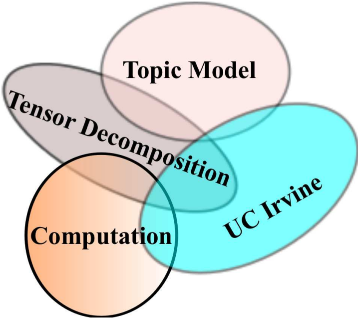
Fast Online Tensor Method for Overlapping Community Detection

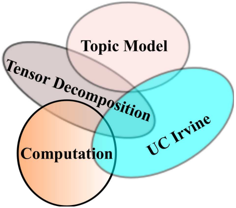
F. Huang, U. N. Niranjan, M. U. Hakeem, A. Anandkumar

Electrical Engineering and Computer Science
U. C. Irvine



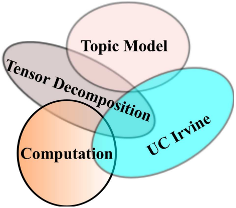
Mixed Membership Model

- Edoardo Airoldi, David Blei, Stephen Fienberg and Eric Xing 2008
- People belong to multiple communities
- k communities and n nodes, $k \ll n$
- Node membership for node u: π_u
  - Mixed membership: $\pi_u \in [0,1]^k$
  - Fractional membership: $\|\pi_u\|_1 = 1$



Mixed Membership Model

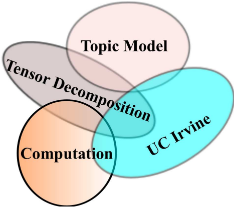
- Edoardo Airoldi, David Blei, Stephen Fienberg and Eric Xing 2008
- People belong to multiple communities
- k communities and n nodes, $k \ll n$
- Node membership for node u: π_u
  - Mixed membership: $\pi_u \in [0,1]^k$
  - Fractional membership: $\|\pi_u\|_1 = 1$



Community Detection: Infer hidden communities from observed network.

Mixed Membership Model

- Edoardo Airoldi, David Blei, Stephen Fienberg and Eric Xing 2008
- People belong to multiple communities
- k communities and n nodes, $k \ll n$
- Node membership for node u: π_u
  - Mixed membership: $\pi_u \in [0,1]^k$
  - Fractional membership: $\|\pi_u\|_1 = 1$

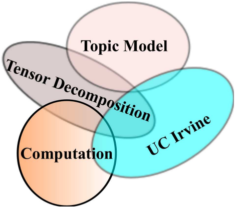


Community Detection: Infer hidden communities from observed network.

Edge Formation Model

- Edges are conditionally independent given community memberships

Mixed Membership Model



- Edoardo Airoldi, David Blei, Stephen Fienberg and Eric Xing 2008
- People belong to multiple communities
- k communities and n nodes, $k \ll n$
- Node membership for node u: π_u
  - Mixed membership: $\pi_u \in [0,1]^k$
  - Fractional membership: $\|\pi_u\|_1 = 1$

Community Detection: Infer hidden communities from observed network.

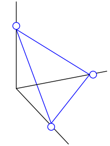
Edge Formation Model

- Edges are conditionally independent given community memberships

Dirichlet Community Membership Model

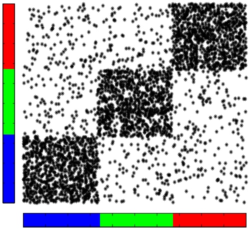
- $\{\pi_u\}$ are independent draws from Dirichlet distribution

$$\mathbb{P}[\pi_u] \propto \prod_{j=1}^{k} \pi_u(j)^{\alpha_j - 1}, \quad \sum_{j=1}^{k} \pi_u(j) = 1, \quad \sum_{j=1}^{k} \alpha_j = \alpha_0$$



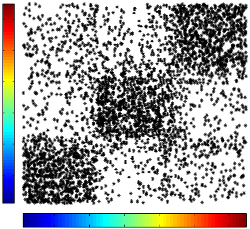
Pure vs. Mixed Membership Community Models

Stochastic Block Model



$\alpha_0 = 0$

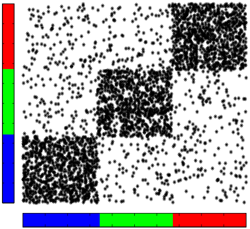
Mixed Membership Model



$\alpha_0 = 1$

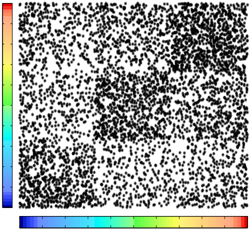
Pure vs. Mixed Membership Community Models

