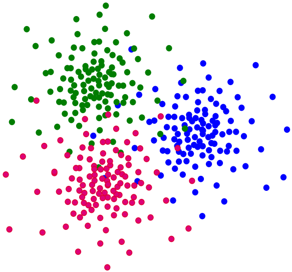
Guaranteed Learning of Latent Variable Models through Spectral and Tensor Methods

Anima Anandkumar

U.C. Irvine

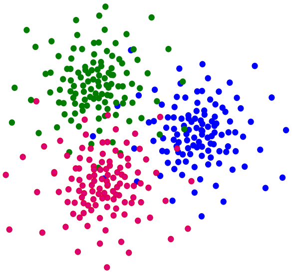
Application 1: Clustering

- Basic operation of grouping data points.
- Hypothesis: each data point belongs to an unknown group.



Application 1: Clustering

- Basic operation of grouping data points.
- Hypothesis: each data point belongs to an unknown group.



Probabilistic/latent variable viewpoint

- The groups represent different distributions. (e.g. Gaussian).
- Each data point is drawn from one of the given distributions. (e.g. Gaussian mixtures).

Application 2: Topic Modeling

Document modeling

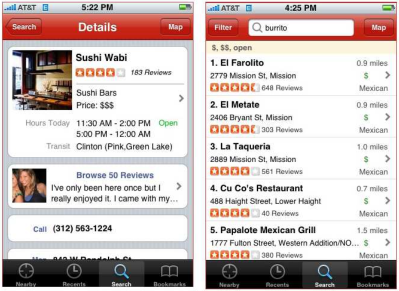
- Observed: words in document corpus.
- Hidden: topics.
- Goal: carry out document summarization.

Application 3: Understanding Human Communities

Social Networks

- Observed: network of social ties, e.g. friendships, co-authorships
- Hidden: groups/communities of actors.

Application 4: Recommender Systems



Recommender System

- Observed: Ratings of users for various products, e.g. yelp reviews.
- Goal: Predict new recommendations.
- Modeling: Find groups/communities of users and products.

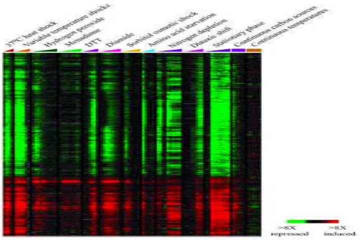
Application 5: Feature Learning

$$\begin{array}{c}
\text{Label} \qquad\qquad \text{Features} \\
\left[\begin{array}{c|ccccc}
0 & 2.1 & 5.2 & 0 & 0 & \cdots \\
1 & 0 & 0 & 2 & 1 & \cdots \\
1 & 1.1 & 0 & 0 & 0 & \cdots \\
0 & 0 & 0 & 7 & 0 & \cdots \\
\vdots & \vdots & \vdots & \vdots & \vdots &
\end{array}\right]
\end{array}$$

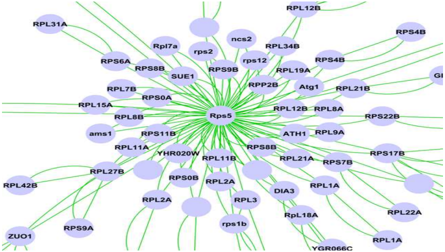
Feature Engineering

- Learn good features/representations for classification tasks, e.g. image and speech recognition.
- Sparse representations, low dimensional hidden structures.

Application 6: Computational Biology



Gasch *et al.* *Mol Biol Cell* 2000.

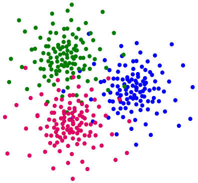


- Observed: gene expression levels
- Goal: discover gene groups
- Hidden variables: regulators controlling gene groups

How to model hidden effects?

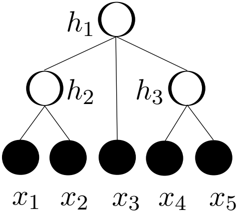
Basic Approach: mixtures/clusters

- Hidden variable h is categorical.



Advanced: Probabilistic models

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



This talk: basic mixture model. Tomorrow: advanced models.

Challenges in Learning

Basic goal in all mentioned applications

Discover hidden structure in data: unsupervised learning.

Challenges in Learning

Basic goal in all mentioned applications

Discover hidden structure in data: unsupervised learning.

Challenge: Conditions for Identifiability

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

Challenges in Learning

Basic goal in all mentioned applications

Discover hidden structure in data: unsupervised learning.

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.
- Practice: EM, Variational Bayes have no consistency guarantees.
- Efficient computational and sample complexities?

Challenges in Learning

Basic goal in all mentioned applications

Discover hidden structure in data: unsupervised learning.

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.
- Practice: EM, Variational Bayes have no consistency guarantees.
- Efficient computational and sample complexities?

In this series: guaranteed and efficient learning through spectral methods

What this talk series will cover

This talk

- Start with the most basic latent variable model: Gaussian mixtures.

What this talk series will cover

This talk

- Start with the most basic latent variable model: Gaussian mixtures.
- Basic spectral approach: principal component analysis (PCA).

What this talk series will cover

This talk

- Start with the most basic latent variable model: Gaussian mixtures.
- Basic spectral approach: principal component analysis (PCA).
- Beyond correlations: higher order moments.

What this talk series will cover

This talk

- Start with the most basic latent variable model: Gaussian mixtures.
- Basic spectral approach: principal component analysis (PCA).
- Beyond correlations: higher order moments.
- Tensor notations, PCA extension to tensors.

What this talk series will cover

This talk

- Start with the most basic latent variable model: Gaussian mixtures.
- Basic spectral approach: principal component analysis (PCA).
- Beyond correlations: higher order moments.
- Tensor notations, PCA extension to tensors.
- Guaranteed learning of Gaussian mixtures.

What this talk series will cover

This talk

- Start with the most basic latent variable model: Gaussian mixtures.
- Basic spectral approach: principal component analysis (PCA).
- Beyond correlations: higher order moments.
- Tensor notations, PCA extension to tensors.
- Guaranteed learning of Gaussian mixtures.
- Introduce topic models. Present a unified viewpoint.

What this talk series will cover

This talk

- Start with the most basic latent variable model: Gaussian mixtures.
- Basic spectral approach: principal component analysis (PCA).
- Beyond correlations: higher order moments.
- Tensor notations, PCA extension to tensors.
- Guaranteed learning of Gaussian mixtures.
- Introduce topic models. Present a unified viewpoint.

