Implementing Tensor Methods: Application to Community Detection

Anima Anandkumar

U.C. Irvine

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

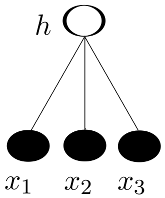
- Hidden State:
  $h \in$ basis $\{e_1, \dots, e_k\}$
  $k = 2$

- Observed States:
  $x_i \in$ basis $\{e_1, \dots, e_d\}$
  $d = 3$

- 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

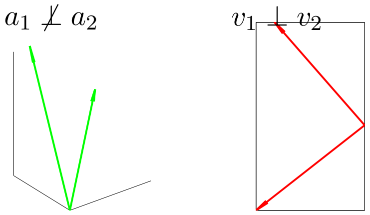
- $y_1^t = W^\top x_1^t$

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

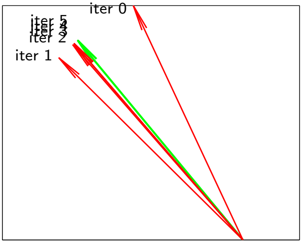
```
V = zeros(k,k); Lambda = zeros(k,1);
for i = 1:k
   v_old = rand(k,1); v_old = normc(v_old);
   for iter = 1 : Maxiter
      v_new = (y1'* ((y2*v_old).*(y3*v_old)))/n;
      if i >1
      % deflation
         for j = 1: i-1
            v_new=v_new-(V(:,j)*(v_old'*V(:,j))2)* Lambda(j);
         end
      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

Applications and Challenges

Social Networks

- Observed: network of social ties,
- Hidden: groups/communities of actors.

Challenges

- Large Scale Networks: $n \sim$ millions or billions
- Latent Communities: $k \sim$ thousands

Topic modeling

- Observed: words in corpus,
- Hidden: topics.

Challenges

- Large Vocabulary: Words $d \sim$ 100,000
- Huge Corpus: Documents $n \sim$ millions
- Latent Topics: $k \sim$ thousands

Resources for this talk

Papers

- "Fast Detection of Overlapping Communities via Online Tensor Methods" by F. Huang, U. N. Niranjan, M. U. Hakeem, A., Preprint, Sept. 2013.
- "Tensor Decompositions on REEF," F. Huang, S. Matusevych, N. Karampatziakis, P. Mineiro, A. , under preparation.

Code

- GPU and CPU codes: `github.com/FurongHuang/Fast-Detection-of-Overlapping-Communities-via-Online-Tens`
- REEF code will be released soon.

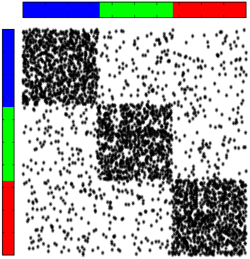
Outline

1. Recap: A Toy Example via MATLAB
2. Community Detection Implementation
   - Subgraph Counts as Graph Moments
   - Whitening
   - Tensor Decomposition
   - Code Optimization
   - Experimental Results
3. Implementing In the Cloud
4. Conclusion

Outline

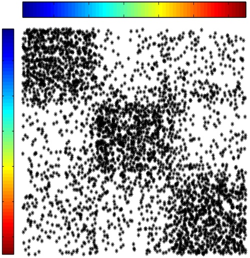
Mixed Membership Community Models

Stochastic Block Model



$\alpha_0 = 0$

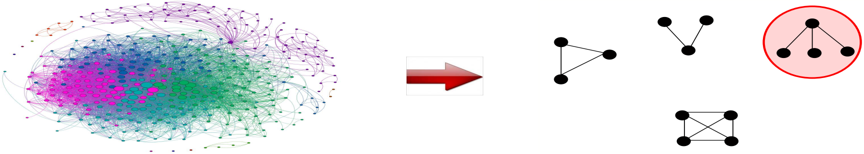
Mixed Membership Model



$\alpha_0 = 1$

Goal: Recover communities Π given adjacency matrix G

Subgraph Counts as Graph Moments



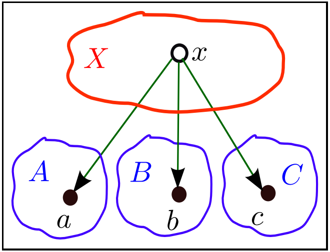
3-star counts sufficient for identifiability and learning of MMSB

3-Star Count Tensor

$$\tilde{M}_3(a,b,c) = \frac{1}{|X|}\#\text{ of common neighbors in } X$$