Stochastic Block Model



$\alpha_0 = 0$

Mixed Membership Model

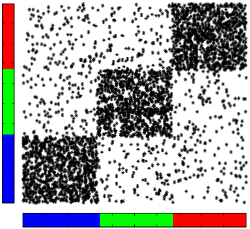


$\alpha_0 = 10$

Goal: Recover communities Π given adjacency matrix G

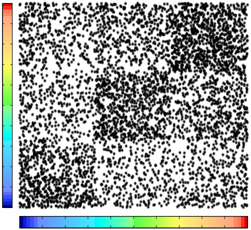
Pure vs. Mixed Membership Community Models

Stochastic Block Model



$\alpha_0 = 0$

Mixed Membership Model



$\alpha_0 = 10$

Goal: Recover communities Π given adjacency matrix G

Challenges in Learning Mixed Membership Models

- Identifiability: when can parameters be estimated?
- Guaranteed learning? What input required?

Approach Overview and Contributions

Approaches(Anandkumar et al, COLT '13)

- Inverse moment method
- Preprocessing to whiten and symmetrize data
- Spectral approach: decompose tensor via batch power method
- Postprocessing: Recover Π from the spectrum by linear operations

Approach Overview and Contributions

Approaches(Anandkumar et al, COLT '13)

- Inverse moment method
- Preprocessing to whiten and symmetrize data
- Spectral approach: decompose tensor via batch power method
- Postprocessing: Recover Π from the spectrum by linear operations

$\boxed{\text{Parallizeable?}}$ $\boxed{\text{Speed?}}$ $\boxed{\text{Scalability?}}$

Approach Overview and Contributions

Approaches(Anandkumar et al, COLT '13)

- Inverse moment method
- Preprocessing to whiten and symmetrize data
- Spectral approach: decompose tensor via batch power method
- Postprocessing: Recover Π from the spectrum by linear operations

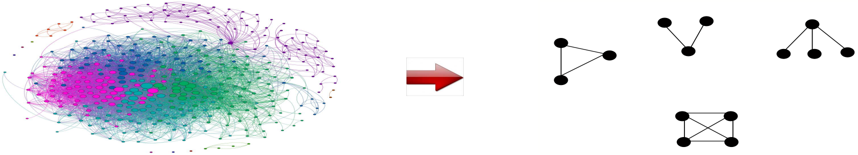
Parallizeable? | Speed? | Scalability?

Contribution Summary

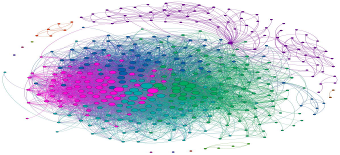
- Randomized Low Rank Approximation for $n \times n$ matrix SVD
- Online tensor decomposition
- GPU Device to minimize data transfer overhead, thus fast updates
- Sparse Implementation scalable to millions of nodes
- Validation Metric: p-value test based "soft-pairing"

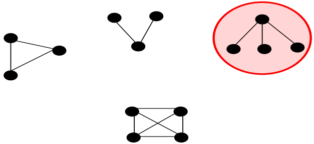
Outline

Subgraph Counts as Graph Moments

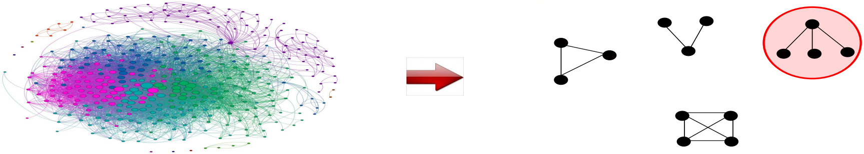


Subgraph Counts as Graph Moments



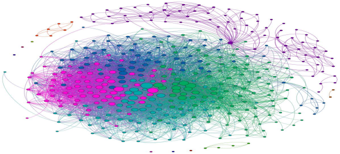


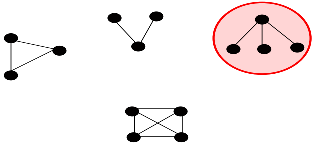
Subgraph Counts as Graph Moments



3-star counts sufficient for identifiability and learning of MMSB

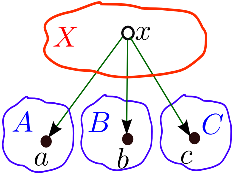
Subgraph Counts as Graph Moments





3-star counts sufficient for identifiability and learning of MMSB

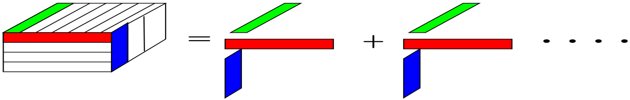
$$M_3(a,b,c) = \frac{1}{|X|} \# \text{ of 3-stars with leaves a,b,c}$$
$$= \frac{1}{|X|} \sum_{x \in X} G(x,a)G(x,b)G(x,c).$$