- Tomorrow: using tensor methods for learning latent variable models and analysis of tensor decomposition method.

What this talk series will cover

This talk

- Start with the most basic latent variable model: Gaussian mixtures.
- Basic spectral approach: principal component analysis (PCA).
- Beyond correlations: higher order moments.
- Tensor notations, PCA extension to tensors.
- Guaranteed learning of Gaussian mixtures.
- Introduce topic models. Present a unified viewpoint.

- Tomorrow: using tensor methods for learning latent variable models and analysis of tensor decomposition method.
- Wednesday: implementation of tensor methods.

Resources for Today's Talk

- "A Method of Moments for Mixture Models and Hidden Markov Models." by A. , D. Hsu, and S.M. Kakade. Proc. of COLT, June 2012.
- "Tensor Decompositions for Learning Latent Variable Models." by A., R. Ge, D. Hsu, S.M. Kakade and M. Telgarsky, Oct. 2012.
- Resources available at `http://newport.eecs.uci.edu/anandkumar/MLSS.html`

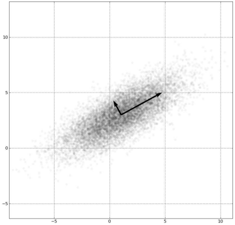
Outline

Warm-up: PCA

Optimization problem

For (centered) points $x_i \in \mathbb{R}^d$, find projection P with $\text{Rank}(P) = k$ s.t.

$$\boxed{\min_{P \in \mathbb{R}^{d \times d}} \frac{1}{n} \sum_{i \in [n]} \|x_i - P x_i\|^2.}$$

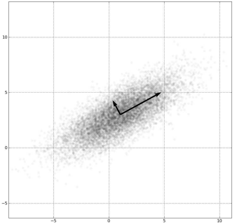


Warm-up: PCA

Optimization problem

For (centered) points $x_i \in \mathbb{R}^d$, find projection P with Rank$(P) = k$ s.t.

$$\boxed{\min_{P \in \mathbb{R}^{d \times d}} \frac{1}{n} \sum_{i \in [n]} \|x_i - P x_i\|^2.}$$



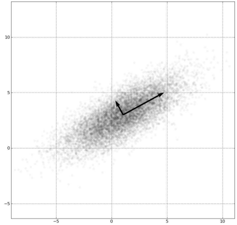
Result: If $S = \mathrm{Cov}(X)$ and $S = U \Lambda U^\top$ is eigen decomposition, we have $P = U_{(k)} U_{(k)}^\top$, where $U_{(k)}$ are top-k eigen vectors.

Warm-up: PCA

Optimization problem

For (centered) points $x_i \in \mathbb{R}^d$, find projection P with $\text{Rank}(P) = k$ s.t.

$$\boxed{\min_{P \in \mathbb{R}^{d\times d}} \frac{1}{n} \sum_{i\in[n]} \|x_i - Px_i\|^2.}$$



Result: If $S = \text{Cov}(X)$ and $S = U\Lambda U^\top$ is eigen decomposition, we have $P = U_{(k)}U_{(k)}^\top$, where $U_{(k)}$ are top-k eigen vectors.

Proof

- By Pythagorean theorem: $\sum_i \|x_i - Px_i\|^2 = \sum_i \|x_i\|^2 - \sum_i \|Px_i\|^2$.
- Maximize: $\frac{1}{n}\sum_i \|Px_i\|^2 = \frac{1}{n}\sum_i \text{Tr}\left[Px_ix_i^\top P^\top\right] = \text{Tr}[PSP^\top]$.

Review of linear algebra

For a matrix S, u is an eigenvector if $Su = \lambda u$ and λ is eigenvalue.

- For symm. $S \in \mathbb{R}^{d \times d}$, there are d eigen values.
- $S = \sum_{i \in [d]} \lambda_i u_i u_i^\top$. U is orthogonal.

Review of linear algebra

For a matrix S, u is an eigenvector if $Su = \lambda u$ and λ is eigenvalue.

- For symm. $S \in \mathbb{R}^{d \times d}$, there are d eigen values.
- $S = \sum_{i \in [d]} \lambda_i u_i u_i^\top$. U is orthogonal.

Rayleigh Quotient

For matrix S with eigenvalues $\lambda_1 \geq \lambda_2 \ldots \lambda_d$ and corresponding eigenvectors $u_1, \ldots u_d$, then

$$\max_{\|z\|=1} z^\top S z = \lambda_1, \quad \min_{\|z\|=1} z^\top S z = \lambda_d,$$

and the optimizing vectors are u_1 and u_d.

Review of linear algebra

For a matrix S, u is an eigenvector if $Su = \lambda u$ and λ is eigenvalue.

- For symm. $S \in \mathbb{R}^{d\times d}$, there are d eigen values.
- $S = \sum_{i\in[d]} \lambda_i u_i u_i^\top$. U is orthogonal.

Rayleigh Quotient

For matrix S with eigenvalues $\lambda_1 \geq \lambda_2 \ldots \lambda_d$ and corresponding eigenvectors $u_1, \ldots u_d$, then

$$\max_{\|z\|=1} z^\top S z = \lambda_1, \quad \min_{\|z\|=1} z^\top S z = \lambda_d,$$

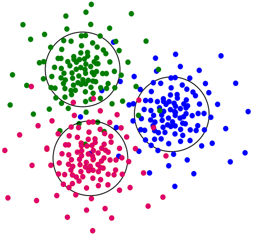
and the optimizing vectors are u_1 and u_d.

Optimal Projection

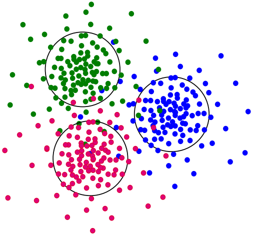
$$\max_{\substack{P: P^2 = I \\ \text{Rank}(P)=k}} \text{Tr}(P^\top S P) = \lambda_1 + \lambda_2 \ldots + \lambda_k \text{ and } P \text{ spans } \{u_1, \ldots, u_k\}.$$

PCA on Gaussian Mixtures

- k Gaussians: each sample is $x = Ah + z$.
- $h \in [e_1, \ldots, e_k]$, the basis vectors. $\mathbb{E}[h] = w$.
- $A \in \mathbb{R}^{d \times k}$: columns are component means.
- Let $\mu := Aw$ be the mean.
- $z \sim \mathcal{N}(0, \sigma^2 I)$ is white Gaussian noise.



