Guaranteed Learning of Latent Variable Models through Spectral and Tensor Methods

Anima Anandkumar

U.C. Irvine

Guaranteed Unsupervised Learning

- Unsupervised Learning: no labeled samples available for training.

Guaranteed Unsupervised Learning

- Unsupervised Learning: no labeled samples available for training.

Challenge: Conditions for Identifiability

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

Guaranteed Unsupervised Learning

- Unsupervised Learning: no labeled samples available for training.

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?

Guaranteed Unsupervised Learning

- Unsupervised Learning: no labeled samples available for training.

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

Probabilistic Models

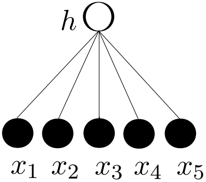
Latent Variable Models

- Concise statistical description through graphical modeling
- Conditional independence relationships or hierarchy of variables.

Probabilistic Models

Latent Variable Models

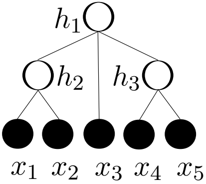
- Concise statistical description through graphical modeling
- Conditional independence relationships or hierarchy of variables.



Probabilistic Models

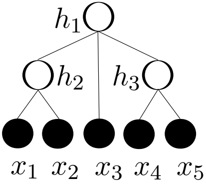
Latent Variable Models

- Concise statistical description through graphical modeling
- Conditional independence relationships or hierarchy of variables.



Probabilistic Models

Latent Variable Models



- Concise statistical description through graphical modeling
- Conditional independence relationships or hierarchy of variables.

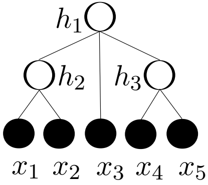
Maximum Likelihood vs. Moment method

- Finding MLE is NP-hard in general.
- Expectation maximization (EM) converges to a local optimum.

Probabilistic Models

Latent Variable Models

- Concise statistical description through graphical modeling
- Conditional independence relationships or hierarchy of variables.



Maximum Likelihood vs. Moment method

- Finding MLE is NP-hard in general.
- Expectation maximization (EM) converges to a local optimum.

- Moment estimate: polynomial computational & sample complexity.
- Le Cam theory: Newton-Ralphson on moment estimate leads to efficient estimator asymptotically.
- Scalable implementation: linear and multilinear algebraic operations.

Game Plan: In this talk

Recall Yesterday's Talk

- Gaussian mixtures and (single) topic models.
- Analysis of third order moments.
- Tensor decomposition method: whitening and power method.

Game Plan: In this talk

Recall Yesterday's Talk

- Gaussian mixtures and (single) topic models.
- Analysis of third order moments.
- Tensor decomposition method: whitening and power method.

Today's talk

- Moments for various latent variable models.
- Analysis of tensor power method.

Game Plan: In this talk

Recall Yesterday's Talk

- Gaussian mixtures and (single) topic models.
- Analysis of third order moments.
- Tensor decomposition method: whitening and power method.

Today's talk

- Moments for various latent variable models.
- Analysis of tensor power method.

Tomorrow's talk

- Implementation of tensor method.

Outline

Recap: Gaussian Mixtures and (single) 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

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

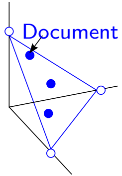
- Unified Linear Model: $\mathbb{E}[x|h] = Ah$
- Gaussian mixture: single view, spherical noise.
- Topic model: multi-view, heteroskedastic noise.

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

Recap: Geometric Picture for Topic Models

- Topic models are exchangeble multiview models.
- $M_2 = \mathbb{E}[x_1 \otimes x_2]$. $M_3 = \mathbb{E}[x_1 \otimes x_2 \otimes x_3]$.

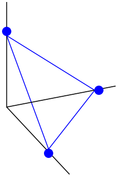
Topic proportions vector (h)



Recap: Geometric Picture for Topic Models

- Topic models are exchangeble multiview models.
- $M_2 = \mathbb{E}[x_1 \otimes x_2]$. $M_3 = \mathbb{E}[x_1 \otimes x_2 \otimes x_3]$.

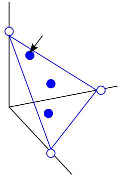
Single topic (h)



Recap: Geometric Picture for Topic Models

- Topic models are exchangeble multiview models.
- $M_2 = \mathbb{E}[x_1 \otimes x_2]$. $M_3 = \mathbb{E}[x_1 \otimes x_2 \otimes x_3]$.

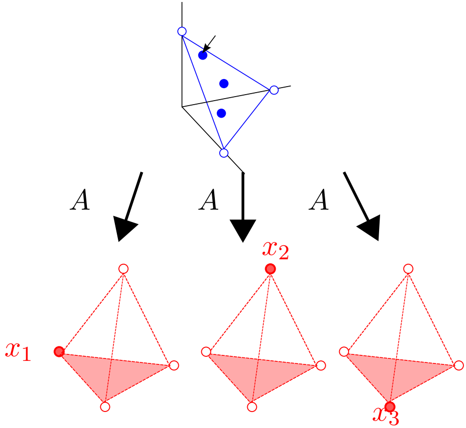
Topic proportions vector (h)



Recap: Geometric Picture for Topic Models

- Topic models are exchangeble multiview models.
- $M_2 = \mathbb{E}[x_1 \otimes x_2]$. $M_3 = \mathbb{E}[x_1 \otimes x_2 \otimes x_3]$.

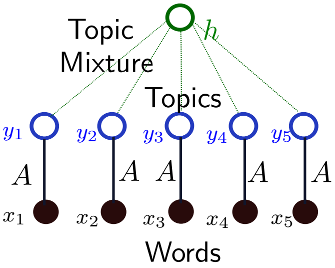
Topic proportions vector (h)



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

Latent Dirichlet Allocation

- l words in a document $x_1, \ldots, x_l$.
- 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.



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

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

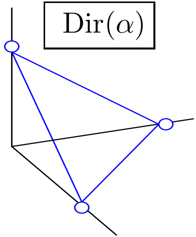
Latent Dirichlet Allocation: h is drawn from a Dirichlet distribution.

Dirichlet Distribution

$$\boxed{\mathbb{P}[h] \propto \prod_{j=1}^{k} h(j)^{\alpha_j - 1},\ \sum_{j=1}^{k} h(j) = 1}$$

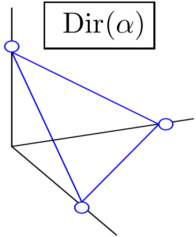
Dirichlet Distribution

$$\boxed{\mathbb{P}[h] \propto \prod_{j=1}^{k} h(j)^{\alpha_j - 1},\ \sum_{j=1}^{k} h(j) = 1}$$



Dirichlet Distribution

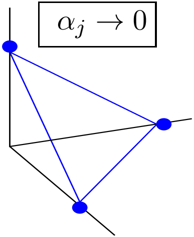
$$\boxed{\mathbb{P}[h] \propto \prod_{j=1}^{k} h(j)^{\alpha_j - 1}, \ \sum_{j=1}^{k} h(j) = 1}$$



- Dirichlet concentration parameter $\boxed{\alpha_0 := \sum_j \alpha_j}$
- Sparsity level in h is $O(\alpha_0)$.