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

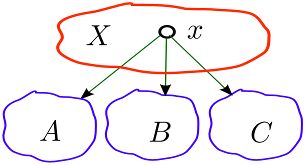
$$\tilde{M}_3 = \frac{1}{|X|}\sum_{x\in X}\left[G_{x,A}^\top \otimes G_{x,B}^\top \otimes G_{x,C}^\top\right]$$



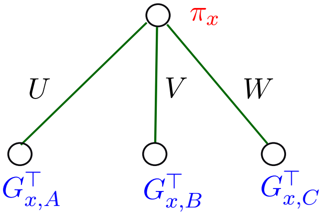
Multi-view Representation

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

3-stars



Graphical model



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

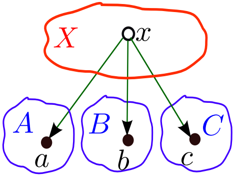
Subgraph Counts as Graph Moments

Second and Third Order Moments

- $\boxed{\hat{M}_2 := \frac{1}{|X|} \sum_x Z_C G_{x,C}^\top G_{x,B} Z_B^\top - \text{shift}}$
- $\boxed{\hat{M}_3 := \frac{1}{|X|} \sum_x \left[G_{x,A}^\top \otimes Z_B G_{x,B}^\top \otimes Z_C G_{x,C}^\top \right] - \text{shift}}$

Symmetrize Transition Matrices

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



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

$$\boxed{\mathbb{E}[\hat{M}_2 | \Pi_{A,B,C}] = \sum_i \frac{\alpha_i}{\alpha_0} u_i \otimes u_i, \quad \mathbb{E}[\hat{M}_3 | \Pi_{A,B,C}] = \sum_i \frac{\alpha_i}{\alpha_0} u_i \otimes u_i \otimes u_i.}$$

Overview of Tensor Method

- Whiten data via SVD of $\hat{M}_2 \in \mathbb{R}^{n \times n}$.
- Estimate the third moment $\hat{M}_3 \in \mathbb{R}^{n \times n \times n}$ and whiten it implicitly to obtain T.
- Run power method (gradient ascent) on T.
- Apply post-processing to obtain communities.
- Compute error scores and validate with ground truth (if available).

Outline

Whitening Matrix Computation

Symmetrization: Finding Second Order Moments M_2

$$\hat{M}_2 = \boxed{Z_C}\,\mathrm{Pairs}_{C,B}\,\boxed{Z_B^\top} - \text{shift}$$

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

$\boxed{\text{Challenges: } n \times n \text{ objects, } n \sim \text{millions or billions}}$

Whitening Matrix Computation

Symmetrization: Finding Second Order Moments M_2

$$\hat{M}_2 = \boxed{Z_C}\, \mathrm{Pairs}_{C,B}\, \boxed{Z_B^\top} - \text{shift}$$

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

Challenges: $n \times n$ objects, n ∼millions or billions

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

Whitening Matrix Computation

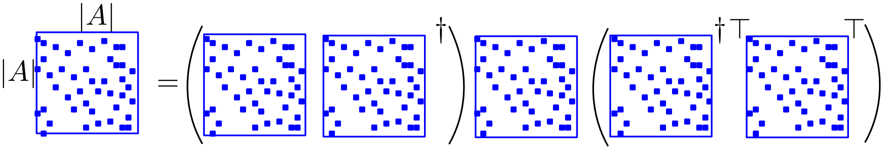
Symmetrization: Finding Second Order Moments M_2

$$\hat{M}_2 = \boxed{Z_C}\,\mathrm{Pairs}_{C,B}\,\boxed{Z_B^\top} - \text{shift}$$

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

Challenges: $n \times n$ objects, n ~millions or billions

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.

