 is the only robust fixed point.

- All other v_i's are saddle points.

Recap of Orthogonal Matrix Eigen Analysis

Symmetric $M \in \mathbb{R}^{d\times d}$

- Eigen-vectors are fixed points: $Mv = \lambda v$.
- Eigen decomposition: $M = \sum_i \lambda_i v_i v_i^\top$. Orthogonal: $\langle v_i, v_j\rangle = 0, i \neq j$.

Uniqueness (Identifiability): Iff. λ_i's are distinct.

Algorithm: Power method: $\boxed{v \mapsto \dfrac{Mv}{\|Mv\|}.}$

Convergence properties

- Let $\lambda_1 > \lambda_2 > \cdots > \lambda_d$.
- Only v_i's are fixed points of power iteration. $Mv_i = \lambda_i v_i$.

- v_1 is the only robust fixed point.

- All other v_i's are saddle points.

Power method recovers v_1 when initialization v satisfies $\langle v, v_1\rangle \neq 0$.

Tensor Rank and Tensor Decomposition

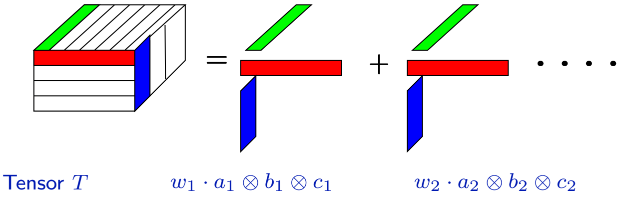
Rank-1 tensor: $T = w \cdot a \otimes b \otimes c \Leftrightarrow T(i,j,l) = w \cdot a(i) \cdot b(j) \cdot c(l)$.

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

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

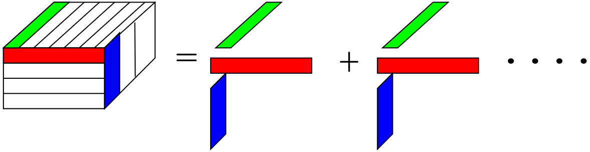


Tensor Rank and Tensor Decomposition

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

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



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

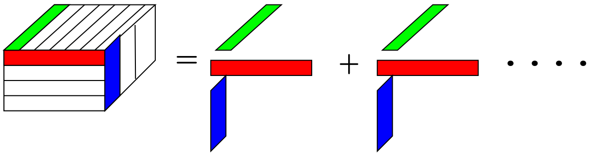
- k: tensor rank, d: ambient dimension.
- $k \le d$: undercomplete and $k > d$: overcomplete.

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

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



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

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

This talk: guarantees for overcomplete tensor decomposition

Background on Tensor Decomposition

$$\boxed{T = \sum_{i \in [k]} w_i a_i \otimes b_i \otimes c_i, \quad a_i, b_i, c_i \in \mathcal{S}^{d-1}.}$$

Theoretical Guarantees

- Tensor decompositions in psychometrics (Cattell '44).
- CP tensor decomposition (Harshman '70, Carol & Chang '70).
- Identifiability of CP tensor decomposition (Kruskal '76).
- Orthogonal decomposition: (Zhang & Golub '01, Kolda '01, Anandkumar etal '12).
- Tensor decomposition through (lifted) linear equations (Lawthauwer '07): works for overcomplete tensors.
- Tensor decomposition through simultaneous diagonalization: perturbation analysis (Goyal et. al '13, Bhaskara '13)

Background on Tensor Decompositions (contd.)

$$\boxed{T = \sum_{i \in [k]} w_i a_i \otimes b_i \otimes c_i, \quad a_i, b_i, c_i \in \mathcal{S}^{d-1}.}$$

Practice: Alternating least squares (ALS)

- Let $A = [a_1 | a_2 \cdots a_k]$ and similarly B, C.
- Fix estimates of two of the modes (say for A and B) and re-estimate the third.
- Iterative updates, low computational complexity.
- No theoretical guarantees.

In this talk: analysis of alternating minimization

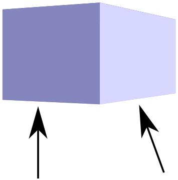
Tensors as Multilinear Transformations

- Tensor $T \in \mathbb{R}^{d \times d \times d}$.
- Vectors $v, w \in \mathbb{R}^d$.

Tensors as Multilinear Transformations

- Tensor $T \in \mathbb{R}^{d\times d\times d}$.
- Vectors $v, w \in \mathbb{R}^d$.