Dirichlet Distribution

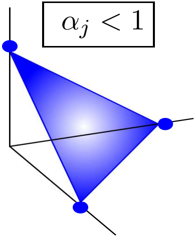
$$\boxed{\mathbb{P}[h] \propto \prod_{j=1}^{k} h(j)^{\alpha_j - 1}, \ \sum_{j=1}^{k} h(j) = 1}$$



- Dirichlet concentration parameter $\boxed{\alpha_0 := \sum_j \alpha_j}$
- Sparsity level in h is $O(\alpha_0)$.

Dirichlet Distribution

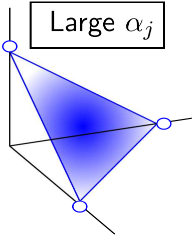
$$\boxed{\mathbb{P}[h] \propto \prod_{j=1}^{k} h(j)^{\alpha_j - 1}, \ \sum_{j=1}^{k} h(j) = 1}$$



- Dirichlet concentration parameter $\boxed{\alpha_0 := \sum_j \alpha_j}$
- Sparsity level in h is $O(\alpha_0)$.

Dirichlet Distribution

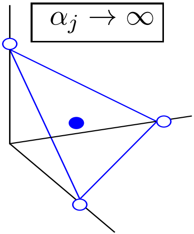
$$\boxed{\mathbb{P}[h] \propto \prod_{j=1}^{k} h(j)^{\alpha_j - 1}, \ \sum_{j=1}^{k} h(j) = 1}$$



- Dirichlet concentration parameter $\boxed{\alpha_0 := \sum_j \alpha_j}$
- Sparsity level in h is $O(\alpha_0)$.

Dirichlet Distribution

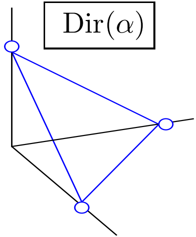
$$\boxed{\mathbb{P}[h] \propto \prod_{j=1}^{k} h(j)^{\alpha_j - 1}, \ \sum_{j=1}^{k} h(j) = 1}$$



- Dirichlet concentration parameter $\boxed{\alpha_0 := \sum_j \alpha_j}$
- Sparsity level in h is $O(\alpha_0)$.

Dirichlet Distribution

$$\boxed{\mathbb{P}[h] \propto \prod_{j=1}^{k} h(j)^{\alpha_j - 1}, \ \sum_{j=1}^{k} h(j) = 1}$$



- Dirichlet concentration parameter $\boxed{\alpha_0 := \sum_j \alpha_j}$
- Sparsity level in h is $O(\alpha_0)$.

Moments under LDA

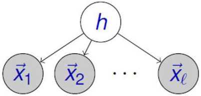
$$
\begin{aligned}
M_2 &:= \mathbb{E}[x_1 \otimes x_2] && - \frac{\alpha_0}{\alpha_0 + 1}\mathbb{E}[x_1] \otimes \mathbb{E}[x_1] \\
M_3 &:= \mathbb{E}[x_1 \otimes x_2 \otimes x_3] && - \frac{\alpha_0}{\alpha_0 + 2}\mathbb{E}[x_1 \otimes x_2 \otimes \mathbb{E}[x_1]] - \text{more stuff...}
\end{aligned}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

Then

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

- Three words per document suffice for learning LDA.

General Multiview Mixtures (Naive Bayes)



$h \in [k]$,

$\vec{x}_1 \in \mathbb{R}^{d_1}, \vec{x}_2 \in \mathbb{R}^{d_2}, \ldots, \vec{x}_\ell \in \mathbb{R}^{d_\ell}$.

$k = \#$ components, $\ell = \#$ views (*e.g.*, audio, video, text).

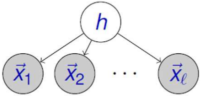
View 1: $\vec{x}_1 \in \mathbb{R}^{d_1}$

View 2: $\vec{x}_2 \in \mathbb{R}^{d_2}$

View 3: $\vec{x}_3 \in \mathbb{R}^{d_3}$

- $\mathbb{E}[x_i|h] = A_i h$ and multiple views.

General Multiview Mixtures (Naive Bayes)



$h \in [k]$,

$\vec{x}_1 \in \mathbb{R}^{d_1}, \vec{x}_2 \in \mathbb{R}^{d_2}, \dots, \vec{x}_\ell \in \mathbb{R}^{d_\ell}$.

k = # components, ℓ = # views (*e.g.*, audio, video, text).

View 1: $\vec{x}_1 \in \mathbb{R}^{d_1}$

View 2: $\vec{x}_2 \in \mathbb{R}^{d_2}$

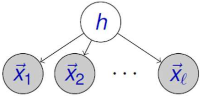
View 3: $\vec{x}_3 \in \mathbb{R}^{d_3}$

- $\mathbb{E}[x_i|h] = A_i h$ and multiple views.

$$\tilde{x}_1 := \mathbb{E}[x_3 \otimes x_2]\mathbb{E}[x_1 \otimes x_2]^\dagger x_1, \quad \tilde{x}_2 := \mathbb{E}[x_3 \otimes x_1]\mathbb{E}[x_2 \otimes x_1]^\dagger x_2,$$

$$M_2 = \mathbb{E}[\tilde{x}_1 \otimes \tilde{x}_1], \qquad M_3 = \mathbb{E}[\tilde{x}_1 \otimes \tilde{x}_2 \otimes x_3].$$

General Multiview Mixtures (Naive Bayes)



$h \in [k]$,

$\vec{x}_1 \in \mathbb{R}^{d_1}, \vec{x}_2 \in \mathbb{R}^{d_2}, \ldots, \vec{x}_\ell \in \mathbb{R}^{d_\ell}$.

k = # components, ℓ = # views (*e.g.*, audio, video, text).

View 1: $\vec{x}_1 \in \mathbb{R}^{d_1}$

View 2: $\vec{x}_2 \in \mathbb{R}^{d_2}$

View 3: $\vec{x}_3 \in \mathbb{R}^{d_3}$

- $\mathbb{E}[x_i|h] = A_i h$ and multiple views.

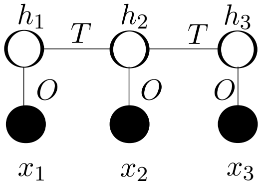
$$\tilde{x}_1 := \mathbb{E}[x_3 \otimes x_2]\mathbb{E}[x_1 \otimes x_2]^\dagger x_1, \quad \tilde{x}_2 := \mathbb{E}[x_3 \otimes x_1]\mathbb{E}[x_2 \otimes x_1]^\dagger x_2,$$

$$M_2 = \mathbb{E}[\tilde{x}_1 \otimes \tilde{x}_1], \quad M_3 = \mathbb{E}[\tilde{x}_1 \otimes \tilde{x}_2 \otimes x_3].$$

$$M_2 = \sum_i w_i a_{3,i} \otimes a_{3,i}, \quad M_3 = \sum_i w_i a_{3,i} \otimes a_{3,i} \otimes a_{3,i}.$$

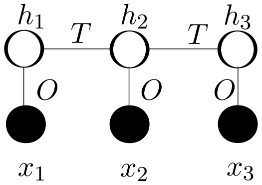
Hidden Markov Models

- $\mathbb{P}[h_{t+1} = i | h_t = j] = T_{i,j}$.
- $\mathbb{E}[x_t | h_t = j] = O e_j$.
- π: Initial distribution (of x_1).