Space Complexity $O(nk)$

Whitening Matrix Computation

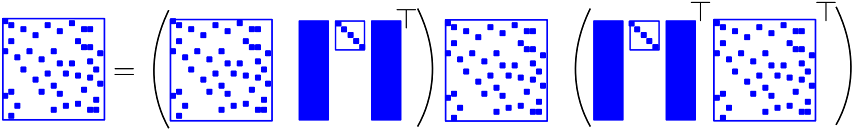
Symmetrization: Finding Second Order Moments M_2

$$\hat{M}_2 = \boxed{Z_C}\,\mathrm{Pairs}_{C,B}\,\boxed{Z_B^\top} - \text{shift}$$

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

Challenges: $n \times n$ objects, n ~millions or billions

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.

Space Complexity $O(nk)$

Whitening Matrix Computation

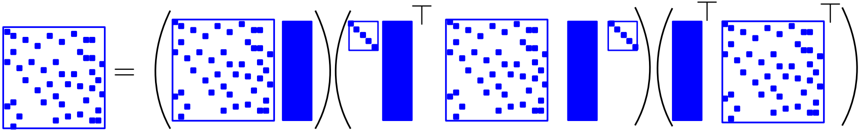
Symmetrization: Finding Second Order Moments M_2

$$\hat{M}_2 = \boxed{Z_C}\,\mathrm{Pairs}_{C,B}\,\boxed{Z_B^\top} - \text{shift}$$

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

Challenges: $n \times n$ objects, n ~millions or billions

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.

Space Complexity $O(nk)$

Whitening Matrix Computation

Orthogonalization: Finding Whitening Matrix W

$W^T M_2 W = I$ is solved by $\boxed{\text{k-svd}(M_2)}$

$\boxed{\text{Challenges: } n \times n \text{ Matrix SVDs, } n \sim \text{millions or billions}}$

Whitening Matrix Computation

Orthogonalization: Finding Whitening Matrix W

$W^T M_2 W = I$ is solved by $\boxed{\text{k-svd}(M_2)}$

$\boxed{\text{Challenges: } n \times n \text{ Matrix SVDs, } n \sim \text{millions or billions}}$

Randomized low rank approx. (GM 13', CW 13')

- Random matrix $S \in \mathbb{R}^{n \times \tilde{k}}$ for dense M_2
- Column selection matrix: random signs $S \in \{0,1\}^{n \times \tilde{k}}$ for sparse M_2.
- $Q=$ orth$(M_2 S)$, $Z = (M_2 Q)^\top M_2 Q$
- $[U_z, L_z, V_z] =$SVD(Z) % $Z \in \mathbb{R}^{k \times k}$
- $V_{M_2} = M_2 Q V_z L_z^{-\frac{1}{2}}$, $L_{M_2} = L_z^{\frac{1}{2}}$

Whitening Matrix Computation

Orthogonalization: Finding Whitening Matrix W

$W^T M_2 W = I$ is solved by $\boxed{\text{k-svd}(M_2)}$

$\boxed{\text{Challenges: } n \times n \text{ Matrix SVDs, } n \sim \text{millions or billions}}$

Randomized low rank approx. (GM 13', CW 13')

- Random matrix $S \in \mathbb{R}^{n \times \tilde{k}}$ for dense M_2
- Column selection matrix: random signs $S \in \{0,1\}^{n \times \tilde{k}}$ for sparse M_2.
- Q= orth$(M_2 S)$, $Z = (M_2 Q)^\top M_2 Q$
- $[U_z, L_z, V_z]$ =SVD(Z) % $Z \in \mathbb{R}^{k \times k}$
- $V_{M_2} = M_2 Q V_z L_z^{-\frac{1}{2}}$, $L_{M_2} = L_z^{\frac{1}{2}}$

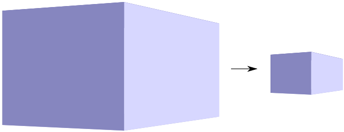
Computational Complexity

- For exact rank-k SVD of $n \times n$ matrix: $O(n^2 k)$.
- For randomized SVD with c cores and sparsity level s per row of M_2:

$$\boxed{\text{Time Complexity } O(nsk/c + k^3)}$$

Outline

Using Whitening to Obtain Orthogonal Tensor



Tensor M_3 Tensor T

Multi-linear transform

- $M_3 \in \mathbb{R}^{n \times n \times n}$ 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 $k \ll n$.