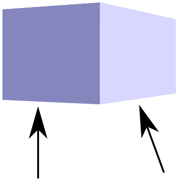
$$T(I, v, w) := \sum_{j,l\in[d]} v_j w_l T(:, j, l) \ \in \mathbb{R}^d.$$



Tensors as Multilinear Transformations

- Tensor $T \in \mathbb{R}^{d\times d\times d}$.
- Vectors $v, w \in \mathbb{R}^d$.

$$T(I, v, w) := \sum_{j,l\in[d]} v_j w_l T(:, j, l) \ \in \mathbb{R}^d.$$



- For matrix $M \in \mathbb{R}^{d\times d}$:

$$M(I, w) = Mw = \sum_{l\in[d]} w_l M(:, l) \ \in \mathbb{R}^d.$$

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 ($\langle v_i, v_j\rangle = 0$, $i \neq j$)

- $\{v_i\}$ are eigenvectors: $T(I, v_i, v_i) = \lambda_i v_i$.

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 ($\langle v_i, v_j\rangle = 0$, $i \neq j$)

- $\{v_i\}$ are eigenvectors: $T(I, v_i, v_i) = \lambda_i v_i$.
- Bad news: There can be other eigenvectors (unlike matrix case).

$$v = \frac{v_1 + v_2}{\sqrt{2}} \text{ satisfies } T(I, v, v) = \frac{1}{\sqrt{2}} v. \quad \lambda_i \equiv 1.$$

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 ($\langle v_i, v_j \rangle = 0,\ i \neq j$)

- $\{v_i\}$ are eigenvectors: $T(I, v_i, v_i) = \lambda_i v_i$.
- Bad news: There can be other eigenvectors (unlike matrix case).

$$v = \frac{v_1 + v_2}{\sqrt{2}} \text{ satisfies } T(I, v, v) = \frac{1}{\sqrt{2}} v. \quad \lambda_i \equiv 1.$$

How do we avoid spurious solutions (not part of decomposition)?

Orthogonal Tensor Power Method

Symmetric orthogonal tensor $T \in \mathbb{R}^{d \times d \times d}$:

$$T = \sum_{i \in [k]} \lambda_i v_i \otimes v_i \otimes v_i.$$

Orthogonal Tensor Power Method

Symmetric orthogonal tensor $T \in \mathbb{R}^{d \times d \times d}$:

$$T = \sum_{i \in [k]} \lambda_i v_i \otimes v_i \otimes v_i.$$

Recall matrix power method: $v \mapsto \dfrac{M(I, v)}{\|M(I, v)\|}$.

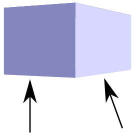
Orthogonal Tensor Power Method

Symmetric orthogonal tensor $T \in \mathbb{R}^{d\times d\times d}$:

$$T = \sum_{i\in[k]} \lambda_i v_i \otimes v_i \otimes v_i.$$

Recall matrix power method: $v \mapsto \frac{M(I,v)}{\|M(I,v)\|}$.

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



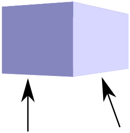
Orthogonal Tensor Power Method

Symmetric orthogonal tensor $T \in \mathbb{R}^{d \times d \times d}$:

$$T = \sum_{i \in [k]} \lambda_i v_i \otimes v_i \otimes v_i.$$

Recall matrix power method: $v \mapsto \dfrac{M(I, v)}{\|M(I, v)\|}$.

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



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

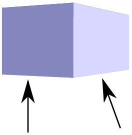
Orthogonal Tensor Power Method

Symmetric orthogonal tensor $T \in \mathbb{R}^{d\times d\times d}$:

$$T = \sum_{i\in[k]} \lambda_i v_i \otimes v_i \otimes v_i.$$

Recall matrix power method: $v \mapsto \dfrac{M(I,v)}{\|M(I,v)\|}$.

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



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

- All other eigenvectors are saddle points.

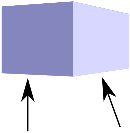
Orthogonal Tensor Power Method

Symmetric orthogonal tensor $T \in \mathbb{R}^{d \times d \times d}$:

$$T = \sum_{i \in [k]} \lambda_i v_i \otimes v_i \otimes v_i.$$

Recall matrix power method: $v \mapsto \dfrac{M(I,v)}{\|M(I,v)\|}$.

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



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

- All other eigenvectors are saddle points.

For an orthogonal tensor, no spurious local optima!