PCA on Gaussian Mixtures

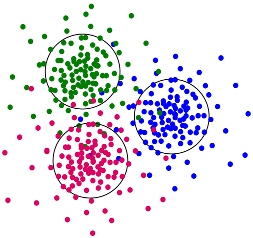


- k Gaussians: each sample is $x = Ah + z$.
- $h \in [e_1, \ldots, e_k]$, the basis vectors. $\mathbb{E}[h] = w$.
- $A \in \mathbb{R}^{d \times k}$: columns are component means.
- Let $\mu := Aw$ be the mean.
- $z \sim \mathcal{N}(0, \sigma^2 I)$ is white Gaussian noise.

$$\boxed{\mathbb{E}[(x-\mu)(x-\mu)^\top] = \sum_{i \in [k]} w_i (a_i - \mu)(a_i - \mu)^\top + \sigma^2 I.}$$

PCA on Gaussian Mixtures

- k Gaussians: each sample is $x = Ah + z$.
- $h \in [e_1, \ldots, e_k]$, the basis vectors. $\mathbb{E}[h] = w$.
- $A \in \mathbb{R}^{d \times k}$: columns are component means.
- Let $\mu := Aw$ be the mean.
- $z \sim \mathcal{N}(0, \sigma^2 I)$ is white Gaussian noise.



$$\boxed{\mathbb{E}[(x-\mu)(x-\mu)^\top] = \sum_{i \in [k]} w_i (a_i - \mu)(a_i - \mu)^\top + \sigma^2 I.}$$

How the above equation is obtained

$$\begin{aligned}\mathbb{E}[(x-\mu)(x-\mu)^\top] &= \mathbb{E}[(Ah-\mu)(Ah-\mu)^\top] + \mathbb{E}[zz^\top] \\ &= \sum_{i \in [k]} w_i (a_i - \mu)(a_i - \mu)^\top + \sigma^2 I.\end{aligned}$$

PCA on Gaussian Mixtures

$$\boxed{\mathbb{E}[(x-\mu)(x-\mu)^\top] = \sum_{i\in[k]} w_i(a_i-\mu)(a_i-\mu)^\top + \sigma^2 I.}$$

- The vectors $\{a_i - \mu\}$ are linearly dependent: $\sum_i w_i(a_i - \mu) = 0$. The PSD matrix $\sum_{i\in[k]} w_i(a_i-\mu)(a_i-\mu)^\top$ has rank $\leq k-1$.

PCA on Gaussian Mixtures

$$\boxed{\mathbb{E}[(x-\mu)(x-\mu)^\top] = \sum_{i\in[k]} w_i(a_i-\mu)(a_i-\mu)^\top + \sigma^2 I.}$$

- The vectors $\{a_i - \mu\}$ are linearly dependent: $\sum_i w_i(a_i-\mu) = 0$. The PSD matrix $\sum_{i\in[k]} w_i(a_i-\mu)(a_i-\mu)^\top$ has rank $\leq k-1$.
- $(k-1)$-PCA on covariance matrix $\cup\{\mu\}$ yields $\text{span}(A)$.
- Lowest eigenvalue of covariance matrix yields σ^2.

PCA on Gaussian Mixtures

$$\boxed{\mathbb{E}[(x-\mu)(x-\mu)^\top] = \sum_{i\in[k]} w_i(a_i-\mu)(a_i-\mu)^\top + \sigma^2 I.}$$

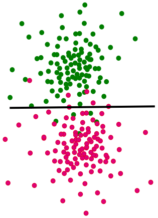
- The vectors $\{a_i - \mu\}$ are linearly dependent: $\sum_i w_i(a_i-\mu) = 0$. The PSD matrix $\sum_{i\in[k]} w_i(a_i-\mu)(a_i-\mu)^\top$ has rank $\leq k-1$.
- $(k-1)$-PCA on covariance matrix $\cup\{\mu\}$ yields span(A).
- Lowest eigenvalue of covariance matrix yields σ^2.

How to Learn A?

Learning through Spectral Clustering

Learning A through Spectral Clustering

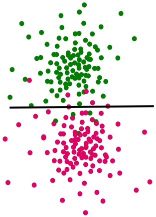
- Project samples x on to span(A).
- Distance-based clustering (e.g. k-means).
- A series of works, e.g. Vempala & Wang.



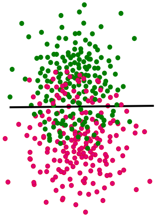
Learning through Spectral Clustering

Learning A through Spectral Clustering

- Project samples x on to span(A).
- Distance-based clustering (e.g. k-means).
- A series of works, e.g. Vempala & Wang.



Failure to cluster under large variance.



Learning Gaussian Mixtures Without Separation Constraints?

Beyond PCA: Spectral Methods on Tensors

- How to learn the component means A (not just its span) without separation constraints?

Beyond PCA: Spectral Methods on Tensors

- How to learn the component means A (not just its span) without separation constraints?
- PCA is a spectral method on (covariance) matrices.
  - Are higher order moments helpful?

Beyond PCA: Spectral Methods on Tensors

- How to learn the component means A (not just its span) without separation constraints?
- PCA is a spectral method on (covariance) matrices.
  - Are higher order moments helpful?
- What if number of components is greater than observed dimensionality $k > d$?

Beyond PCA: Spectral Methods on Tensors

- How to learn the component means A (not just its span) without separation constraints?
- PCA is a spectral method on (covariance) matrices.
  - Are higher order moments helpful?
- What if number of components is greater than observed dimensionality $k > d$?
  - Do higher order moments help to learn overcomplete models?

Beyond PCA: Spectral Methods on Tensors

- How to learn the component means A (not just its span) without separation constraints?
- PCA is a spectral method on (covariance) matrices.
  - Are higher order moments helpful?
- What if number of components is greater than observed dimensionality $k > d$?
  - Do higher order moments help to learn overcomplete models?
- What if the data is not Gaussian?

Beyond PCA: Spectral Methods on Tensors

- How to learn the component means A (not just its span) without separation constraints?
- PCA is a spectral method on (covariance) matrices.
  - Are higher order moments helpful?
- What if number of components is greater than observed dimensionality $k > d$?
  - Do higher order moments help to learn overcomplete models?
- What if the data is not Gaussian?
  - Moment-based Estimation of probabilistic latent variable models?

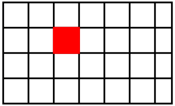
Outline

Tensor Notation for Higher Order Moments

- Multi-variate higher order moments form tensors.
- Are there spectral operations on tensors akin to PCA?