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

$$\mathbb{E}[M_3|\Pi_{A,B,C}] = \sum_{i \in [k]} \lambda_i [(F_A)_i \otimes (F_B)_i \otimes (F_C)_i]$$

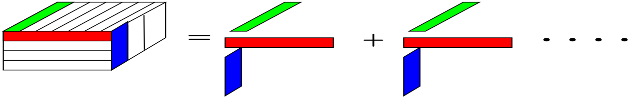
Tensor Dimensionality Reduction



Tensor $\mathbb{E}[M_3|\Pi_{A,B,C}]$ $\quad\lambda_1(F_A)_1 \otimes (F_B)_1 \otimes (F_C)_1 \quad \lambda_2(F_A)_2 \otimes (F_B)_2 \otimes (F_C)_2$

$$\boxed{\mathbb{E}[M_3|\Pi_{A,B,C}] = \sum_{i\in[k]} \lambda_i[(F_A)_i \otimes (F_B)_i \otimes (F_C)_i]}$$

Tensor Dimensionality Reduction

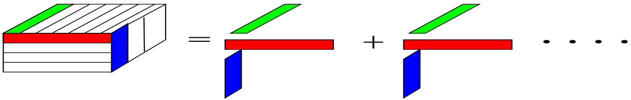


Tensor $\mathbb{E}[M_3|\Pi_{A,B,C}]$ $\quad \lambda_1 (F_A)_1 \otimes (F_B)_1 \otimes (F_C)_1 \quad \lambda_2 (F_A)_2 \otimes (F_B)_2 \otimes (F_C)_2$

$$\boxed{\mathbb{E}[M_3|\Pi_{A,B,C}] = \sum_{i\in[k]} \lambda_i [(F_A)_i \otimes (F_B)_i \otimes (F_C)_i]}$$

Goal: Recover $F_A, F_B, F_C, \vec{\lambda}$ through CP tensor decomposition

Tensor Dimensionality Reduction



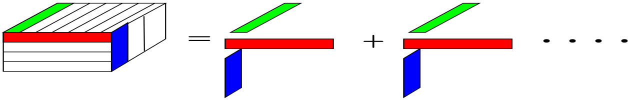
Tensor $\mathbb{E}[M_3|\Pi_{A,B,C}]$ $\lambda_1(F_A)_1 \otimes (F_B)_1 \otimes (F_C)_1$ $\lambda_2(F_A)_2 \otimes (F_B)_2 \otimes (F_C)_2$

$$\boxed{\mathbb{E}[M_3|\Pi_{A,B,C}] = \sum_{i\in[k]} \lambda_i[(F_A)_i \otimes (F_B)_i \otimes (F_C)_i]}$$

Goal: Recover $F_A, F_B, F_C, \vec{\lambda}$ through CP tensor decomposition

Preprocessing/Whitening

Tensor Dimensionality Reduction



Tensor $\mathbb{E}[M_3|\Pi_{A,B,C}]$ $\lambda_1(F_A)_1 \otimes (F_B)_1 \otimes (F_C)_1$ $\lambda_2(F_A)_2 \otimes (F_B)_2 \otimes (F_C)_2$

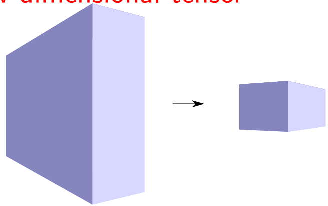
$$\boxed{\mathbb{E}[M_3|\Pi_{A,B,C}] = \sum_{i\in[k]} \lambda_i[(F_A)_i \otimes (F_B)_i \otimes (F_C)_i]}$$

Goal: Recover $F_A, F_B, F_C, \vec{\lambda}$ through CP tensor decomposition

Preprocessing/Whitening

Orthogonalize and Symmetrize low dimensional tensor

- Convert $M_3^{\alpha_0}$ of size $O(n \times n \times n)$ to a tensor T of size $k \times k \times k$
- Find the whitening matrix W
- Perform multilinear transformations on $M_3^{\alpha_0}$ with W to get T



Tensor M_3 Tensor T

Whitening Matrix Computation

Symmetrization: Finding Second Order Moments $M_2^{\alpha_0}$

$$\mathrm{Pairs}_{B,C} := G_{X,B}^\top \otimes G_{X,C}^\top$$

$$M_2^{\alpha_0} = \boxed{\left(\mathrm{Pairs}_{A,B}\,\mathrm{Pairs}_{C,B}^{\dagger}\right)} \; \mathrm{Pairs}_{C,B} \; \boxed{\left(\mathrm{Pairs}_{B,C}^{\dagger}\right)^\top \mathrm{Pairs}_{A,C}^\top} - \text{shift}$$