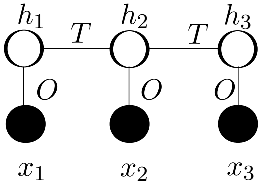
Hidden Markov Models

- $\mathbb{P}[h_{t+1} = i | h_t = j] = T_{i,j}$.
- $\mathbb{E}[x_t | h_t = j] = O e_j$.
- π: Initial distribution (of x_1).
- Three view model. $w := T\pi$.



Hidden Markov Models

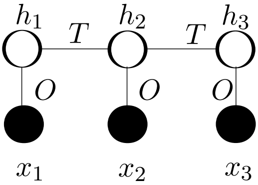
- $\mathbb{P}[h_{t+1} = i | h_t = j] = T_{i,j}$.
- $\mathbb{E}[x_t | h_t = j] = O e_j$.
- π: Initial distribution (of x_1).
- Three view model. $w := T\pi$.



$$
\begin{aligned}
\mathbb{E}[x_1 | h_2] &= O\mathrm{Diag}(\pi) T^\top \mathrm{Diag}(w)^{-1} h_2 \\
\mathbb{E}[x_2 | h_2] &= O h_2 \\
\mathbb{E}[x_3 | h_2] &= O T h_2.
\end{aligned}
$$

Hidden Markov Models

- $\mathbb{P}[h_{t+1} = i | h_t = j] = T_{i,j}$.
- $\mathbb{E}[x_t | h_t = j] = O e_j$.
- π: Initial distribution (of x_1).
- Three view model. $w := T\pi$.



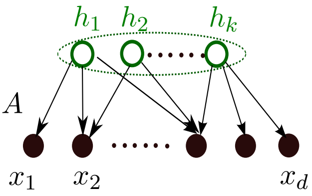
$$\mathbb{E}[x_1 | h_2] = O\text{Diag}(\pi) T^\top \text{Diag}(w)^{-1} h_2$$
$$\mathbb{E}[x_2 | h_2] = O h_2$$
$$\mathbb{E}[x_3 | h_2] = OT h_2.$$

Condition for non-degeneracy

- $O \in \mathbb{R}^{d \times k}$ has full column rank.
- T is invertible, π and $T\pi$ have positive entries.

Independent Component Analysis

- Independent sources, unknown mixing.
- Blind source separation.
- Application: speech, image, video..
- k sources. d dimensions.



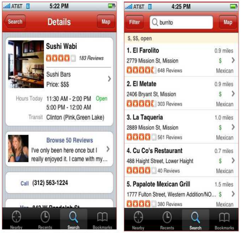
- $x = Ah + z$. $z \sim \mathcal{N}(0, \sigma^2 I)$. Sources h_i are independent.
- Form cumulant tensor

$$M_4 := \mathbb{E}[x^{\otimes 4}] - \mathbb{E}[x_{i_1} x_{i_2}]\mathbb{E}[x_{i_3} x_{i_4}] \ldots$$
$$= \sum_i \kappa_i a_i \otimes a_i \otimes a_i \otimes a_i.$$

- Kurtosis: $\kappa_i := \mathbb{E}[h_i^4] - 3$.
- Assumption: sources have non-zero kurtosis $(\kappa_i \neq 0)$.

Outline

Social Networks & Recommender Systems



Social Networks

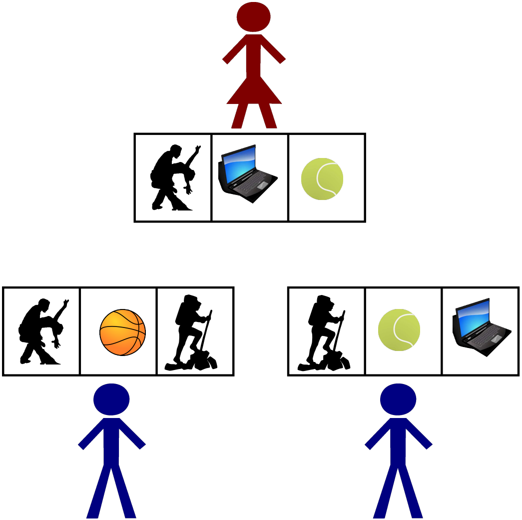
- Network of social ties, e.g. friendships, co-authorships
- Hidden: communities of actors.

Recommender Systems

- Observed: Ratings of users for various products.
- Goal: New recommendations.
- Modeling: User/product groups.

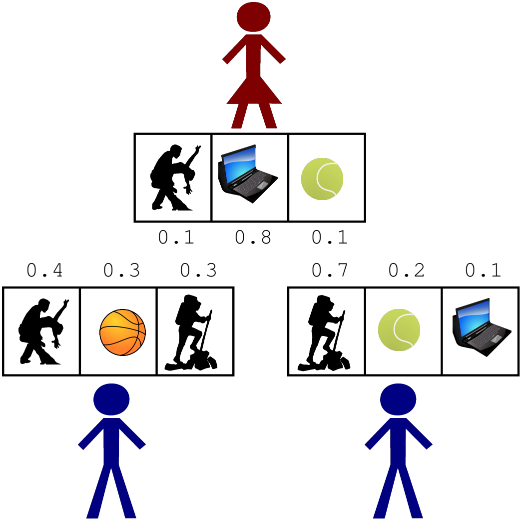
Network Community Models

- How are communities formed? How do communities interact?



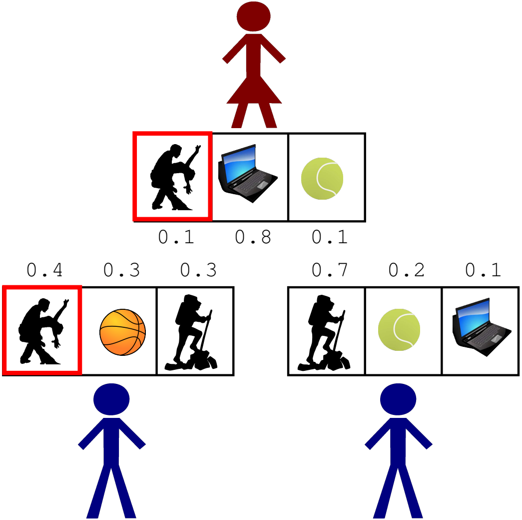
Network Community Models

- How are communities formed? How do communities interact?



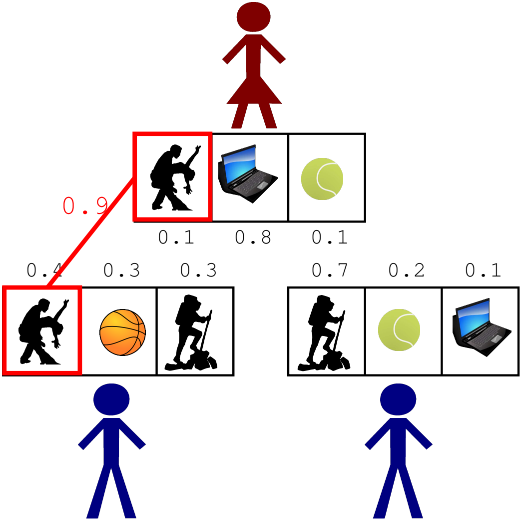
Network Community Models

- How are communities formed? How do communities interact?