Matrix vs. tensor power iteration

Matrix power iteration:

Tensor power iteration:

Matrix vs. tensor power iteration

Matrix power iteration:

1. Requires gap between largest and second-largest eigenvalue.
   Property of the matrix only.

Tensor power iteration:

1. Requires gap between largest and second-largest $\lambda_i|c_i|$ where initialization vector $v = \sum_i c_i v_i$.
   Property of the tensor and initialization v.

Matrix vs. tensor power iteration

Matrix power iteration:

1. Requires gap between largest and second-largest eigenvalue.
   Property of the matrix only.

2. Converges to top eigenvector.

Tensor power iteration:

1. Requires gap between largest and second-largest $\lambda_i |c_i|$ where initialization vector $v = \sum_i c_i v_i$.
   Property of the tensor and initialization v.

2. Converges to v_i for which $v_i |c_i| = \max$! could be any of them.

Matrix vs. tensor power iteration

Matrix power iteration:

1. Requires gap between largest and second-largest eigenvalue.
   Property of the matrix only.

2. Converges to top eigenvector.

3. Linear convergence. Need $O(\log(1/\epsilon))$ iterations.

Tensor power iteration:

1. Requires gap between largest and second-largest $\lambda_i |c_i|$ where initialization vector $v = \sum_i c_i v_i$.
   Property of the tensor and initialization v.

2. Converges to v_i for which $v_i |c_i| = \max$! could be any of them.

3. Quadratic convergence. Need $O(\log \log(1/\epsilon))$ iterations.

Beyond Orthogonal Tensor Decomposition

Limitations

- Not ALL tensors have orthogonal decomposition (unlike matrices).
- Orthogonal forms: cannot handle overcomplete tensors $(k > d)$.
- Overcomplete representations: redundancy leads to flexible modeling, noise resistant, no domain knowledge.

Beyond Orthogonal Tensor Decomposition

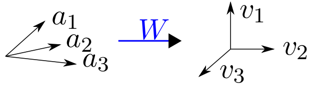
Limitations

- Not ALL tensors have orthogonal decomposition (unlike matrices).
- Orthogonal forms: cannot handle overcomplete tensors $(k > d)$.
- Overcomplete representations: redundancy leads to flexible modeling, noise resistant, no domain knowledge.

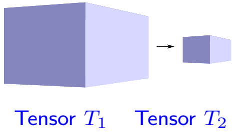
Undercomplete tensors $(k \leq d)$ with full rank components

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

- Whitening matrix W:



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



- Limitations: depends on condition number, sensitive to noise.

Beyond Orthogonal Tensor Decomposition

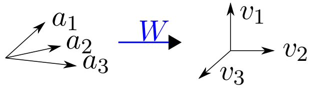
Limitations

- Not ALL tensors have orthogonal decomposition (unlike matrices).
- Orthogonal forms: cannot handle overcomplete tensors $(k > d)$.
- Overcomplete representations: redundancy leads to flexible modeling, noise resistant, no domain knowledge.

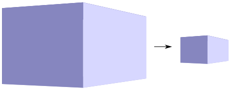
Undercomplete tensors $(k \leq d)$ with full rank components

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

- Limitations: depends on condition number, sensitive to noise.

This talk: guarantees for overcomplete tensor decomposition

Outline

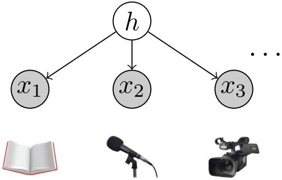
Non-orthogonal Tensor Decomposition

Multiview linear mixture model

- Linear model:

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

- $\mathbb{E}[x_1 \otimes x_2 \otimes x_3] = \sum_{i\in[k]} w_i a_i \otimes b_i \otimes c_i.$



Non-orthogonal Tensor Decomposition

$$\boxed{T = \sum_{i \in [k]} w_i a_i \otimes b_i \otimes c_i, \quad a_i, b_i, c_i \in \mathcal{S}^{d-1}.}$$

Practice: Alternating least squares (ALS)

- Many spurious local optima.
- No theoretical guarantee.

Non-orthogonal Tensor Decomposition

$$\boxed{T = \sum_{i \in [k]} w_i a_i \otimes b_i \otimes c_i, \quad a_i, b_i, c_i \in \mathcal{S}^{d-1}.}$$

Practice: Alternating least squares (ALS)

- Many spurious local optima.
- No theoretical guarantee.