Whitening Matrix Computation

Symmetrization: Finding Second Order Moments $M_2^{\alpha_0}$

$$\mathrm{Pairs}_{B,C} := G_{X,B}^{\top} \otimes G_{X,C}^{\top}$$

$$M_2^{\alpha_0} = \boxed{\left(\mathrm{Pairs}_{A,B}\, \mathrm{Pairs}_{C,B}^{\dagger}\right)}\ \mathrm{Pairs}_{C,B}\ \boxed{\left(\mathrm{Pairs}_{B,C}^{\dagger}\right)^{\top} \mathrm{Pairs}_{A,C}^{\top}} - \text{shift}$$

Order Manipulation: Low Rank Approx. is the key, avoid $n \times n$ objects

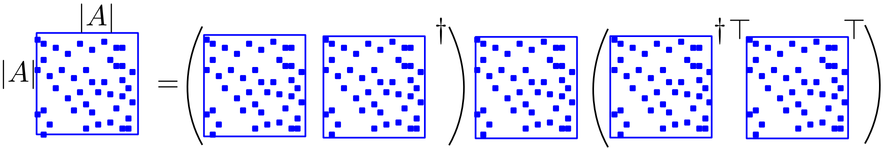
Whitening Matrix Computation

Symmetrization: Finding Second Order Moments $M_2^{\alpha_0}$

$$\mathrm{Pairs}_{B,C} := G_{X,B}^{\top} \otimes G_{X,C}^{\top}$$

$$M_2^{\alpha_0} = \boxed{\left(\mathrm{Pairs}_{A,B}\,\mathrm{Pairs}_{C,B}^{\dagger}\right)}\ \mathrm{Pairs}_{C,B}\ \boxed{\left(\mathrm{Pairs}_{B,C}^{\dagger}\right)^{\top}\ \mathrm{Pairs}_{A,C}^{\top}} - \text{shift}$$

Order Manipulation: Low Rank Approx. is the key, avoid $n \times n$ objects



n=1M, k=5K: Size(Matrix $_{n\times n}$)=58TB vs Size(Matrix $_{n\times k}$)= 3.7GB.

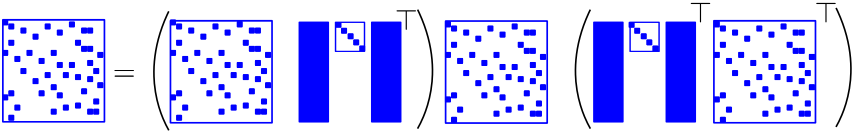
Whitening Matrix Computation

Symmetrization: Finding Second Order Moments $M_2^{\alpha_0}$

$$\mathrm{Pairs}_{B,C} := G_{X,B}^{\top} \otimes G_{X,C}^{\top}$$

$$M_2^{\alpha_0} = \boxed{\left(\mathrm{Pairs}_{A,B}\,\mathrm{Pairs}_{C,B}^{\dagger}\right)}\ \mathrm{Pairs}_{C,B}\ \boxed{\left(\mathrm{Pairs}_{B,C}^{\dagger}\right)^{\top}\ \mathrm{Pairs}_{A,C}^{\top}} - \text{shift}$$

Order Manipulation: Low Rank Approx. is the key, avoid $n \times n$ objects



n=1M, k=5K: Size(Matrix $_{n\times n}$)=58TB vs Size(Matrix $_{n\times k}$)= 3.7GB.

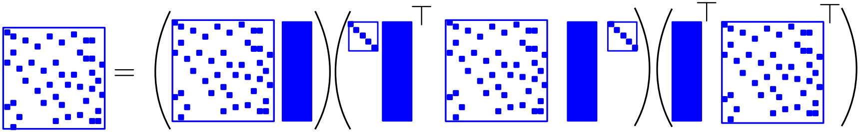
Whitening Matrix Computation

Symmetrization: Finding Second Order Moments $M_2^{\alpha_0}$

$$\mathrm{Pairs}_{B,C} := G_{X,B}^\top \otimes G_{X,C}^\top$$

$$M_2^{\alpha_0} = \boxed{\left(\mathrm{Pairs}_{A,B}\,\mathrm{Pairs}_{C,B}^\dagger\right)}\;\mathrm{Pairs}_{C,B}\;\boxed{\left(\mathrm{Pairs}_{B,C}^\dagger\right)^\top\;\mathrm{Pairs}_{A,C}^\top} - \text{shift}$$

Order Manipulation: Low Rank Approx. is the key, avoid $n \times n$ objects



n=1M, k=5K: Size(Matrix $_{n\times n}$)=58TB vs Size(Matrix $_{n\times k}$)= 3.7GB.