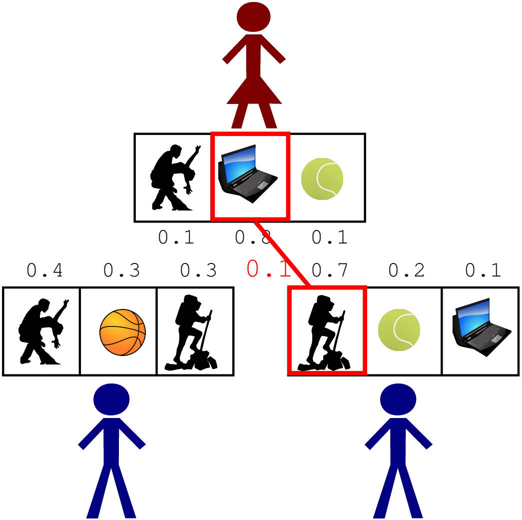
Network Community Models

- How are communities formed? How do communities interact?



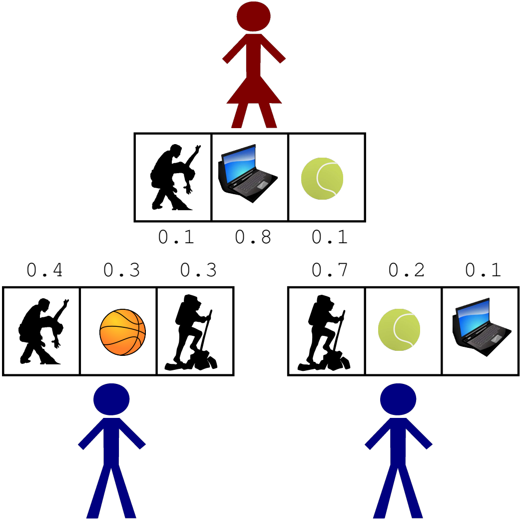
Network Community Models

- How are communities formed? How do communities interact?



Network Community Models

- How are communities formed? How do communities interact?



Mixed Membership Model (Airoldi et al)

- k communities and n nodes. Graph $G \in \mathbb{R}^{n \times n}$ (adjacency matrix).
- Fractional memberships: $\pi_x \in \mathbb{R}^k$ membership of node x.

$$\boxed{\Delta^{k-1} := \{\pi_x \in \mathbb{R}^k, \pi_x(i) \in [0,1], \sum_i \pi_x(i) = 1, \quad \forall\, x \in [n]\}.}$$

- Node memberships $\{\pi_u\}$ drawn from Dirichlet distribution.

Mixed Membership Model (Airoldi et al)

- k communities and n nodes. Graph $G \in \mathbb{R}^{n\times n}$ (adjacency matrix).
- Fractional memberships: $\pi_x \in \mathbb{R}^k$ membership of node x.

$$\boxed{\Delta^{k-1} := \{\pi_x \in \mathbb{R}^k, \pi_x(i) \in [0,1], \sum_i \pi_x(i) = 1, \quad \forall\, x \in [n]\}.}$$

- Node memberships $\{\pi_u\}$ drawn from Dirichlet distribution.
- Edges conditionally independent given community memberships: $G_{i,j} \perp\!\!\!\perp G_{a,b} | \pi_i, \pi_j, \pi_a, \pi_b$.
- Edge probability averaged over community memberships

$$\boxed{\mathbb{P}[G_{i,j} = 1 | \pi_i, \pi_j] = \mathbb{E}[G_{i,j} | \pi_i, \pi_j] = \pi_i^\top P \pi_j.}$$

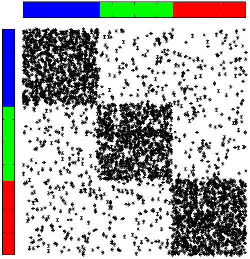
- $P \in \mathbb{R}^{k\times k}$: average edge connectivity for pure communities.

Airoldi, Blei, Fienberg, and Xing. Mixed membership stochastic blockmodels. J. of Machine Learning Research, June 2008.