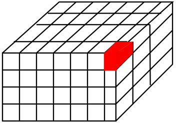
Matrix

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



Third order moment for Gaussian mixtures

- Consider mixture of k Gaussians: each sample is $x = Ah + z$.
- $h \in [e_1, \ldots, e_k]$, the basis vectors. $\mathbb{E}[h] = w$.
- $A \in \mathbb{R}^{d \times k}$: columns are component means. $\mu := Aw$ be the mean.
- $z \sim \mathcal{N}(0, \sigma^2 I)$ is white Gaussian noise.

$$\mathbb{E}[x \otimes x \otimes x] = \sum_i w_i a_i \otimes a_i \otimes a_i + \sigma^2 \sum_i (\mu \otimes e_i \otimes e_i + \ldots)$$

Intuition behind equation

$$\begin{aligned}\mathbb{E}[x \otimes x \otimes x] &= \mathbb{E}[(Ah) \otimes (Ah) \otimes (Ah)] + \mathbb{E}[(Ah) \otimes z \otimes z] + \ldots \\ &= \sum_i w_i \cdot a_i \otimes a_i \otimes a_i + \sigma^2 \sum_i \mu \otimes e_i \otimes e_i + \ldots\end{aligned}$$

How to recover parameters A and w from third order moment?

Simplifications

$$\mathbb{E}[x \otimes x \otimes x] = \sum_i w_i a_i \otimes a_i \otimes a_i + \sigma^2 \sum_i (\mu \otimes e_i \otimes e_i + \dots)$$

- σ^2 is obtained from $\sigma_{\min}(\mathrm{Cov}(x))$.
- $\mathbb{E}[x \otimes x] = \sum_i w_i a_i \otimes a_i + \sigma^2 I$.

Simplifications

$$\mathbb{E}[x \otimes x \otimes x] = \sum_i w_i a_i \otimes a_i \otimes a_i + \sigma^2 \sum_i (\mu \otimes e_i \otimes e_i + \ldots)$$

- σ^2 is obtained from $\sigma_{\min}(\mathrm{Cov}(x))$.
- $\mathbb{E}[x \otimes x] = \sum_i w_i a_i \otimes a_i + \sigma^2 I$.

Can obtain

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

$$M_2 = \sum_i w_i a_i \otimes a_i.$$

How to obtain parameters A and w from M_2 and M_3?

Tensor Slices

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

Multilinear transformation of tensor

$$\boxed{M_3(B, C, D) := \sum_i w_i (B^\top a_i) \cdot (C^\top a_i) \cdot (D^\top a_i)}$$

Tensor Slices

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

Multilinear transformation of tensor

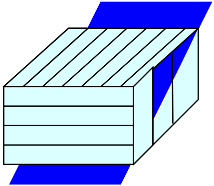
$$\boxed{M_3(B, C, D) := \sum_i w_i (B^\top a_i) \cdot (C^\top a_i) \cdot (D^\top a_i)}$$

Slice of a tensor

$M_3(I, I, r) = \sum_i w_i a_i \otimes a_i \langle a_i, r\rangle = A\mathrm{Diag}(w)\mathrm{Diag}(A^\top r)A^\top$

$$M_3(I, I, r) = A \cdot \mathrm{Diag}(w)\mathrm{Diag}(A^\top r) \cdot A^\top$$

$$M_2 = A \cdot \mathrm{Diag}(w) \cdot A^\top$$



Eigen-decomposition

$$\boxed{M_3(I, I, r) = A \cdot \mathsf{Diag}(w)\mathsf{Diag}(A^\top r) \cdot A^\top, \quad M_2 = A \cdot \mathsf{Diag}(w) \cdot A^\top.}$$

Eigen-decomposition

$$\boxed{M_3(I, I, r) = A \cdot \mathrm{Diag}(w)\mathrm{Diag}(A^\top r) \cdot A^\top, \quad M_2 = A \cdot \mathrm{Diag}(w) \cdot A^\top.}$$

Assumption: $A \in \mathbb{R}^{d \times k}$ has full column rank.

Eigen-decomposition

$$\boxed{M_3(I,I,r) = A \cdot \mathsf{Diag}(w)\mathsf{Diag}(A^\top r) \cdot A^\top, \quad M_2 = A \cdot \mathsf{Diag}(w) \cdot A^\top.}$$

Assumption: $A \in \mathbb{R}^{d\times k}$ has full column rank.

- $M_2 = U\Lambda U^\top$ be eigen-decomposition. $U \in \mathbb{R}^{d\times k}$.
- $U^\top M_2 U = \Lambda \in \mathbb{R}^{k\times k}$ is invertible.

Eigen-decomposition

$$\boxed{M_3(I,I,r) = A \cdot \mathrm{Diag}(w)\mathrm{Diag}(A^\top r) \cdot A^\top, \quad M_2 = A \cdot \mathrm{Diag}(w) \cdot A^\top.}$$

Assumption: $A \in \mathbb{R}^{d\times k}$ has full column rank.

- $M_2 = U\Lambda U^\top$ be eigen-decomposition. $U \in \mathbb{R}^{d\times k}$.
- $U^\top M_2 U = \Lambda \in \mathbb{R}^{k\times k}$ is invertible.

$$\boxed{X = \left(U^\top M_3(I,I,r)U\right)\left(U^\top M_2 U\right)^{-1} = V \cdot \mathrm{Diag}(\tilde{\lambda}) \cdot V^{-1}.}$$

Eigen-decomposition

$$\boxed{M_3(I,I,r) = A \cdot \mathrm{Diag}(w)\mathrm{Diag}(A^\top r) \cdot A^\top, \quad M_2 = A \cdot \mathrm{Diag}(w) \cdot A^\top.}$$

Assumption: $A \in \mathbb{R}^{d\times k}$ has full column rank.

- $M_2 = U\Lambda U^\top$ be eigen-decomposition. $U \in \mathbb{R}^{d\times k}$.
- $U^\top M_2 U = \Lambda \in \mathbb{R}^{k\times k}$ is invertible.

$$\boxed{X = \left(U^\top M_3(I,I,r)U\right)\left(U^\top M_2 U\right)^{-1} = V \cdot \mathrm{Diag}(\tilde{\lambda}) \cdot V^{-1}.}$$

- Substitution: $X = (U^\top A)\mathrm{Diag}(A^\top r)(U^\top A)^{-1}$.
- We have $\boxed{v_i \propto U^\top a_i}$.