Batch Gradient Descent

Power Iteration with Deflation

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

Alternating Least Squares

$$\min_{\sigma, A, B, C} \left\| T - \sum_{i=1}^{k} \lambda_i A(:,i) \otimes B(:,i) \otimes C(:,i) \right\|_F^2$$

such that $A^\top A = I$, $B^\top B = I$ and $C^\top C = I$.

Challenges:

$\boxed{\text{Requires forming the tensor/passing over data in each iteration}}$

Stochastic (Implicit) Tensor Gradient Descent

Whitened third order moments:

$$T = M_3(W, W, W).$$

Objective:

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

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

Stochastic (Implicit) Tensor Gradient Descent

Whitened third order moments:

$$T = M_3(W, W, W).$$

Objective:

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

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

Expand the objective:

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

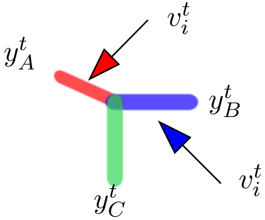
Orthogonality cost vs Correlation Reward

Stochastic (Implicit) Tensor Gradient Descent

Updating Equation

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

Orthogonality cost vs Correlation Reward



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

Space: $O(k^2)$, Time: $O(k^3/c)\times$ iterations with c cores.

Unwhitening

Post Processing for memberships

- Λ: eigenvalues. Φ: eigenvectors.
- G: adjacency matrix, γ: normalization.
- W: Whitening Matrix.

$$\hat{\Pi}_{A^c} = \operatorname{diag}(\gamma)^{1/3} \operatorname{diag}(\Lambda)^{-1} \Phi^\top W^\top G_{A,A^c},$$

where $A^c := X \cup B \cup C$.

- Threshold the values.

Space Complexity $O(nk)$

Time Complexity $O(nsk/c)$ with c cores.

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	$O(nsk/c + k^3)$	$O(Nk^3/c)$	$O(nsk/c)$	$O(mkN)$

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

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

Our approach: $O(nsk/c + k^3)$

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	$O(nsk/c + k^3)$	$O(Nk^3/c)$	$O(nsk/c)$	$O(mkN)$

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

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

Our approach: $O(nsk/c + k^3)$

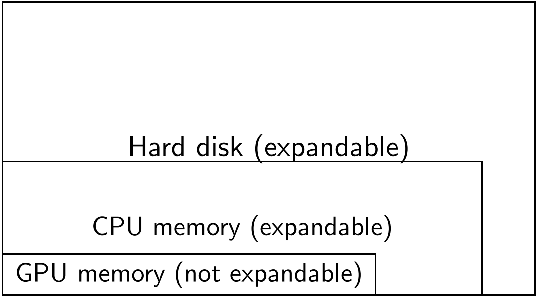
In practice STGD is extremely fast and is not the bottleneck

Outline

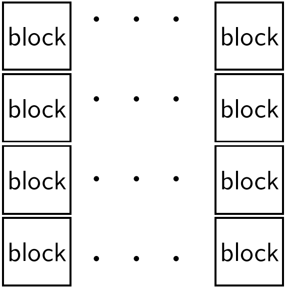
GPU/CPU Implementation

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

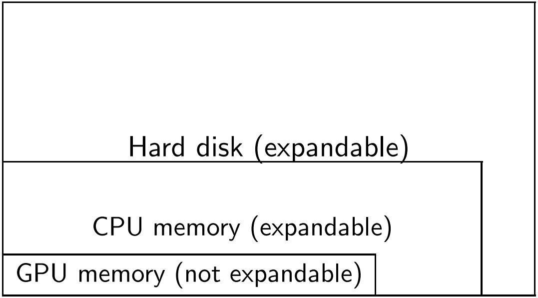


Partitioned matrix

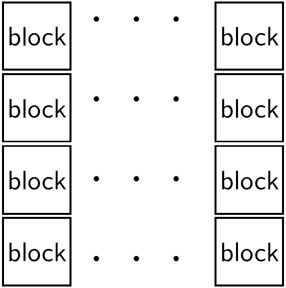
GPU/CPU Implementation

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



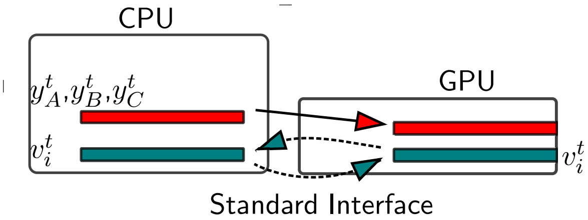
Partitioned matrix

CPU

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

Scaling Of The Stochastic Iterations

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

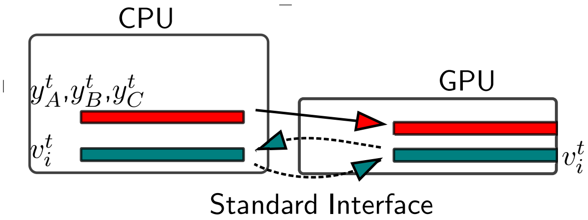


- Parallelize across eigenvectors.