Networks under Community Models

Networks under Community Models

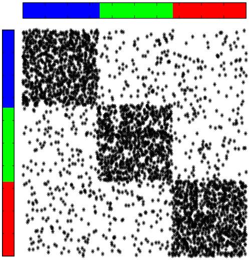
Stochastic Block Model



$\alpha_0 = 0$

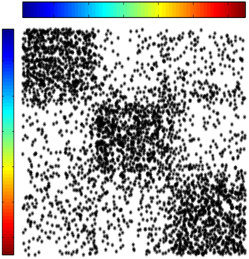
Networks under Community Models

Stochastic Block Model



$\alpha_0 = 0$

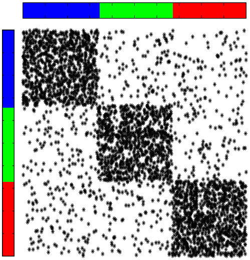
Mixed Membership Model



$\alpha_0 = 1$

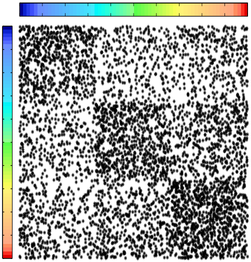
Networks under Community Models

Stochastic Block Model



$\alpha_0 = 0$

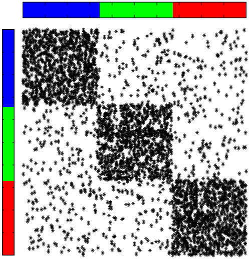
Mixed Membership Model



$\alpha_0 = 10$

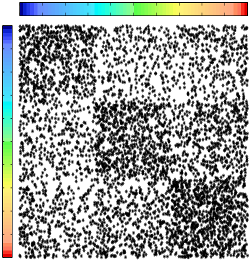
Networks under Community Models

Stochastic Block Model



$\alpha_0 = 0$

Mixed Membership Model

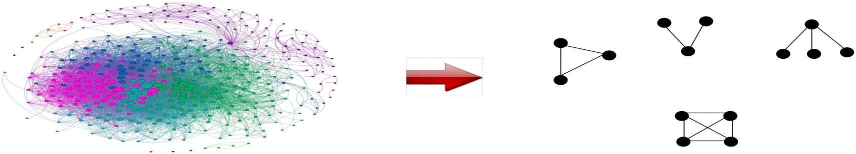


$\alpha_0 = 10$

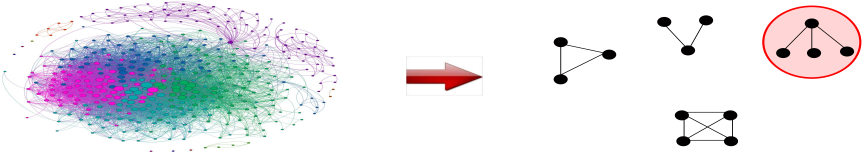
Unifying Assumption

- Edges conditionally independent given community memberships

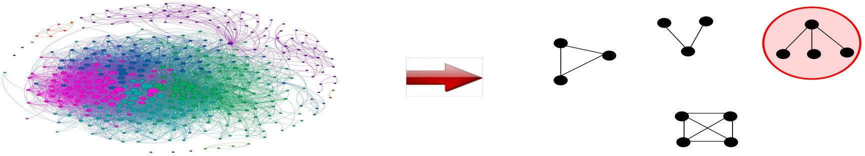
Subgraph Counts as Graph Moments



Subgraph Counts as Graph Moments

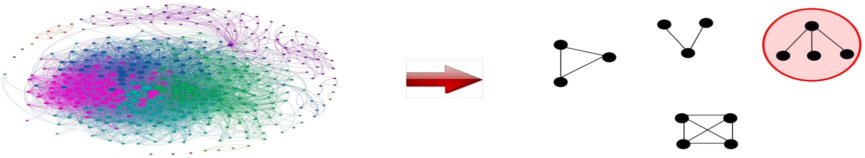


Subgraph Counts as Graph Moments



3-star counts sufficient for identifiability and learning of MMSB

Subgraph Counts as Graph Moments



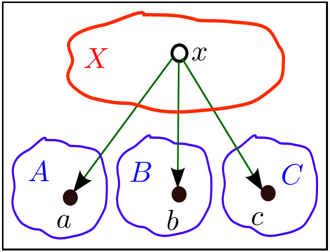
3-star counts sufficient for identifiability and learning of MMSB

3-Star Count Tensor

$$M_3(a,b,c) = \frac{1}{|X|}\# \text{ of common neighbors in } X$$

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

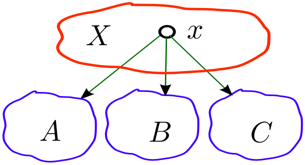
$$M_3 = \frac{1}{|X|}\sum_{x \in X} [G_{x,A}^{\top} \otimes G_{x,B}^{\top} \otimes G_{x,C}^{\top}]$$



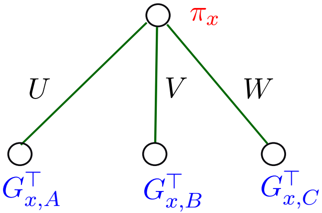
Multi-view Representation

- Conditional independence of the three views
- π_x: community membership vector of node x.

3-stars



Graphical model



- Linear Multiview Model: $\boxed{\mathbb{E}[G^\top_{x,A}|\Pi] = \Pi^\top_A P^\top \pi_x = U\pi_x.}$

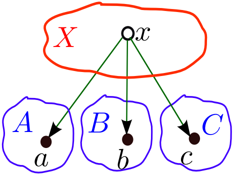
Subgraph Counts as Graph Moments

Second and Third Order Moments

- $\boxed{M_2 := \frac{1}{|X|} \sum_x Z_C G_{x,C}^\top G_{x,B} Z_B^\top - \text{shift}}$
- $\boxed{M_3 := \frac{1}{|X|} \sum_x \left[G_{x,A}^\top \otimes Z_B G_{x,B}^\top \otimes Z_C G_{x,C}^\top \right] - \text{shift}}$

Symmetrize Transition Matrices

- $\mathrm{Pairs}_{C,B} := G_{X,C}^\top \otimes G_{X,B}^\top$
- $Z_B := \mathrm{Pairs}\,(A,C)\,(\mathrm{Pairs}\,(B,C))^\dagger$
- $Z_C := \mathrm{Pairs}\,(A,B)\,(\mathrm{Pairs}\,(C,B))^\dagger$



- Linear Multiview Model: $\mathbb{E}[G_{x,A}^\top | \Pi] = U\pi_x$.

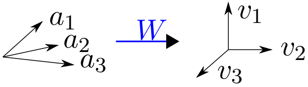
$$\boxed{\mathbb{E}[M_2|\Pi_{A,B,C}] = \sum_i \frac{\alpha_i}{\alpha_0} u_i \otimes u_i, \quad \mathbb{E}[M_3|\Pi_{A,B,C}] = \sum_i \frac{\alpha_i}{\alpha_0} u_i \otimes u_i \otimes u_i.}$$

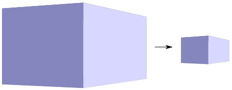
Outline

Recap of Tensor Method

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

- Whitening matrix W from SVD of M_2.





- Multilinear transform: $T = M_3(W, W, W)$.

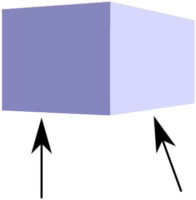
Tensor M_3 Tensor T

- Eigenvectors of T through power method and deflation.

$$\boxed{v \mapsto \frac{T(I, v, v)}{\|T(I, v, v)\|}}.$$

Orthogonal Tensor Eigen Decomposition

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

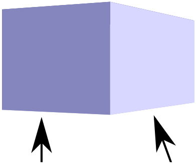


- $T(I, v_1, v_1) = \sum_i \lambda_i \langle v_i, v_1 \rangle^2 v_i = \lambda_1 v_1.$
- v_i are eigenvectors of tensor T.

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

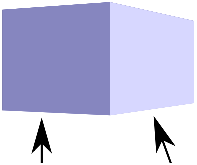
- Start from an initial vector v.

$$\boxed{v \mapsto \frac{T(I, v, v)}{\|T(I, v, v)\|}.}$$

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.

$$\boxed{v \mapsto \frac{T(I, v, v)}{\|T(I, v, v)\|}.}$$

Questions

- Is there convergence? Does the convergence depend on initialization?
- What about performance under noise?

Recap of Matrix Eigen Analysis

- For symmetric $M \in \mathbb{R}^{k \times k}$, eigen decomposition: $M = \sum_i \lambda_i v_i v_i^\top$.
- Eigen vectors are fixed points: $Mv = \lambda v$.
  - In our notation: $M(I, v) = \lambda v$.

Uniqueness (Identifiability): Iff. λ_i are distinct.

Recap of Matrix Eigen Analysis

- For symmetric $M \in \mathbb{R}^{k \times k}$, eigen decomposition: $M = \sum_i \lambda_i v_i v_i^\top$.
- Eigen vectors are fixed points: $Mv = \lambda v$.
  - In our notation: $M(I, v) = \lambda v$.

Uniqueness (Identifiability): Iff. λ_i are distinct.

Power method: $\boxed{v \mapsto \frac{M(I, v)}{\|M(I, v)\|}.}$

Recap of Matrix Eigen Analysis

- For symmetric $M \in \mathbb{R}^{k\times k}$, eigen decomposition: $M = \sum_i \lambda_i v_i v_i^\top$.
- Eigen vectors are fixed points: $Mv = \lambda v$.
  - In our notation: $M(I, v) = \lambda v$.

Uniqueness (Identifiability): Iff. λ_i are distinct.

Power method: $\boxed{v \mapsto \frac{M(I,v)}{\|M(I,v)\|}.}$

Convergence properties

- Let $\lambda_1 > \lambda_2 \ldots > \lambda_d$. $\{v_i\}$ form a basis.
- Let initialization $v = \sum_i c_i v_i$.
- If $c_1 \neq 0$, power method converges to v_1.

Recap of Matrix Eigen Analysis

- For symmetric $M \in \mathbb{R}^{k\times k}$, eigen decomposition: $M = \sum_i \lambda_i v_i v_i^\top$.
- Eigen vectors are fixed points: $Mv = \lambda v$.
  - In our notation: $M(I, v) = \lambda v$.