Eigen-decomposition

$$\boxed{M_3(I,I,r) = A \cdot \mathrm{Diag}(w)\mathrm{Diag}(A^\top r) \cdot A^\top, \quad M_2 = A \cdot \mathrm{Diag}(w) \cdot A^\top.}$$

Assumption: $A \in \mathbb{R}^{d\times k}$ has full column rank.

- $M_2 = U\Lambda U^\top$ be eigen-decomposition. $U \in \mathbb{R}^{d\times k}$.
- $U^\top M_2 U = \Lambda \in \mathbb{R}^{k\times k}$ is invertible.

$$\boxed{X = \left(U^\top M_3(I,I,r)U\right)\left(U^\top M_2 U\right)^{-1} = V \cdot \mathrm{Diag}(\tilde{\lambda}) \cdot V^{-1}.}$$

- Substitution: $X = (U^\top A)\mathrm{Diag}(A^\top r)(U^\top A)^{-1}$.
- We have $\boxed{v_i \propto U^\top a_i}$.

Technical Detail

$r = U\theta$ and θ drawn uniformly from sphere to ensure eigen gap.

Learning Gaussian Mixtures through Eigen-decomposition of Tensor Slices

$$\boxed{M_3(I, I, r) = A \cdot \mathrm{Diag}(w)\mathrm{Diag}(A^\top r) \cdot A^\top, \quad M_2 = A \cdot \mathrm{Diag}(w) \cdot A^\top.}$$

$$\boxed{\left(U^\top M_3(I, I, r) U\right) \left(U^\top M_2 U\right)^{-1} = V \mathrm{Diag}(\tilde{\lambda}) V^{-1}.}$$

Recovery of A (method 1)

- Since $v_i \propto U^\top a_i$, recover A upto scale: $\boxed{a_i \propto U v_i}$.

Learning Gaussian Mixtures through Eigen-decomposition of Tensor Slices

$$\boxed{M_3(I,I,r) = A \cdot \text{Diag}(w)\text{Diag}(A^\top r) \cdot A^\top, \quad M_2 = A \cdot \text{Diag}(w) \cdot A^\top.}$$

$$\boxed{\left(U^\top M_3(I,I,r)U\right)\left(U^\top M_2 U\right)^{-1} = V\text{Diag}(\tilde{\lambda})V^{-1}.}$$

Recovery of A (method 1)

- Since $v_i \propto U^\top a_i$, recover A upto scale: $\boxed{a_i \propto U v_i}$.

Recovery of A (method 2)

- Let $\Theta \in \mathbb{R}^{k \times k}$ be a random rotation matrix.
- Consider k slices $M_3(I,I,\theta_i)$ and find eigen-decomposition above.
- Let $\tilde{\Lambda} = [\tilde{\lambda}_1|\ldots|\tilde{\lambda}_k]$ be the matrix of eigenvalues of all slices.
- Recover A as : $A = U\Theta^{-1}\tilde{\Lambda}$.

Putting it together

Implications

- Learn Gaussian mixtures through slices of third-order moment.
- Guaranteed learning through eigen decomposition.

"A Method of Moments for Mixture Models and Hidden Markov Models." by A. , D. Hsu, and S.M. Kakade. Proc. of COLT, June 2012.

Putting it together

Implications

- Learn Gaussian mixtures through slices of third-order moment.
- Guaranteed learning through eigen decomposition.

"A Method of Moments for Mixture Models and Hidden Markov Models." by A. , D. Hsu, and S.M. Kakade. Proc. of COLT, June 2012.

Shortcomings

- The resulting product is not symmetric. Eigen-decomposition $V\text{Diag}(\tilde{\lambda})V^{-1}$ does not result in orthonormal V. More involved in practice.

Putting it together

Implications

- Learn Gaussian mixtures through slices of third-order moment.
- Guaranteed learning through eigen decomposition.

"A Method of Moments for Mixture Models and Hidden Markov Models." by A. , D. Hsu, and S.M. Kakade. Proc. of COLT, June 2012.

Shortcomings

- The resulting product is not symmetric. Eigen-decomposition $V\text{Diag}(\tilde{\lambda})V^{-1}$ does not result in orthonormal V. More involved in practice.
- Require good eigen-gap in $\text{Diag}(\tilde{\lambda})$ for recovery. For $r = U\theta$, where θ is drawn uniformly from unit sphere, gap is $1/k^{2.5}$. Numerical instability in practice.

Putting it together

Implications

- Learn Gaussian mixtures through slices of third-order moment.
- Guaranteed learning through eigen decomposition.

"A Method of Moments for Mixture Models and Hidden Markov Models." by A. , D. Hsu, and S.M. Kakade. Proc. of COLT, June 2012.

Shortcomings

- The resulting product is not symmetric. Eigen-decomposition $V\text{Diag}(\tilde{\lambda})V^{-1}$ does not result in orthonormal V. More involved in practice.
- Require good eigen-gap in $\text{Diag}(\tilde{\lambda})$ for recovery. For $r = U\theta$, where θ is drawn uniformly from unit sphere, gap is $1/k^{2.5}$. Numerical instability in practice.
- $M_3(I, I, r)$ is only a (random) slice of the tensor. Full information is not utilized.

Putting it together

Implications

- Learn Gaussian mixtures through slices of third-order moment.
- Guaranteed learning through eigen decomposition.

"A Method of Moments for Mixture Models and Hidden Markov Models." by A. , D. Hsu, and S.M. Kakade. Proc. of COLT, June 2012.

Shortcomings

- The resulting product is not symmetric. Eigen-decomposition $V\text{Diag}(\tilde{\lambda})V^{-1}$ does not result in orthonormal V. More involved in practice.
- Require good eigen-gap in $\text{Diag}(\tilde{\lambda})$ for recovery. For $r = U\theta$, where θ is drawn uniformly from unit sphere, gap is $1/k^{2.5}$. Numerical instability in practice.
- $M_3(I, I, r)$ is only a (random) slice of the tensor. Full information is not utilized.

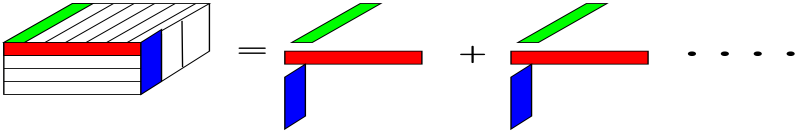
More efficient learning methods using higher order moments?