- STGD is iterative: device code reuse buffers for updates.

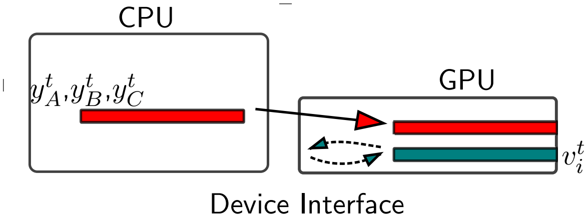
Scaling Of The Stochastic Iterations

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

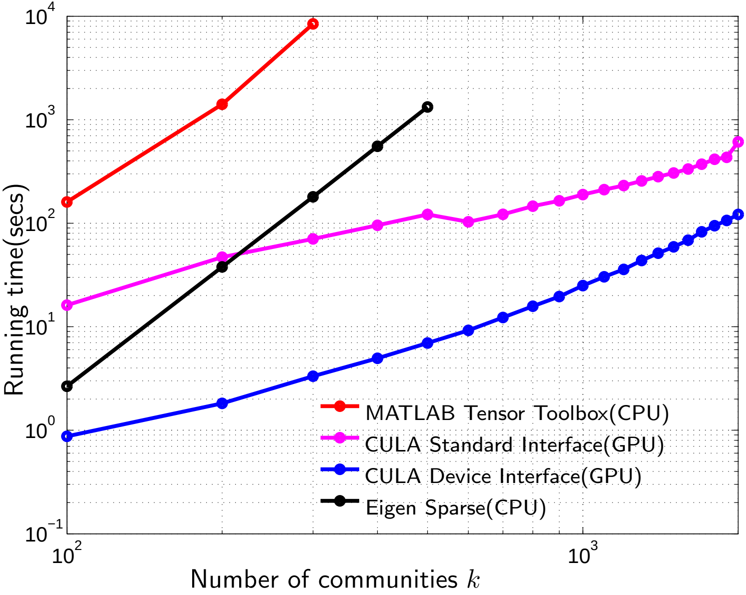
- Parallelize across eigenvectors.



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



Scaling Of The Stochastic Iterations



Validation Metrics

Ground-truth membership available

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

Validation Metrics

Ground-truth membership available

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

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

Validation Metrics

Ground-truth membership available

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

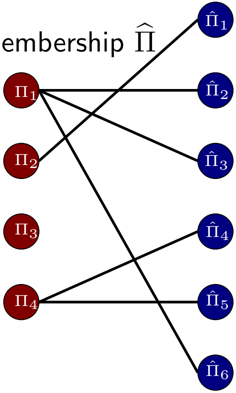
Problem: How to relate Π and $\widehat{\Pi}$?

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

Validation Metrics

Ground-truth membership available

- Ground-truth membership matrix Π vs 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: $\mathcal{E} := \frac{1}{nk}\sum\{\text{paired membership errors}\}$

$$= \frac{1}{k}\sum_{(i,j)\in E_{\{\mathsf{P}_{\mathsf{val}}\}}}\left\{\frac{1}{n}\sum_{x\in|X|}|\widehat{\Pi}_i(x)-\Pi_j(x)|\right\}$$

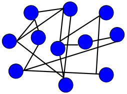
Insights

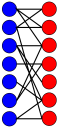
- l_1 norm error between $\widehat{\Pi_i}$ and the corresponding paired Π_j
- false pairings penalization

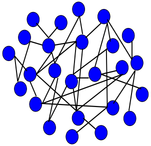
  too many falsely discovered pairings, error > 1

Outline

Summary of Results







Facebook	Yelp	DBLP(sub)
$n \sim 20k$	$n \sim 40k$	$n \sim 1$ 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.