Uniqueness (Identifiability): Iff. λ_i are distinct.

Power method: $\boxed{v \mapsto \frac{M(I,v)}{\|M(I,v)\|}.}$

Convergence properties

- Let $\lambda_1 > \lambda_2 \ldots > \lambda_d$. $\{v_i\}$ form a basis.
- Let initialization $v = \sum_i c_i v_i$.
- If $c_1 \neq 0$, power method converges to v_1.

Perturbation analysis (Davis-Kahan): $T + E$

Require $\|E\| < \min_{i\neq j} |\lambda_i - \lambda_j|$.

Optimization viewpoint of matrix analysis

- $M = \sum_{i \in [k]} \lambda_i v_i \otimes v_i, \quad \lambda_1 > \lambda_2 \ldots.$
- Rayleigh quotient at v: $\boxed{M(v,v) = v^\top M v = \sum_i \lambda_i \langle v_i, v \rangle^2.}$
- Optimization problem: $\boxed{\max_v M(v,v) \text{ s.t. } \|v\| = 1.}$

Optimization viewpoint of matrix analysis

- $M = \sum_{i \in [k]} \lambda_i v_i \otimes v_i, \quad \lambda_1 > \lambda_2 \ldots.$
- Rayleigh quotient at v: $\boxed{M(v,v) = v^\top M v = \sum_i \lambda_i \langle v_i, v\rangle^2.}$
- Optimization problem: $\boxed{\max_v M(v,v) \text{ s.t. } \|v\| = 1.}$
- Non-convex problem. Global maximizer is v_1 (top eigenvector).

Optimization viewpoint of matrix analysis

- $M = \sum_{i \in [k]} \lambda_i v_i \otimes v_i, \quad \lambda_1 > \lambda_2 \ldots.$
- Rayleigh quotient at v: $\boxed{M(v,v) = v^\top M v = \sum_i \lambda_i \langle v_i, v \rangle^2.}$
- Optimization problem: $\boxed{\max_v M(v,v) \text{ s.t. } \|v\| = 1.}$
- Non-convex problem. Global maximizer is v_1 (top eigenvector).

What are the local optimizers?

Optimization viewpoint of matrix analysis

Optimization: $\boxed{\max_v M(v,v) \text{ s.t. } \|v\| = 1.}$

$$\boxed{\text{Lagrangian: } L(v,\lambda) := M(v,v) - \lambda(v^\top v - 1).}$$

Optimization viewpoint of matrix analysis

Optimization: $\boxed{\max_v M(v,v) \text{ s.t. } \|v\| = 1.}$

$$\boxed{\text{Lagrangian: } L(v,\lambda) := M(v,v) - \lambda(v^\top v - 1).}$$

- First derivative: $\nabla L(v,\lambda) = 2(M(I,v) - \lambda v)$.

Optimization viewpoint of matrix analysis

Optimization: $\boxed{\max_v M(v,v) \text{ s.t. } \|v\| = 1.}$

$$\boxed{\text{Lagrangian: } L(v,\lambda) := M(v,v) - \lambda(v^\top v - 1).}$$

- First derivative: $\nabla L(v,\lambda) = 2(M(I,v) - \lambda v)$.
- Stationary points are eigenvectors: $\nabla L(v,\lambda) = 0$.

Optimization viewpoint of matrix analysis

Optimization: $\boxed{\max_v M(v,v) \text{ s.t. } \|v\| = 1.}$

$$\boxed{\text{Lagrangian: } L(v,\lambda) := M(v,v) - \lambda(v^\top v - 1).}$$

- First derivative: $\nabla L(v,\lambda) = 2(M(I,v) - \lambda v)$.
- Stationary points are eigenvectors: $\nabla L(v,\lambda) = 0$.
- Power method $v \mapsto \frac{M(I,v)}{\|M(I,v)\|}$ is a version of gradient ascent.

Optimization viewpoint of matrix analysis

Optimization: $\boxed{\max_v M(v,v) \text{ s.t. } \|v\| = 1.}$

$\boxed{\text{Lagrangian: } L(v,\lambda) := M(v,v) - \lambda(v^\top v - 1).}$

- First derivative: $\nabla L(v,\lambda) = 2(M(I,v) - \lambda v)$.
- Stationary points are eigenvectors: $\nabla L(v,\lambda) = 0$.
- Power method $v \mapsto \frac{M(I,v)}{\|M(I,v)\|}$ is a version of gradient ascent.
- Second derivative: $\nabla^2 L(v,\lambda) = 2(M - \lambda I)$.

Optimization viewpoint of matrix analysis

Optimization: $\boxed{\max_v M(v,v) \text{ s.t. } \|v\| = 1.}$

$$\boxed{\text{Lagrangian: } L(v,\lambda) := M(v,v) - \lambda(v^\top v - 1).}$$

- First derivative: $\nabla L(v,\lambda) = 2(M(I,v) - \lambda v)$.
- Stationary points are eigenvectors: $\nabla L(v,\lambda) = 0$.
- Power method $v \mapsto \frac{M(I,v)}{\|M(I,v)\|}$ is a version of gradient ascent.
- Second derivative: $\nabla^2 L(v,\lambda) = 2(M - \lambda I)$.

Local optimality condition for constrained optimization

$\boxed{w^\top \nabla^2 L(v,\lambda) w < 0}$ for all $w \perp v$, at a stationary point v.

Optimization viewpoint of matrix analysis

Optimization: $\boxed{\max_v M(v,v) \text{ s.t. } \|v\| = 1.}$

$\boxed{\text{Lagrangian: } L(v,\lambda) := M(v,v) - \lambda(v^\top v - 1).}$

- First derivative: $\nabla L(v,\lambda) = 2(M(I,v) - \lambda v)$.
- Stationary points are eigenvectors: $\nabla L(v,\lambda) = 0$.
- Power method $v \mapsto \frac{M(I,v)}{\|M(I,v)\|}$ is a version of gradient ascent.
- Second derivative: $\nabla^2 L(v,\lambda) = 2(M - \lambda I)$.

Local optimality condition for constrained optimization

$\boxed{w^\top \nabla^2 L(v,\lambda) w < 0}$ for all $w \perp v$, at a stationary point v.

- Verify: v_1 is the only local optimum.
- Verify: All other eigenvectors are saddle points.

Optimization viewpoint of matrix analysis

Optimization: $\boxed{\max_v M(v,v) \text{ s.t. } \|v\| = 1.}$

$$\boxed{\text{Lagrangian: } L(v,\lambda) := M(v,v) - \lambda(v^\top v - 1).}$$

- First derivative: $\nabla L(v,\lambda) = 2(M(I,v) - \lambda v)$.
- Stationary points are eigenvectors: $\nabla L(v,\lambda) = 0$.
- Power method $v \mapsto \frac{M(I,v)}{\|M(I,v)\|}$ is a version of gradient ascent.
- Second derivative: $\nabla^2 L(v,\lambda) = 2(M - \lambda I)$.

Local optimality condition for constrained optimization

$\boxed{w^\top \nabla^2 L(v,\lambda) w < 0}$ for all $w \perp v$, at a stationary point v.

- Verify: v_1 is the only local optimum.
- Verify: All other eigenvectors are saddle points.