Whitening Matrix Computation

Orthogonalization: Finding Whitening Matrix W

- $W^T M_2^{\alpha_0} W = I$ is solved by $\boxed{\text{k-svd}(M_2^{\alpha_0})}$
- Randomized low rank approx. (Gittens & Mahoney 13', Clarkson & Woodruff 13')

  Both dense and sparse format, tall-thin SVD only

Outline

Tensor Eigen Analysis

Orthogonal symmetric tensor $T = \sum_i \rho_i r_i^{\otimes 3}$

Fact $\boxed{T(I, r_i, r_i) = \sum_j \rho_j \langle r_i, r_j \rangle^2 r_j = \rho_i r_i}$

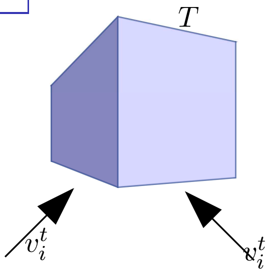
Tensor Eigen Analysis

Orthogonal symmetric tensor $T = \sum_i \rho_i r_i^{\otimes 3}$

Fact $\boxed{T(I, r_i, r_i) = \sum_j \rho_j \langle r_i, r_j \rangle^2 r_j = \rho_i r_i}$

Power Method:

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



Problem: Recover k eigenvectors serially by deflation

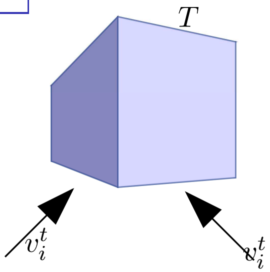
Tensor Eigen Analysis

Orthogonal symmetric tensor $T = \sum_i \rho_i r_i^{\otimes 3}$

Fact $\boxed{T(I, r_i, r_i) = \sum_j \rho_j \langle r_i, r_j \rangle^2 r_j = \rho_i r_i}$

Power Method:

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



Problem: Recover k eigenvectors serially by deflation

Solution: Stochastic method and simultaneous recovery

Stochastic (Implicit) Tensor Gradient Descent

$$\arg\min_{\mathbf{v}} \left\{ \left\| \theta \sum_{i \in [k]} \otimes^3 v_i - \sum_{t \in X} T^t \right\|_F^2 \right\},$$

where v_i are the unknown tensor eigenvectors, $T^t = g_A^t \otimes g_B^t \otimes g_C^t$ such that $g_A^t = W^\top G_{\{x,A\}}, \ldots$

Stochastic (Implicit) Tensor Gradient Descent

$$\arg\min_{\mathbf{v}} \Big\{ \big\|\theta \sum_{i\in[k]} \otimes^3 v_i - \sum_{t\in X} T^t \big\|_F^2 \Big\},$$

where v_i are the unknown tensor eigenvectors, $T^t = g_A^t \otimes g_B^t \otimes g_C^t$ such that $g_A^t = W^\top G_{\{x,A\}}, \ldots$

Expand the objective: $\theta \big\| \sum_{i\in[k]} \otimes^3 v_i \big\|_F^2 - \big\langle \sum_{i\in[k]} \otimes^3 v_i, T^t \big\rangle$
Orthogonality cost vs Correlation Reward

Stochastic (Implicit) Tensor Gradient Descent

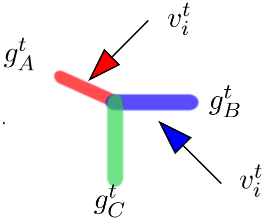
$$\arg\min_{\mathbf{v}} \left\{ \left\| \theta \sum_{i \in [k]} \otimes^3 v_i - \sum_{t \in X} T^t \right\|_F^2 \right\},$$

where v_i are the unknown tensor eigenvectors, $T^t = g_A^t \otimes g_B^t \otimes g_C^t$ such that $g_A^t = W^\top G_{\{x,A\}}, \ldots$

Expand the objective: $\theta \left\| \sum_{i\in[k]} \otimes^3 v_i \right\|_F^2 - \left\langle \sum_{i\in[k]} \otimes^3 v_i, T^t \right\rangle$

Orthogonality cost vs Correlation Reward

$$v_i^{t+1} \leftarrow v_i^t - 3\theta\beta^t \sum_{j=1}^{k} \left[\left\langle v_j^t, v_i^t \right\rangle^2 v_j^t \right] + \beta^t \left\langle v_i^t, g_A^t \right\rangle \left\langle v_i^t, g_B^t \right\rangle g_C^t + \ldots$$



Orthogonality cost vs Correlation Reward

Never form the tensor explicitly; multilinear operation on implicit tensor.