Rank-1 ALS (Best Rank-1 Approximation)

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

Non-orthogonal Tensor Decomposition

$$\boxed{T = \sum_{i \in [k]} w_i a_i \otimes b_i \otimes c_i, \quad a_i, b_i, c_i \in \mathcal{S}^{d-1}.}$$

Practice: Alternating least squares (ALS)

- Many spurious local optima.
- No theoretical guarantee.

Rank-1 ALS (Best Rank-1 Approximation)

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

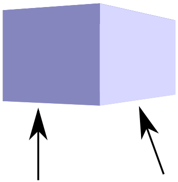
- Fix $a^{(t)}, b^{(t)}$ and update $c^{(t+1)} \implies$

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

$\boxed{\text{Rank-1 ALS iteration} \equiv \text{asymmetric power iteration}}$

Alternating minimization

Rank-1 ALS iteration (power iteration)



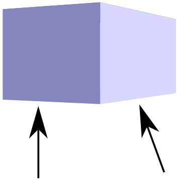
- Initialization: $a^{(0)}, b^{(0)}, c^{(0)}$.
- Update in t^{th} step: fix $a^{(t)}, b^{(t)}$ and

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

- After (approx.) convergence, restart.

Alternating minimization

Rank-1 ALS iteration (power iteration)



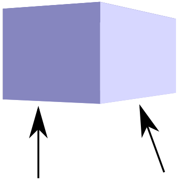
- Initialization: $a^{(0)}, b^{(0)}, c^{(0)}$.
- Update in t^{th} step: fix $a^{(t)}, b^{(t)}$ and

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

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

Alternating minimization

Rank-1 ALS iteration (power iteration)



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

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

- After (approx.) convergence, restart.

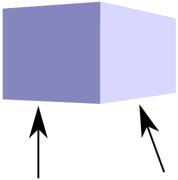
- Simple update: trivially parallelizable and hence scalable.
- Linear computation in dimension, rank, number of different runs.

Challenges

- Optimization problem: non-convex, multiple local optima.
- Alternating minimization: improves the objective in each step?
- Recovery of a_i, b_i, c_i's? Not true in general.
- Noisy tensor decomposition.

Alternating minimization

Rank-1 ALS iteration (power iteration)



- Initialization: $a^{(0)}, b^{(0)}, c^{(0)}$.
- Update in t^{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.

Challenges

- Optimization problem: non-convex, multiple local optima.
- Alternating minimization: improves the objective in each step?
- Recovery of a_i, b_i, c_i's? Not true in general.
- Noisy tensor decomposition.

Natural conditions under which Alt-Min has guarantees?

Special case: Orthogonal Setting

$$\boxed{T = \sum_{i \in [k]} w_i a_i \otimes b_i \otimes c_i, \quad a_i, b_i, c_i \in \mathcal{S}^{d-1}.}$$

- $\langle a_i, a_j \rangle = 0$, for $i \neq j$. Similarly for b, c.
- Alternating updates:

$$c^{(t+1)} \propto T(a^{(t)}, b^{(t)}, I) = \sum_{i \in [k]} w_i \langle a_i, a^{(t)} \rangle \langle b_i, b^{(t)} \rangle c_i.$$

- a_i, b_i, c_i are stationary points.

Special case: Orthogonal Setting

$$\boxed{T = \sum_{i \in [k]} w_i a_i \otimes b_i \otimes c_i, \quad a_i, b_i, c_i \in \mathcal{S}^{d-1}.}$$

- $\langle a_i, a_j \rangle = 0$, for $i \neq j$. Similarly for b, c.
- Alternating updates:

$$c^{(t+1)} \propto T(a^{(t)}, b^{(t)}, I) = \sum_{i \in [k]} w_i \langle a_i, a^{(t)} \rangle \langle b_i, b^{(t)} \rangle c_i.$$

- a_i, b_i, c_i are stationary points.
- ONLY local optima for best rank-1 approximation problem.
- Guaranteed recovery through alternating minimization.

Special case: Orthogonal Setting

$$\boxed{T = \sum_{i \in [k]} w_i a_i \otimes b_i \otimes c_i, \quad a_i, b_i, c_i \in \mathcal{S}^{d-1}.}$$

- $\langle a_i, a_j \rangle = 0$, for $i \neq j$. Similarly for b, c.
- Alternating updates:

$$c^{(t+1)} \propto T(a^{(t)}, b^{(t)}, I) = \sum_{i \in [k]} w_i \langle a_i, a^{(t)} \rangle \langle b_i, b^{(t)} \rangle c_i.$$

- a_i, b_i, c_i are stationary points.
- ONLY local optima for best rank-1 approximation problem.
- Guaranteed recovery through alternating minimization.
- Perturbation Analysis [AGH+2012]: Under poly(d) number of random initializations and bounded noise conditions.