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

Analysis of Tensor Power Method

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

Bad news about tensors

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

Analysis of Tensor Power Method

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

Bad news about tensors

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

We will see that a tractable case is when we are promised that an orthogonal decomposition exists.

Analysis of Tensor Power Method

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

Bad news about tensors

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

We will see that a tractable case is when we are promised that an orthogonal decomposition exists.

Characterization of components $\{v_i\}$

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

Analysis of Tensor Power Method

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

Bad news about tensors

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

We will see that a tractable case is when we are promised that an orthogonal decomposition exists.

Characterization of components $\{v_i\}$

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

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

Analysis of Tensor Power Method

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

Bad news about tensors

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

We will see that a tractable case is when we are promised that an orthogonal decomposition exists.

Characterization of components $\{v_i\}$

- $\{v_i\}$ 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)?

Optimization viewpoint of tensor analysis

Optimization: $\boxed{\max_v T(v,v,v) \text{ s.t. } \|v\| = 1.}$

$\boxed{\text{Lagrangian: } L(v,\lambda) := T(v,v,v) - \lambda(v^\top v - 1).}$

Optimization viewpoint of tensor analysis

Optimization: $\boxed{\max_v T(v, v, v) \text{ s.t. } \|v\| = 1.}$

$$\boxed{\text{Lagrangian: } L(v, \lambda) := T(v, v, v) - \lambda(v^\top v - 1).}$$

- First derivative: $\nabla L(v, \lambda) = 3(T(I, v, v) - \lambda v)$.

Optimization viewpoint of tensor analysis

Optimization: $\boxed{\max_v T(v,v,v) \text{ s.t. } \|v\| = 1.}$

$\boxed{\text{Lagrangian: } L(v,\lambda) := T(v,v,v) - \lambda(v^\top v - 1).}$

- First derivative: $\nabla L(v,\lambda) = 3(T(I,v,v) - \lambda v)$.
- Stationary points are eigenvectors: $\nabla L(v,\lambda) = 0$.

Optimization viewpoint of tensor analysis

Optimization: $\boxed{\max_v T(v,v,v) \text{ s.t. } \|v\| = 1.}$

$$\boxed{\text{Lagrangian: } L(v,\lambda) := T(v,v,v) - \lambda(v^\top v - 1).}$$

- First derivative: $\nabla L(v,\lambda) = 3(T(I,v,v) - \lambda v)$.
- Stationary points are eigenvectors: $\nabla L(v,\lambda) = 0$.
- Power method $v \mapsto \frac{T(I,v,v)}{\|T(I,v,v)\|}$ is a version of gradient ascent.

Optimization viewpoint of tensor analysis

Optimization: $\boxed{\max_v T(v,v,v) \text{ s.t. } \|v\| = 1.}$

$$\boxed{\text{Lagrangian: } L(v,\lambda) := T(v,v,v) - \lambda(v^\top v - 1).}$$

- First derivative: $\nabla L(v,\lambda) = 3(T(I,v,v) - \lambda v)$.
- Stationary points are eigenvectors: $\nabla L(v,\lambda) = 0$.
- Power method $v \mapsto \frac{T(I,v,v)}{\|T(I,v,v)\|}$ is a version of gradient ascent.
- Second derivative: $\nabla^2 L(v,\lambda) = 3(2T(I,I,v) - \lambda I)$.

Optimization viewpoint of tensor analysis

Optimization: $\boxed{\max_v T(v,v,v) \text{ s.t. } \|v\| = 1.}$

$$\boxed{\text{Lagrangian: } L(v,\lambda) := T(v,v,v) - \lambda(v^\top v - 1).}$$

- First derivative: $\nabla L(v,\lambda) = 3(T(I,v,v) - \lambda v)$.
- Stationary points are eigenvectors: $\nabla L(v,\lambda) = 0$.
- Power method $v \mapsto \frac{T(I,v,v)}{\|T(I,v,v)\|}$ is a version of gradient ascent.
- Second derivative: $\nabla^2 L(v,\lambda) = 3(2T(I,I,v) - \lambda I)$.

Local optimality condition for constrained optimization

$\boxed{w^\top \nabla^2 L(v,\lambda) w < 0}$ for all $w \perp v$, at a stationary point v.

Optimization viewpoint of tensor analysis

Optimization: $\boxed{\max_v T(v,v,v) \text{ s.t. } \|v\| = 1.}$

$\boxed{\text{Lagrangian: } L(v,\lambda) := T(v,v,v) - \lambda(v^\top v - 1).}$

- First derivative: $\nabla L(v,\lambda) = 3(T(I,v,v) - \lambda v)$.
- Stationary points are eigenvectors: $\nabla L(v,\lambda) = 0$.
- Power method $v \mapsto \frac{T(I,v,v)}{\|T(I,v,v)\|}$ is a version of gradient ascent.
- Second derivative: $\nabla^2 L(v,\lambda) = 3(2T(I,I,v) - \lambda I)$.

Local optimality condition for constrained optimization

$\boxed{w^\top \nabla^2 L(v,\lambda) w < 0}$ for all $w \perp v$, at a stationary point v.

- Verify: $\{v_i\}$ are the only local optima.
- Verify: All other eigenvectors are saddle points.

Optimization viewpoint of tensor analysis

Optimization: $\boxed{\max_v T(v,v,v) \text{ s.t. } \|v\| = 1.}$

$$\boxed{\text{Lagrangian: } L(v,\lambda) := T(v,v,v) - \lambda(v^\top v - 1).}$$

- First derivative: $\nabla L(v,\lambda) = 3(T(I,v,v) - \lambda v)$.
- Stationary points are eigenvectors: $\nabla L(v,\lambda) = 0$.
- Power method $v \mapsto \frac{T(I,v,v)}{\|T(I,v,v)\|}$ is a version of gradient ascent.
- Second derivative: $\nabla^2 L(v,\lambda) = 3(2T(I,I,v) - \lambda I)$.

Local optimality condition for constrained optimization

$\boxed{w^\top \nabla^2 L(v,\lambda) w < 0}$ for all $w \perp v$, at a stationary point v.

- Verify: $\{v_i\}$ are the only local optima.
- Verify: All other eigenvectors are saddle points.

For an orthogonal tensor, no spurious local optima!

Review: matrix power iteration

Recall matrix power iteration for matrix $M := \sum_i \lambda_i \; v_i v_i^\top$:

$$\boxed{\text{Start with some } v\text{, and for } j = 1, 2, \dots : \quad v \mapsto \; Mv \; = \; \sum_i \lambda_i \; \left(v_i^\top v\right) \; v_i.}$$

i.e., component in v_i direction is scaled by λ_i.

Review: matrix power iteration

Recall matrix power iteration for matrix $M := \sum_i \lambda_i \; v_i v_i^\top$:

Start with some v, and for $j = 1, 2, \ldots$:

$$v \mapsto \; Mv \; = \; \sum_i \lambda_i \; \left(v_i^\top v\right) \; v_i.$$

i.e., component in v_i direction is scaled by λ_i.

If $\lambda_1 > \lambda_2 \geq \cdots$, then in t iterations,

$$\frac{\left(v_1^\top v\right)^2}{\sum_i \left(v_i^\top v\right)^2} \geq 1 - k\left(\frac{\lambda_2}{\lambda_1}\right)^{2t}.$$

Converges *linearly* to v_1 assuming gap $\lambda_2/\lambda_1 < 1$.