Computational Complexity ($k \ll n$)

- $n = \#$ of nodes
- $k = \#$ of communities
- $N = \#$ of iterations
- $m = \#$ of sampled node pairs (variational)

Module	Pre	STGD	Post	Var
Space	$O(nk)$	$O(k^2)$	$O(nk)$	$O(nk)$
Time	$O(n+k^3)$	$O(Nk)$	$O(n)$	$O(mkN)$

Variational method: $O(m \times k)$ **for each iteration**

$$O(n \times k) < O(m \times k) < O(n^2 \times k)$$

Our approach: $O(n + k^3)$

Computational Complexity ($k \ll n$)

- $n = \#$ of nodes
- $N = \#$ of iterations
- $k = \#$ of communities
- $m = \#$ of sampled node pairs (variational)

Module	Pre	STGD	Post	Var
Space	$O(nk)$	$O(k^2)$	$O(nk)$	$O(nk)$
Time	$O(n + k^3)$	$O(Nk)$	$O(n)$	$O(mkN)$

Variational method: $O(m \times k)$ **for each iteration**

$$O(n \times k) < O(m \times k) < O(n^2 \times k)$$

Our approach: $O(n + k^3)$

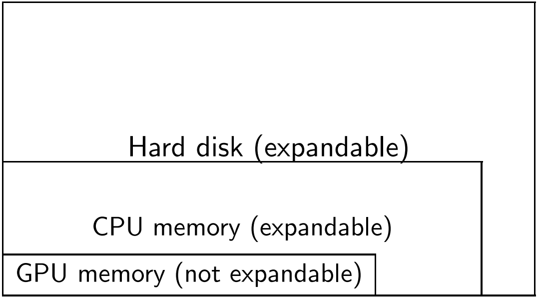
In practice STGD is extremely fast and is not the bottleneck

Outline

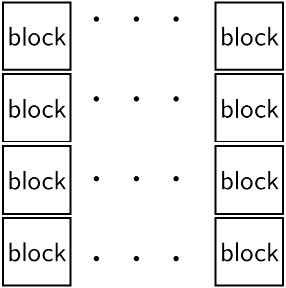
GPU/CPU Implementation

GPU (SIMD)

- GPU: Hundreds of cores; parallelism for matrix/vector operations
- Speed-up: Order of magnitude gains
- Big data challenges: GPU memory $\ll$ CPU memory $\ll$ Hard disk



Storage hierarchy

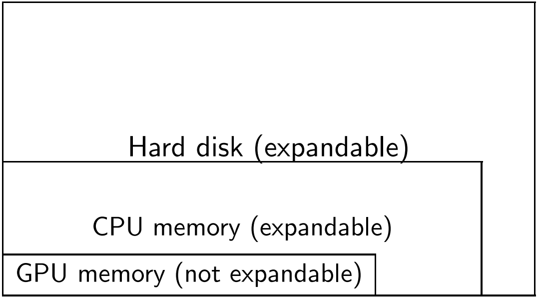


Partitioned matrix

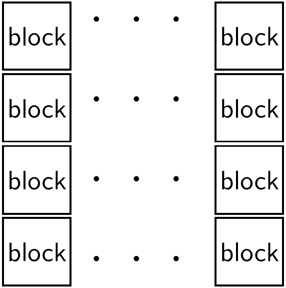
GPU/CPU Implementation

GPU (SIMD)

- GPU: Hundreds of cores; parallelism for matrix/vector operations
- Speed-up: Order of magnitude gains
- Big data challenges: GPU memory $\ll$ CPU memory $\ll$ Hard disk



Storage hierarchy



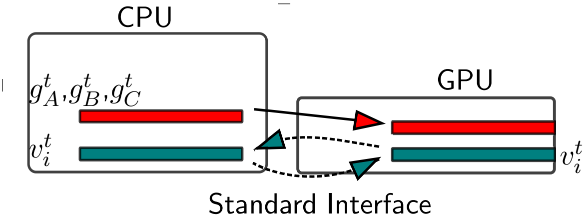
Partitioned matrix

CPU

- CPU: Sparse Representation, Expandable Memory
- Randomized Dimensionality Reduction

Scaling Of The Stochastic Iterations

$$v_i^{t+1} \leftarrow v_i^t - 3\theta\beta^t \sum_{j=1}^{k} \left[\left\langle v_j^t, v_i^t \right\rangle^2 v_j^t \right] + \beta^t \langle v_i^t, g_A^t \rangle \langle v_i^t, g_B^t \rangle g_C^t + \dots$$



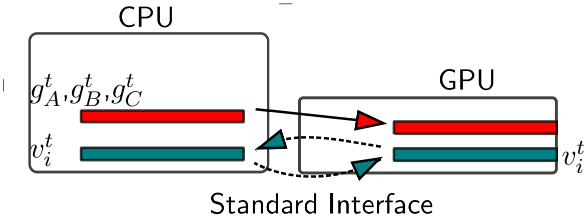
- Parallelize across eigenvectors.