Outline

Tensor Factorization

- Recover A and w from M_2 and M_3.

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

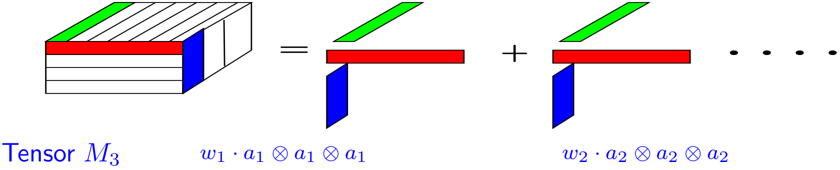


- $a \otimes a \otimes a$ is a rank-1 tensor since, its $(i_1, i_2, i_3)^{\text{th}}$ entry is $a_{i_1} a_{i_2} a_{i_3}$.
- M_3 is a sum of rank-1 terms.

Tensor Factorization

- Recover A and w from M_2 and M_3.

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

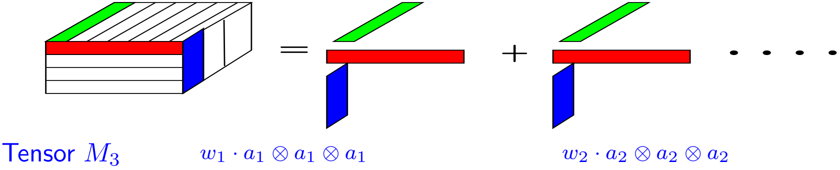


- $a \otimes a \otimes a$ is a rank-1 tensor since, its $(i_1, i_2, i_3)^{\text{th}}$ entry is $a_{i_1} a_{i_2} a_{i_3}$.
- M_3 is a sum of rank-1 terms.
- When is it the most compact representation? (Identifiability).
- Can we recover the decomposition? (Algorithm?)

Tensor Factorization

- Recover A and w from M_2 and M_3.

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



- $a \otimes a \otimes a$ is a rank-1 tensor since, its $(i_1, i_2, i_3)^{\text{th}}$ entry is $a_{i_1} a_{i_2} a_{i_3}$.
- M_3 is a sum of rank-1 terms.
- When is it the most compact representation? (Identifiability).
- Can we recover the decomposition? (Algorithm?)
- The most compact representation is known as CP-decomposition (CANDECOMP/PARAFAC).

Initial Ideas

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

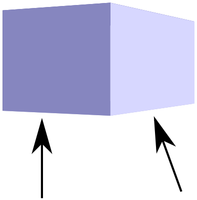
What if A has orthogonal columns?

Initial Ideas

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

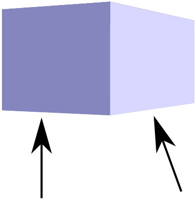
What if A has orthogonal columns?

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



Initial Ideas

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



What if A has orthogonal columns?

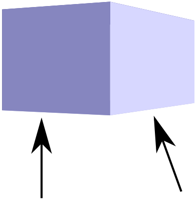
- $M_3(I, a_1, a_1) = \sum_i w_i \langle a_i, a_1 \rangle^2 a_i = w_1 a_1.$
- a_i are eigenvectors of tensor M_3.
- Analogous to matrix eigenvectors: $Mv = M(I, v) = \lambda v.$

Initial Ideas

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

What if A has orthogonal columns?

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



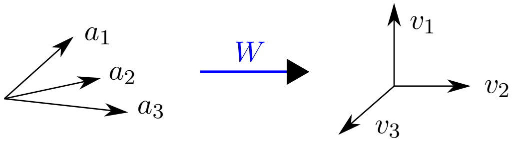
Two Problems

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

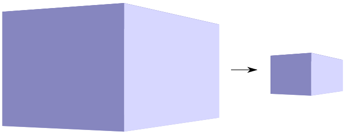
Whitening

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

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



Using Whitening to Obtain Orthogonal Tensor



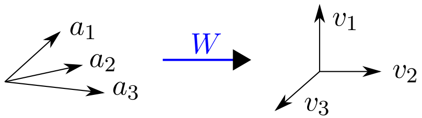
Tensor M_3 Tensor T

Multi-linear transform

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

How to Find Whitening Matrix?

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



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

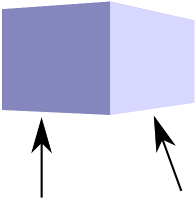
$$T = M_3(W,W,W) = \sum_i w_i^{-1/2}(W^\top a_i \sqrt{w_i})^{\otimes 3}$$
$$= \sum_i \lambda_i v_i \otimes v_i \otimes v_i, \quad \lambda_i := w_i^{-1/2}.$$

T is an orthogonal tensor.

Orthogonal Tensor Eigen Decomposition

$$T = \sum_{i \in [k]} \lambda_i v_i \otimes v_i \otimes v_i, \quad \langle v_i, v_j \rangle = \delta_{i,j}, \ \forall\, i, j.$$

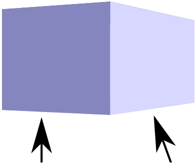
- $T(I, v_1, v_1) = \sum_i \lambda_i \langle v_i, v_1 \rangle^2 v_i = \lambda_1 v_1.$
- v_i are eigenvectors of tensor T.



Orthogonal Tensor Eigen Decomposition

$$T = \sum_{i \in [k]} \lambda_i v_i \otimes v_i \otimes v_i, \quad \langle v_i, v_j \rangle = \delta_{i,j}, \ \forall\, i, j.$$

- $T(I, v_1, v_1) = \sum_i \lambda_i \langle v_i, v_1 \rangle^2 v_i = \lambda_1 v_1$.
- v_i are eigenvectors of tensor T.



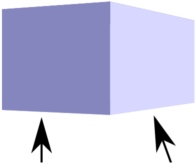
Tensor Power Method

- Start from an initial vector v. $\boxed{v \mapsto \frac{T(I, v, v)}{\|T(I, v, v)\|}}$.

Orthogonal Tensor Eigen Decomposition

$$T = \sum_{i \in [k]} \lambda_i v_i \otimes v_i \otimes v_i, \quad \langle v_i, v_j \rangle = \delta_{i,j}, \ \forall\, i, j.$$

- $T(I, v_1, v_1) = \sum_i \lambda_i \langle v_i, v_1 \rangle^2 v_i = \lambda_1 v_1.$
- v_i are eigenvectors of tensor T.