Tensor power iteration convergence analysis

Let $c_i := v_i^\top v$ initial component in v_i direction; assume WLOG

$$\lambda_1|c_1| > \lambda_2|c_2| \geq \lambda_3|c_3| \geq \cdots .$$

Tensor power iteration convergence analysis

Let $c_i := v_i^\top v$ initial component in v_i direction; assume WLOG

$$\lambda_1 |c_1| > \lambda_2 |c_2| \geq \lambda_3 |c_3| \geq \cdots .$$

Then

$$v \;\mapsto\; \sum_i \lambda_i \, \left(v_i^\top v\right)^2 v_i \;=\; \sum_i \lambda_i c_i^2 \; v_i$$

i.e., component in v_i direction is squared then scaled by λ_i.

Tensor power iteration convergence analysis

Let $c_i := v_i^\top v$ initial component in v_i direction; assume WLOG

$$\lambda_1|c_1| > \lambda_2|c_2| \geq \lambda_3|c_3| \geq \cdots .$$

Then

$$v \;\mapsto\; \sum_i \lambda_i \, \left(v_i^\top v\right)^2 v_i \;=\; \sum_i \lambda_i c_i^2 \; v_i$$

i.e., component in v_i direction is squared then scaled by λ_i.

By induction, in t iterations

$$v \;=\; \sum_i \lambda_i^{2^t-1} c_i^{2^t} \; v_i,$$

so

$$\frac{\left(v_1^\top v\right)^2}{\sum_i \left(v_i^\top v\right)^2} \geq 1 - k\left(\frac{\lambda_1}{\max_{i\neq 1}\lambda_i}\right)^2 \left|\frac{v_2 c_2}{v_1 c_1}\right|^{2^{t+1}} .$$

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 $\lambda_i |c_i|$.
   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 $\lambda_i|c_i|$.
   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 $\lambda_i|c_i|$.
   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.

Perturbation Analysis

$$\hat{T} = T + E, \quad T = \sum_i \lambda_i v_i \otimes v_i \otimes v_i, \quad \|E\| := \max_{x:\|x\|=1} |E(x,x,x)| \leq \epsilon.$$

Theorem: Let N be number of iterations. If

$$\boxed{N \geq \log k + \log\log \frac{\lambda_{\max}}{\epsilon}, \quad \epsilon < \frac{\lambda_{\min}}{k},}$$

then output (v, λ) (after polynomial restarts) satisfies

$$\|v - v_1\| \leq O\left(\frac{\epsilon}{\lambda_1}\right), \quad \|\lambda - \lambda_1\| \leq O(\epsilon),$$

where v_1 is s.t. $\lambda_1 |c_1| > \lambda_2 |c_2| \ldots, \quad c_i := \langle v_i, v \rangle$, and v is the (successful) initializer.

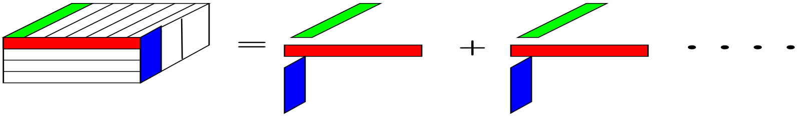
- Careful analysis of deflation: avoid buildup of errors.
- Implies polynomial sample complexity for learning.

Outline

Beyond Orthogonal Tensor Decomposition

- $a \otimes a \otimes a$ is a rank-1 tensor whose $\mathbf{i}^{\text{th}}$ entry is $a(i_1) \cdot a(i_2) \cdot a(i_3)$.
- For tensor T, find decomposition into rank one terms

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



Tensor T $\quad w_1 \cdot a_1 \otimes a_1 \otimes a_1 \quad w_2 \cdot a_2 \otimes a_2 \otimes a_2$

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

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

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

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

$$\boxed{\max_i \|\widehat{a}_i - a_i\| = \tilde{O}\left(\sqrt{\frac{k}{n}}\right) + \tilde{O}\left(\frac{\sqrt{k}}{d}\right)}$$

- 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

Conditions for recovery

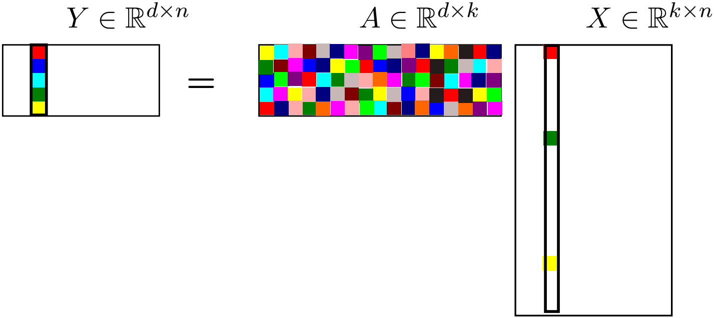
- No. of mixture components: $k = C \cdot d$
- No. of unlabeled samples: $n = \tilde{\Omega}(k \cdot d)$.
- Computational complexity: $\tilde{O}\left(e^{C^2}\right)$

Our result: achieved error with n unlabeled samples

$$\boxed{\max_i \|\widehat{a}_i - a_i\| = \tilde{O}\left(\sqrt{\frac{k}{n}}\right) + \tilde{O}\left(\frac{\sqrt{k}}{d}\right)}$$

- 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 $k = \Theta(d)$.

Learning Overcomplete Dictionaries

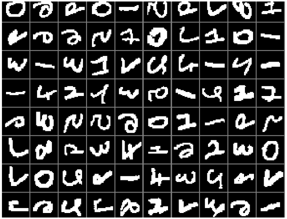


- Linear model: $Y = AX$, both A, X unknown.
- Sparse X: each column is randomly s-sparse
- Overcomplete dictionary $A \in \mathbb{R}^{d\times k}$: $k \geq d$.
- Incoherence: $\max_{i\neq j} |\langle a_i, a_j\rangle| \approx 0$. (satisfied by random vectors)

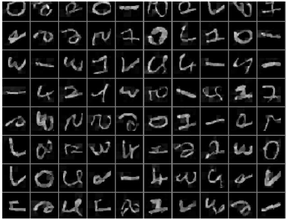
"Learning Sparsely Used Overcomplete Dictionaries" by A. Agarwal, A., P. Jain, P. Netrapalli, R. Tandon. COLT 2014.

Experiments on MNIST

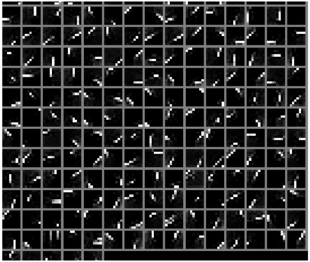
Original



Reconstruction



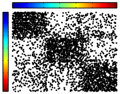
Learnt Representation



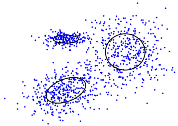
Outline

Conclusion

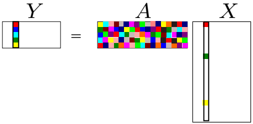
Guaranteed Learning of Latent Variable Models



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



- Tensor approach: mixed membership communities, topic models, latent trees...
- Sparsity-based approach: overcomplete models, e.g sparse coding and topic models.



Tomorrow's lecture

- Implementation of tensor approaches.