Our Setup

So far

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

  Tractable cases? Covers overcomplete tensors?

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

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 (generic) vectors.

Guaranteed recovery for alternating minimization?

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

Analysis of One Step Update

$$\boxed{T = \sum_{i \in [k]} w_i a_i \otimes b_i \otimes c_i, \quad a_i, b_i, c_i \in \mathcal{S}^{d-1}.}$$

Basic Intuition

- Let $\hat{a}, \hat{b}$ be "close to" a_1, b_1. Alternating update:

$$\begin{aligned}\hat{c} \propto T(\hat{a}, \hat{b}, I) &= \sum_{i \in [k]} w_i \langle a_i, \hat{a} \rangle \langle b_i, \hat{b} \rangle c_i, \\ &= w_1 \langle a_1, \hat{a} \rangle \langle b_1, \hat{b} \rangle c_1 + T_{-1}(\hat{a}, \hat{b}, I).\end{aligned}$$

Analysis of One Step Update

$$\boxed{T = \sum_{i \in [k]} w_i a_i \otimes b_i \otimes c_i, \quad a_i, b_i, c_i \in \mathcal{S}^{d-1}.}$$

Basic Intuition

- Let $\hat{a}, \hat{b}$ be "close to" a_1, b_1. Alternating update:

$$\begin{aligned} \hat{c} \propto T(\hat{a}, \hat{b}, I) &= \sum_{i \in [k]} w_i \langle a_i, \hat{a} \rangle \langle b_i, \hat{b} \rangle c_i, \\ &= w_1 \langle a_1, \hat{a} \rangle \langle b_1, \hat{b} \rangle c_1 + T_{-1}(\hat{a}, \hat{b}, I). \end{aligned}$$

- $T_{-1}(\hat{a}, \hat{b}, I) = 0$ in orthogonal case, when $\hat{a} = a_1, \hat{b} = b_1$.

Analysis of One Step Update

$$\boxed{T = \sum_{i \in [k]} w_i a_i \otimes b_i \otimes c_i, \quad a_i, b_i, c_i \in \mathcal{S}^{d-1}.}$$

Basic Intuition

- Let $\hat{a}, \hat{b}$ be "close to" a_1, b_1. Alternating update:

$$\begin{aligned} \hat{c} \propto T(\hat{a}, \hat{b}, I) &= \sum_{i \in [k]} w_i \langle a_i, \hat{a} \rangle \langle b_i, \hat{b} \rangle c_i, \\ &= w_1 \langle a_1, \hat{a} \rangle \langle b_1, \hat{b} \rangle c_1 + T_{-1}(\hat{a}, \hat{b}, I). \end{aligned}$$

- $T_{-1}(\hat{a}, \hat{b}, I) = 0$ in orthogonal case, when $\hat{a} = a_1, \hat{b} = b_1$.
- Can it be controlled for incoherent (random) vectors?

Results for one step update

- Incoherence: $|\langle a_i, a_j \rangle| = O\left(1/\sqrt{d}\right)$ for $i \neq j$. Similarly for b, c.
- Spectral norm: $\|A\|, \|B\|, \|C\| \leq 1 + O\left(\sqrt{\frac{k}{d}}\right)$. $\|T\| \leq (1 + o(1))$.
- Tensor rank: $k = o(d^{1.5})$. Weights: For simplicity, $w_i \equiv 1$.

Results for one step update

- Incoherence: $|\langle a_i, a_j \rangle| = O\left(1/\sqrt{d}\right)$ for $i \neq j$. Similarly for b, c.
- Spectral norm: $\|A\|, \|B\|, \|C\| \leq 1 + O\left(\sqrt{\frac{k}{d}}\right)$. $\|T\| \leq (1 + o(1))$.
- Tensor rank: $k = o(d^{1.5})$. Weights: For simplicity, $w_i \equiv 1$.