Tensor Power Method

- Start from an initial vector v. $\boxed{v \mapsto \frac{T(I, v, v)}{\|T(I, v, v)\|}.}$

Canonical recovery method

- Randomly initialize the power method. Run to convergence to obtain v with eigenvalue λ.
- Deflate: $T - \lambda v \otimes v \otimes v$ and repeat.

Putting it together

- Gaussian mixture: $x = Ah + z$, where $\mathbb{E}[h] = w$.
- $z \sim \mathcal{N}(0, \sigma^2 I)$.

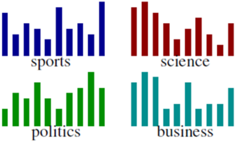
$$M_2 = \sum_i w_i a_i \otimes a_i, \quad M_3 = \sum_i w_i a_i \otimes a_i \otimes a_i.$$

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

What about learning other latent variable models?

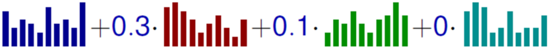
Outline

Topic Modeling



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

E.g.,



$\sim_{\text{iid}}\ 0.6\cdot$ sports $+0.3\cdot$ science $+0.1\cdot$ politics $+0\cdot$ business

aardvark	0
athlete	3
⋮	⋮
zygote	1

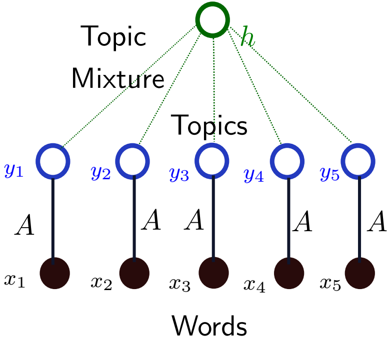
$\Pr_\theta[\text{“play”} \mid \text{sports}] = 0.0002$
$\Pr_\theta[\text{“game”} \mid \text{sports}] = 0.0003$
$\Pr_\theta[\text{“season”} \mid \text{sports}] = 0.0001$
$\vdots$

Probabilistic Topic Models

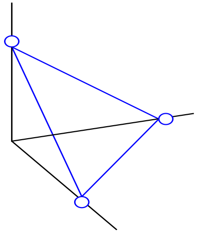
- Useful abstraction for automatic categorization of documents
- Observed: words. Hidden: topics.
- Bag of words: order of words does not matter

Graphical model representation

- l words in a document $x_1, \ldots, x_l$.
- h: proportions of topics in a document.
- Word x_i generated from topic y_i.
- Exchangeability: $\boxed{x_1 \perp\!\!\!\perp x_2 \perp\!\!\!\perp \ldots | h}$
- $\boxed{A(i,j) := \mathbb{P}[x_m = i | y_m = j]}$: topic-word matrix.



Distribution of the topic proportions vector h

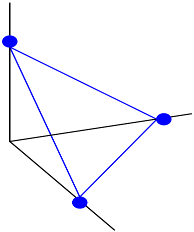


If there are k topics, distribution over the simplex Δ^{k-1}

$$\boxed{\Delta^{k-1} := \{h \in \mathbb{R}^k, h_i \in [0,1], \sum_i h_i = 1\}.}$$

- Simplification for today: there is only one topic in each document.
  - $h \in \{e_1, \dots, e_k\}$: basis vectors.
- Tomorrow: Latent Dirichlet allocation.

Distribution of the topic proportions vector h



If there are k topics, distribution over the simplex Δ^{k-1}

$$\boxed{\Delta^{k-1} := \{h \in \mathbb{R}^k, h_i \in [0,1], \sum_i h_i = 1\}.}$$

- Simplification for today: there is only one topic in each document.
  - $h \in \{e_1, \ldots, e_k\}$: basis vectors.
- Tomorrow: Latent Dirichlet allocation.

Formulation as Linear Models

Distribution of the words $x_1, x_2, \ldots$

- Order d words in vocabulary. If x_1 is j^{th} word, assign $e_j \in \mathbb{R}^d$.
- Distribution of each x_i: supported on vertices of Δ^{d-1}.

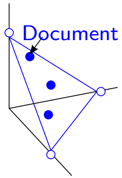
Properties

$$\Pr[x_1|\text{topic h } = i] = \mathbb{E}[x_1|\text{topic h} = i] = a_i$$

- Linear Model: $\boxed{\mathbb{E}[x_i|h] = Ah}$.
- Multiview model: h is fixed and multiple words (x_i) are generated.

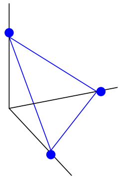
Geometric Picture for Topic Models

Topic proportions vector (h)



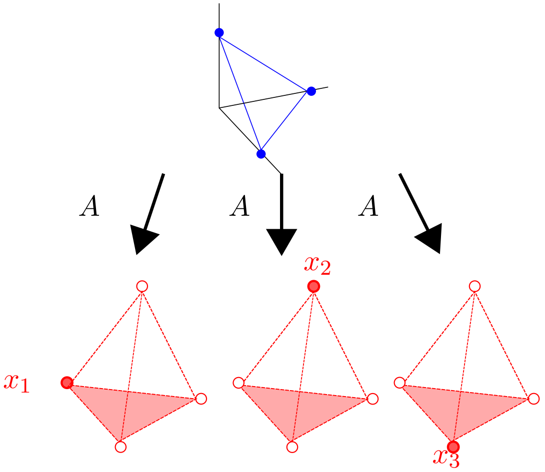
Geometric Picture for Topic Models

Single topic (h)



Geometric Picture for Topic Models

Single topic (h)



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

Gaussian Mixtures vs. Topic Models

(spherical) Mixture of Gaussian:

- k means: $a_1, \dots a_k$
- Component $h = i$ with prob. w_i
- observe x, with spherical noise,

$$x = a_i + z, \quad z \sim \mathcal{N}(0, \sigma_i^2 I)$$

(single) Topic Models

- k topics: $a_1, \dots a_k$
- Topic $h = i$ with prob. w_i
- observe l (exchangeable) words

$$x_1, x_2, \dots x_l \text{ i.i.d. from } a_i$$

- Unified Linear Model: $\mathbb{E}[x|h] = Ah$
- Gaussian mixture: single view, spherical noise.
- Topic model: multi-view, heteroskedastic noise.
- Three words per document suffice for learning.