Experimental Results on Yelp

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

Experimental Results on Yelp

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	Restaurants, Bars, American, Nightlife, Food, Pubs, Tempe
Pizzeria Bianco	Restaurants, Pizza, Phoenix
FEZ	Restaurants, Bars, American, Nightlife, Mediterranean, Lounges, Phoenix
Matt's Big Breakfast	Restaurants, Phoenix, Breakfast& Brunch
Cornish Pasty Co	Restaurants, Bars, Nightlife, Pubs, Tempe

Outline

1. Recap: A Toy Example via MATLAB
2. Community Detection Implementation
   - Subgraph Counts as Graph Moments
   - Whitening
   - Tensor Decomposition
   - Code Optimization
   - Experimental Results
3. Implementing In the Cloud
4. Conclusion

Review of linear algebra

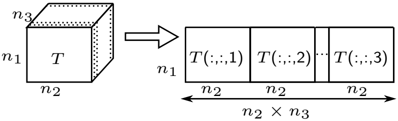
Tensor Modes

- Analogy to Matrix Rows and Matrix Columns.
- For an order-d tensor $A \in \mathbb{R}^{n_1 \times n_2 \ldots n_d}$:

  mode-1 has dimension n_1,
  mode-2 has dimension n_2, and so on.

Tensor Unfolding

In a mode-k unfolding, the mode-k fibers are assembled to produce an n_k-by-N/n_k matrix where $N = n_1 \ldots n_d$.



- Mode-1 Unfolding of $A \in \mathbb{R}^{2\times2\times2} = \begin{bmatrix} a_{111} & a_{121} & a_{112} & a_{122} \\ a_{211} & a_{221} & a_{212} & a_{222}. \end{bmatrix}$

Tensor Decomposition In The Cloud

- Tensor decomposition is equivalent to

$$\min_{\sigma,A,B,C} \left\| T - \sum_{i=1}^{k} \sigma_i A(:,i) \otimes B(:,i) \otimes C(:,i) \right\|_F^2$$

Tensor Decomposition In The Cloud

- Tensor decomposition is equivalent to

$$\min_{\sigma,A,B,C} \left\| T - \sum_{i=1}^{k} \sigma_i A(:,i) \otimes B(:,i) \otimes C(:,i) \right\|_F^2$$

- Alternating Least Square is the solution:

$$A' \leftarrow T_a f(C,B)\left(C^\top C \star B^\top B\right)^\dagger$$

$$B' \leftarrow T_b f(C,A')\left(C^\top C \star A'^\top A'\right)^\dagger$$

$$C' \leftarrow T_c f(B',A')\left(B'^\top B' \star A'^\top A'\right)^\dagger$$

where T_a is the mode-1 unfolding of T, T_b is the mode-2 unfolding of T, and T_c is the mode-3 unfolding of T.

Challenges I

How to parallelize?

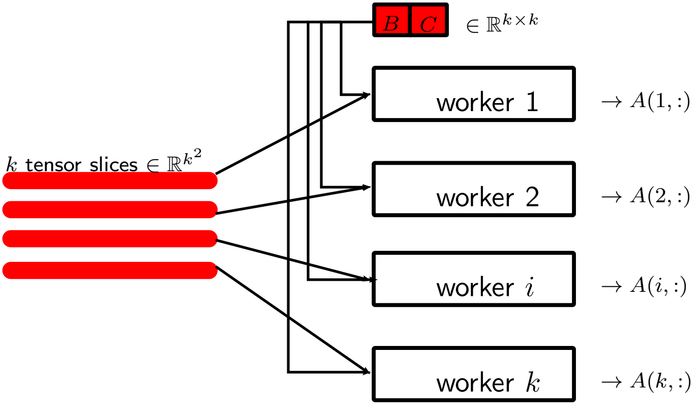
- Observations: $A'(i,:) \leftarrow T_a(i,:) f(C,B) \left(C^\top C \star B^\top B\right)^\dagger$
- $T_a \in \mathbb{R}^{k\times k^2}$, B and $C \in \mathbb{R}^{k\times k}$

Challenges I

How to parallelize?