Lemma [AGJ2014]

For small enough ϵ such that $\max\{\|a_1 - \hat{a}\|, \|b_1 - \hat{b}\|\} \leq \epsilon$, after one step

$$\boxed{\|c_1 - \hat{c}\| \leq O\left(\frac{\sqrt{k}}{d} + \max\left(\frac{1}{\sqrt{d}}, \frac{k}{d^{1.5}}\right)\epsilon + \epsilon^2\right).}$$

- $\dfrac{\sqrt{k}}{d}$: approximation error.
- rest: error contraction.

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: $k = o(d^{1.5})$.
- Recovery error: $\epsilon_R := \|E\| + \tilde{O}\left(\frac{\sqrt{k}}{d}\right)$

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: $k = o(d^{1.5})$.
- Recovery error: $\epsilon_R := \|E\| + \tilde{O}\left(\frac{\sqrt{k}}{d}\right)$

Theorem (Local Convergence)[AGJ2014]

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

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

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: $k = o(d^{1.5})$.
- Recovery error: $\epsilon_R := \|E\| + \tilde{O}\left(\frac{\sqrt{k}}{d}\right)$

Theorem (Local Convergence)[AGJ2014]

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

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

- Linear convergence: up to approximation error.
- Guarantees for overcomplete tensors: $k = o(d^{1.5})$ and for p^{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: $N = \Theta\left(\log(1/\epsilon_R)\right)$. Recall ϵ_R: 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: $N = \Theta\left(\log(1/\epsilon_R)\right)$. Recall ϵ_R: recovery error.

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

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: $N = \Theta\left(\log(1/\epsilon_R)\right)$. Recall ϵ_R: recovery error.

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

Corollary: Differing Dimensions

- If $a_i, b_i \in \mathbb{R}^{d_u}$ and $c_i \in \mathbb{R}^{d_o}$, and $d_u \geq k \geq d_o$.
- $k = O(\sqrt{d_u d_o})$ for incoherent vectors. $k = O(d_u)$ if A, B orthogonal.
- Same guarantees. Can handle one overcomplete mode.

Latest Result: Global Convergence

- Assume Gaussian means a_i's.
- Improved initialization requirement for convergence of third order tensor power iteration

$$|\langle a_1, \hat{a}^{(0)} \rangle| \geq d^{\beta} \frac{\sqrt{k}}{d}, \quad \beta > (\log d)^{-c}.$$

Spherical Gaussian Mixture or Multiview Mixture Model

- Initialize with samples with norm of noise bounded by $\sqrt{d}\sigma$ such that

$$\sigma = o\left(\sqrt{\frac{d}{k}}\right).$$

"Analyzing Tensor Power Method Dynamics: Applications to Learning Overcomplete Latent Variable Models" by A. Anandkumar, R. Ge. and M. Janzamin, Nov. 2014.

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: $\|\hat{T} - T\| \leq \| \operatorname{mat}(\hat{T}) - \operatorname{mat}(T)\|$, 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 ϵ-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 $\displaystyle \frac{1}{n} \sum_i \langle z_1^i, u \rangle \langle z_2^i, v \rangle \langle z_3^i, w \rangle$, for all $u, v, w \in \mathcal{S}^{d-1}$.
- Classify inner products into buckets and bound them separately.

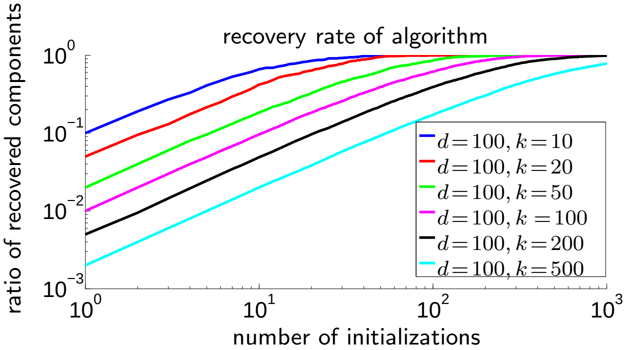
Tight sample bounds for a range of latent variable models

"Provable Learning of Overcomplete Latent Variable Models: Semi-supervised and Unsupervised Settings" by A. Anandkumar, R. Ge. and M. Janzamin, Aug. 2014.

Outline

Synthetic experiments

- Learning multiview Gaussian mixture.
- Random mixture components.
- $d = 100,\ k = \{10, 20, 50, 100, 200, 500\}$.
- $n = 1000$.
- Random initialization.



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.

- Coming: removing approximation error $\frac{\sqrt{k}}{d}$.

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.
- Coming: removing approximation error $\frac{\sqrt{k}}{d}$.

Thank you!