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

Outline

Recap: Basic Tensor Decomposition Method

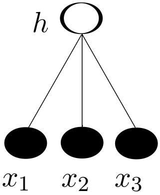
Toy Example in MATLAB

- Simulated Samples: Exchangeable Model
- Whiten The Samples
  - Second Order Moments
  - Matrix Decomposition
- Orthogonal Tensor Eigen Decomposition
  - Third Order Moments
  - Power Iteration

Simulated Samples: Exchangeable Model

Model Parameters

- Hidden State:
  $h \in$ basis $\{e_1, \ldots, e_k\}$
  $k = 2$
- Observed States:
  $x_i \in$ basis $\{e_1, \ldots, e_d\}$
  $d = 3$
- Conditional Independency:
  $x_1 \perp\!\!\!\perp x_2 \perp\!\!\!\perp x_3 | h$
  Transition Matrix: A
- Exchangeability:
  $\mathbb{E}[x_i|h] = Ah$, $\forall i \in 1, 2, 3$



Simulated Samples: Exchangeable Model

Model Parameters

- Hidden State:
  $h \in$ basis $\{e_1, \ldots, e_k\}$
  $k = 2$
- Observed States:
  $x_i \in$ basis $\{e_1, \ldots, e_d\}$
  $d = 3$
- Conditional Independency:
  $x_1 \perp\!\!\!\perp x_2 \perp\!\!\!\perp x_3 | h$
  Transition Matrix: A
- Exchangeability:
  $\mathbb{E}[x_i | h] = Ah, \ \forall i \in 1, 2, 3$

Generate Samples Snippet

```
for t = 1 : n
   % generate h for this sample
   h_category=(rand()>0.5) + 1;
   h(t,h_category)=1;
   transition_cum=cumsum(A_true(:,h_category));
   % generate x1 for this sample | h
   x_category=find(transition_cum> rand(),1);
   x1(t,x_category)=1;
   % generate x2 for this sample | h
   x_category=find(transition_cum >rand(),1);
   x2(t,x_category)=1;
   % generate x3 for this sample | h
   x_category=find(transition_cum > rand(),1);
   x3(t,x_category)=1;
   end
```

Whiten The Samples

Second Order Moments

- $M_2 = \frac{1}{n} \sum_t x_1^t \otimes x_2^t$

Whitening Matrix

- $W = U_w L_w^{-0.5}$,
  $[U_w, L_w] =$k-svd(M_2)

Whiten Data

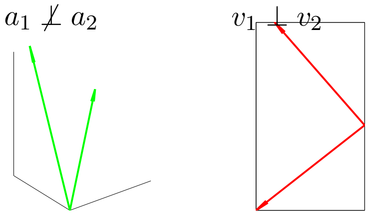
- $y_1^t = W^\top x_1^t$

Orthogonal Basis

- $V = W^\top A \to V^\top V = I$

Whitening Snippet

```
fprintf('The second order moment M2:');
M2 = x1'*x2/n
[Uw, Lw, Vw]= svd(M2);
fprintf('M2 singular values:'); Lw
W = Uw(:,1:k)* sqrt(pinv(Lw(1:k,1:k)));
y1 = x1 * W; y2 = x2 * W; y3 = x3 * W;
```



Orthogonal Tensor Eigen Decomposition

Third Order Moments

$$T = \frac{1}{n} \sum_{t \in [n]} y_1^t \otimes y_2^t \otimes y_3^t \approx \sum_{i \in [k]} \lambda_i v_i \otimes v_i \otimes v_i, \quad V^\top V = I$$

Gradient Ascent

$$T(I, v_1, v_1) = \frac{1}{n} \sum_{t \in [n]} \langle v_1, y_2^t \rangle \langle v_1, y_3^t \rangle y_1^t \approx \sum_i \lambda_i \langle v_i, v_1 \rangle^2 v_i = \lambda_1 v_1.$$

- v_i are eigenvectors of tensor T.

Orthogonal Tensor Eigen Decomposition

$$T \leftarrow T - \sum_j \lambda_j v_j^{\otimes^3}, \quad v \leftarrow \frac{T(I,v,v)}{\|T(I,v,v)\|}$$

Power Iteration Snippet

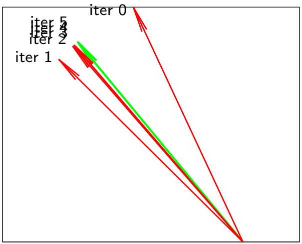
```
V = zeros(k,k); Lambda = zeros(k,1);
for i = 1:k
   v_old = rand(k,1); v_old = normc(v_old);
   for iter = 1 : Maxiter
      v_new = (y1'* ((y2*v_old).*(y3*v_old)))/n;
      if i >1
      % deflation
         for j = 1: i-1
            v_new=v_new-(V(:,j)*(v_old'*V(:,j))2)* Lambda(j);
         end
      end
      lambda = norm(v_new);v_new = normc(v_new);
      if norm(v_old - v_new) < TOL
         fprintf('Converged at iteration %d.', iter);
         V(:,i) = v_new; Lambda(i,1) = lambda;
         break;
      end
      v_old = v_new;
   end
end
```

Orthogonal Tensor Eigen Decomposition

$$T \leftarrow T - \sum_j \lambda_j v_j^{\otimes^3}, \quad v \leftarrow \frac{T(I,v,v)}{\|T(I,v,v)\|}$$

Power Iteration Snippet

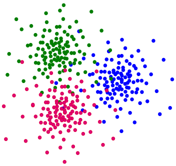
```
V = zeros(k,k); Lambda = zeros(k,1);
for i = 1:k
    v_old = rand(k,1); v_old = normc(v_old);
    for iter = 1 : Maxiter
        v_new = (y1'* ((y2*v_old).*(y3*v_old)))/n;
        if i >1
        % deflation
            for j = 1: i-1
                v_new=v_new-(V(:,j)*(v_old'*V(:,j))2)* Lambda(j);
            end
        end
        lambda = norm(v_new);v_new = normc(v_new);
        if norm(v_old - v_new) < TOL
            fprintf('Converged at iteration %d.', iter);
            V(:,i) = v_new; Lambda(i,1) = lambda;
            break;
        end
        v_old = v_new;
    end
end
```

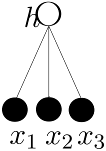


Green: Groundtruth

Red: Estimation at each iteration

Conclusion





- Gaussian mixtures and topic models. Unified linear representation.
- Learning through higher order moments.
- Tensor decomposition via whitening and power method.

Tomorrow's lecture

Latent variable models and moments. Analysis of tensor power method.

Wednesday's lecture: Implementation of tensor method.

Code snippet available at

`http://newport.eecs.uci.edu/anandkumar/MLSS.html`