- Observations: $A'(\color{red}{i},:) \leftarrow T_a(\color{red}{i},:) f(C,B) \left(C^\top C \star B^\top B\right)^\dagger$
- $T_a \in \mathbb{R}^{k \times k^2}$, B and $C \in \mathbb{R}^{k \times k}$

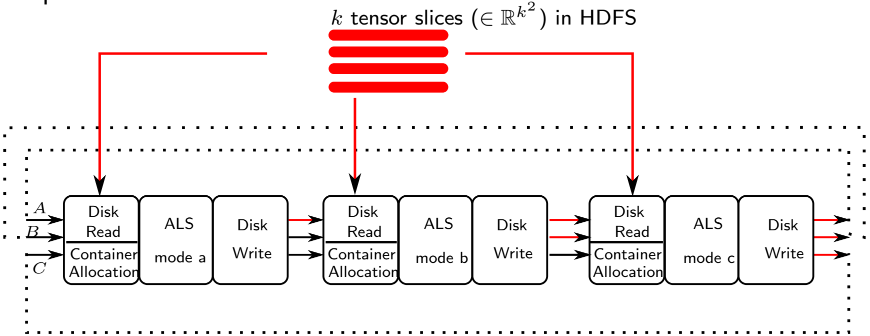
Update Rows Independently



Challenges II

Communication and System Architecture Overhead

- Map-Reduce Framework

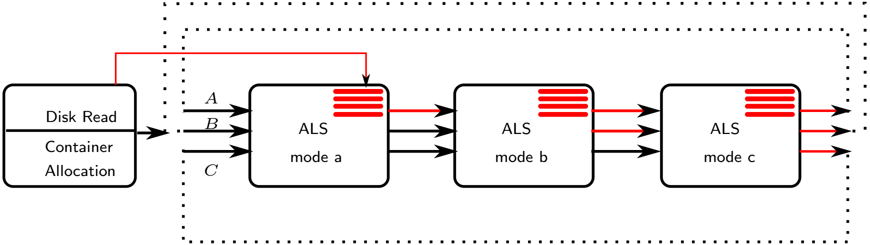


- Overhead: Disk reading, Container Allocation, Intense Key/Value Design

Challenges II

Solution: REEF

- Big data framework called REEF (Retainable Evaluator Execution Framework)



- Advantage: Open source distributed system with one time container allocation , keep the tensor in memory

Correctness

Evaluation Score

$$\text{perplexity} := \exp\left(-\frac{\sum_i \text{log-likelihood in doc } i}{\sum_i \text{words in doc } i}\right)$$

New York Times Corpus

- Documents $n = 300,000$
- Vocabulary $d = 100,000$
- Topics $k = 100$

	Stochastic Variational Inference	Tensor Decomposition
Perplexity	4000	3400

SVI drawbacks:

- Hyper parameters
- Learning rate
- Initial points

Running Time

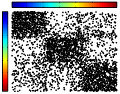
Computational Complexity

Complexity	Whitening	Tensor Slices $(1,\ldots,k)$	ALS
Time	$O(k^3)$	$O(k^2)$ per slice	$O(k^3)$
Space	$O(kd)$	$O(k^2)$ per slice	$O(k^2)$
Degree of Parallelism	∞	∞ per slice	k
Communication	$O(kd)$	$O(k^2)$	$O(k^2)$

	SVI	1 node Map Red	1 node REEF	4 node REEF
overall	2 hours	4 hours 31 mins	68 mins	36 mins
Whiten		16 mins	16 mins	16 mins
Matricize		15 mins	15 mins	4 mins
ALS		4 hours	37 mins	16 mins

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...

In practice

- Scalable and embarrassingly parallel: handle large datasets.
- Efficient performance: perplexity or ground truth validation.

Theoretical guarantees and promising practical performance