- STGD is iterative: device code reuse buffers for updates.

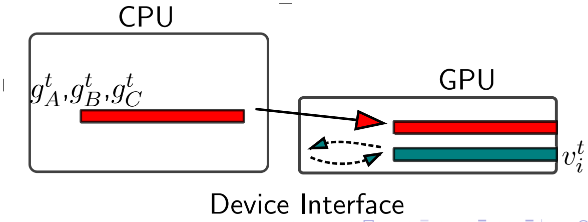
Scaling Of The Stochastic Iterations

$$v_i^{t+1} \leftarrow v_i^t - 3\theta\beta^t \sum_{j=1}^{k} \left[\langle v_j^t, v_i^t \rangle^2 v_j^t \right] + \beta^t \langle v_i^t, g_A^t \rangle \langle v_i^t, g_B^t \rangle g_C^t + \dots$$

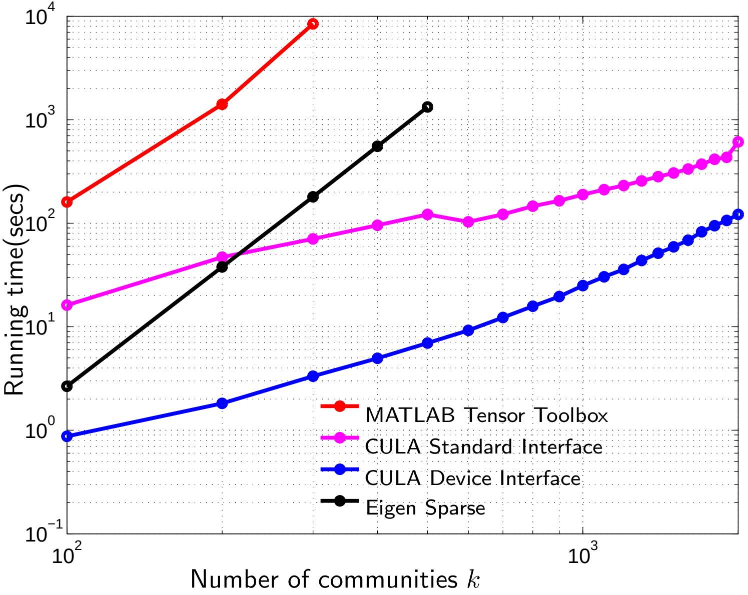
- Parallelize across eigenvectors.



- STGD is iterative: device code reuse buffers for updates.



Scaling Of The Stochastic Iterations



Validation Metrics

Ground-truth membership available

- Ground-truth membership matrix Π
- Estimated membership $\widehat{\Pi}$

Validation Metrics

Ground-truth membership available

- Ground-truth membership matrix Π
- Estimated membership $\widehat{\Pi}$

Problem: How to relate Π and $\widehat{\Pi}$?

Validation Metrics

Ground-truth membership available

- Ground-truth membership matrix Π
- Estimated membership $\widehat{\Pi}$

Problem: How to relate Π and $\widehat{\Pi}$?

Solution: p-value test based soft-"pairing"

Validation Metrics

Ground-truth membership available

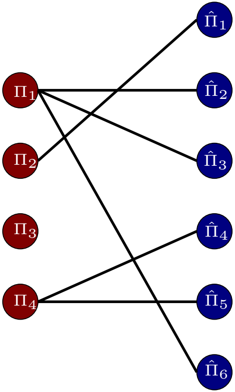
- Ground-truth membership matrix Π
- Estimated membership $\widehat{\Pi}$

Problem: How to relate Π and $\widehat{\Pi}$?

Solution: p-value test based soft-"pairing"

Evaluation Metrics

- Recovery Ratio: % of ground-truth com recovered
- Error Score: $\mathcal{E} := \frac{1}{nk} \sum \{\text{paired membership errors}\}$



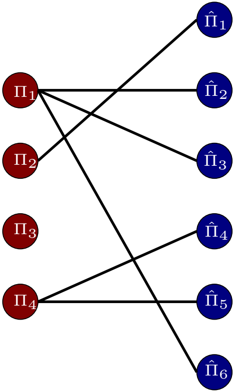
Validation Metrics

Ground-truth membership available

- Ground-truth membership matrix Π
- Estimated membership $\widehat{\Pi}$

Problem: How to relate Π and $\widehat{\Pi}$?

Solution: p-value test based soft-"pairing"



Evaluation Metrics

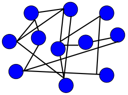
- Recovery Ratio: % of ground-truth com recovered
- Error Score: $\mathcal{E} := \frac{1}{nk} \sum \{\text{paired membership errors}\}$

Comparison with NMI

- Not a true information theoretical measure for overlapping community
- Not a suitable measure for unequal size communities

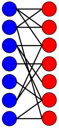
Outline

Summary of Results



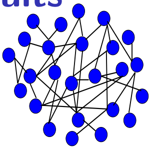
Facebook

$n \sim 20k$



Yelp

$n \sim 40k$



DBLP(sub)

$n \sim 1$ million$(\sim 100k)$

Error ($\mathcal{E}$) and Recovery ratio ($\mathcal{R}$)

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

Thanks to Prem Gopalan and David Mimno for providing variational code.

Summary of Results - Yelp Dataset

Lowest error business categories & largest weight businesses

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

Summary of Results - Yelp Dataset

Lowest error business categories & largest weight businesses

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

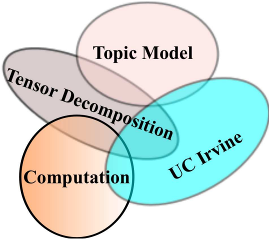
Bridgeness: Distance from vector $[1/\hat{k}, \ldots, 1/\hat{k}]^\top$

Top-5 bridging nodes (businesses)

Business	Categories
Four Peaks Brewing Co	Restaurants, Bars, American, Nightlife, Food, Pubs, Tempe
Pizzeria Bianco	Restaurants, Pizza, Phoenix
FEZ	Restaurants, Bars, American, Nightlife, Mediterranean, Lounges, Phoeni
Matt's Big Breakfast	Restaurants, Phoenix, Breakfast& Brunch
Cornish Pasty Company	Restaurants, Bars, Nightlife, Pubs, Tempe

Outline

Conclusion



Mixed Membership Models

- Can model overlapping communities
- Efficient to learn from low order moments

Tensor Spectral Method

- GPU/CPU implementation on large dataset with millions of nodes
- Orders of magnitude speed gain than stochastic variational method
- Innovative Evaluation Metric

Questions?

arXiv:1309.0787v3 [cs.LG]

Contact us:

furongh@uci.edu
un.niranjan@uci.edu
a.anandkumar@uci.edu
http://newport.eecs.uci.edu/anandkumar/Lab/Lab.html

Questions?

Thank you!

---

arXiv:1309.0787v3 [cs.LG]

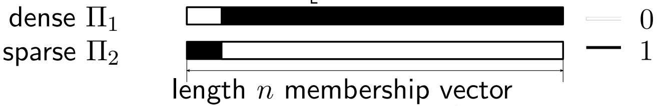
Contact us:

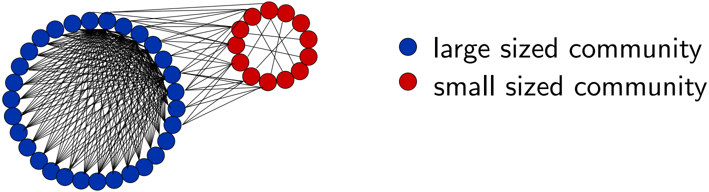
furongh@uci.edu
un.niranjan@uci.edu
a.anandkumar@uci.edu
`http://newport.eecs.uci.edu/anandkumar/Lab/Lab.html`

Comparison With NMI Score

$$H(\widehat{\mathbf{\Pi}}_{\text{mix}}|\mathbf{\Pi}_{\text{mix}})_{\text{norm}} := \frac{1}{k}\sum_{j\in[k]} \min_{i\in[\widehat{k}]} \frac{H\left(\widehat{\Pi}_{\text{mix}_i}|\Pi_{\text{mix}_j}\right)}{H(\Pi_{\text{mix}_j})}$$

$$N_{\text{mix}}(\widehat{\mathbf{\Pi}}_{\text{mix}} : \mathbf{\Pi}_{\text{mix}}) := 1 - \frac{1}{2}\left[H(\mathbf{\Pi}_{\text{mix}}|\widehat{\mathbf{\Pi}}_{\text{mix}})_{\text{norm}} + H(\widehat{\mathbf{\Pi}}_{\text{mix}}|\mathbf{\Pi}_{\text{mix}})_{\text{norm}}\right].$$





- $P(\Pi_{\text{mix}_1} = 0) = \frac{\#\text{ of 0s in } \Pi_1}{n}$ equals to $P(\Pi_{\text{mix}_2} = 1) = \frac{\#\text{ of 1s in } \Pi_2}{n}$.
- $P(\Pi_{\text{mix}_1} = 1) = \frac{\#\text{ of 1s in } \Pi_1}{n}$ equals to $P(\Pi_{\text{mix}_2} = 0) = \frac{\#\text{ of 0s in } \Pi_2}{n}$.
- Therefore, $H(\Pi_{\text{mix}_1}) = H(\Pi_{\text{mix}_2})$.