Sylvester Tensor Equation for Multi-Way Association

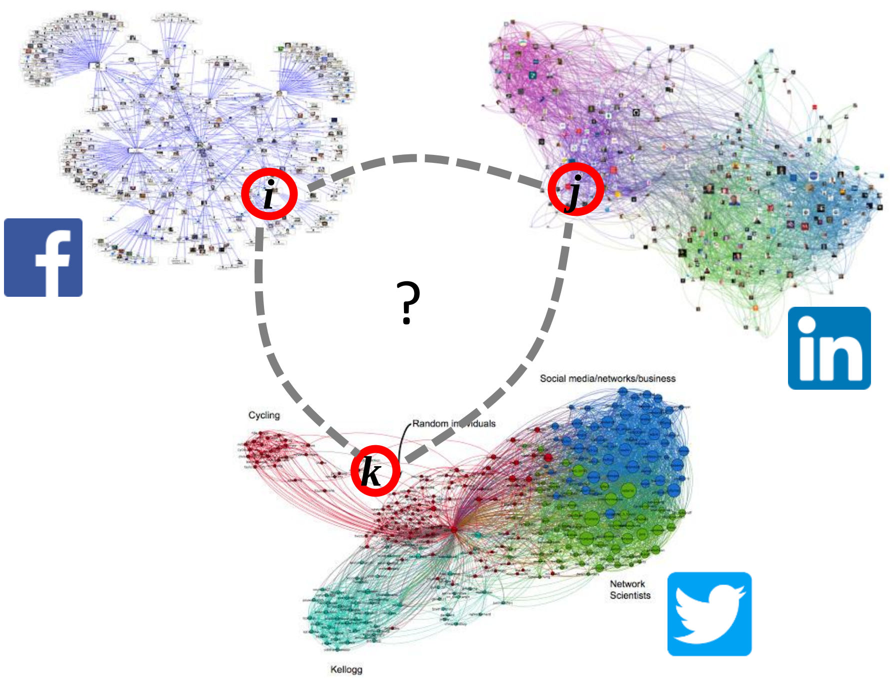


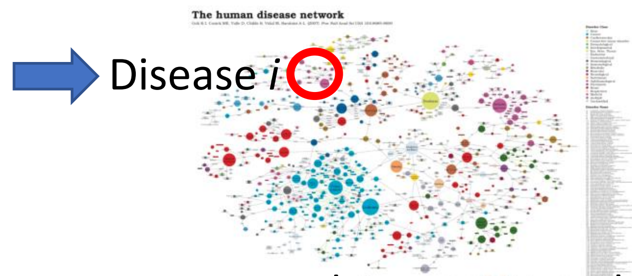
Boxin Du* Lihui Liu* Hanghang Tong*

* University of Illinois at Urbana-Champaign

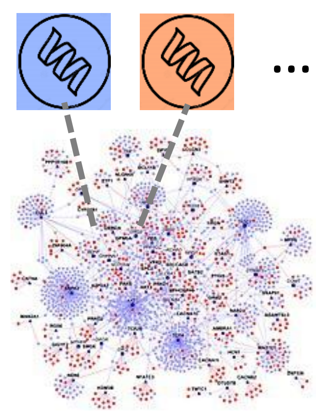
Multi-network Mining Examples

- Link identical/similar users from multiple social networks.
- Discover relevant drugs and genes for a specific disease.

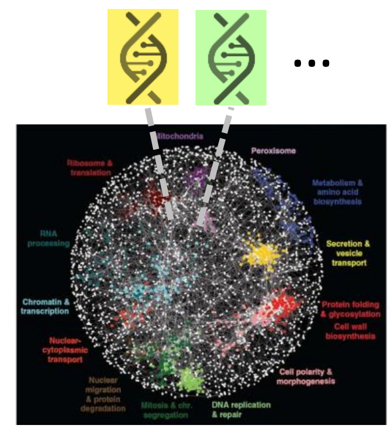




Human disease network



PPI network



Genetic network

- Chu, Xiaokai, et al. "Cross-network embedding for multi-network alignment." The world wide web conference. 2019.
- Xun, Guangxu, et al. "Generating medical hypotheses based on evolutionary medical concepts." *2017 IEEE International Conference on Data Mining (ICDM)*. IEEE, 2017.

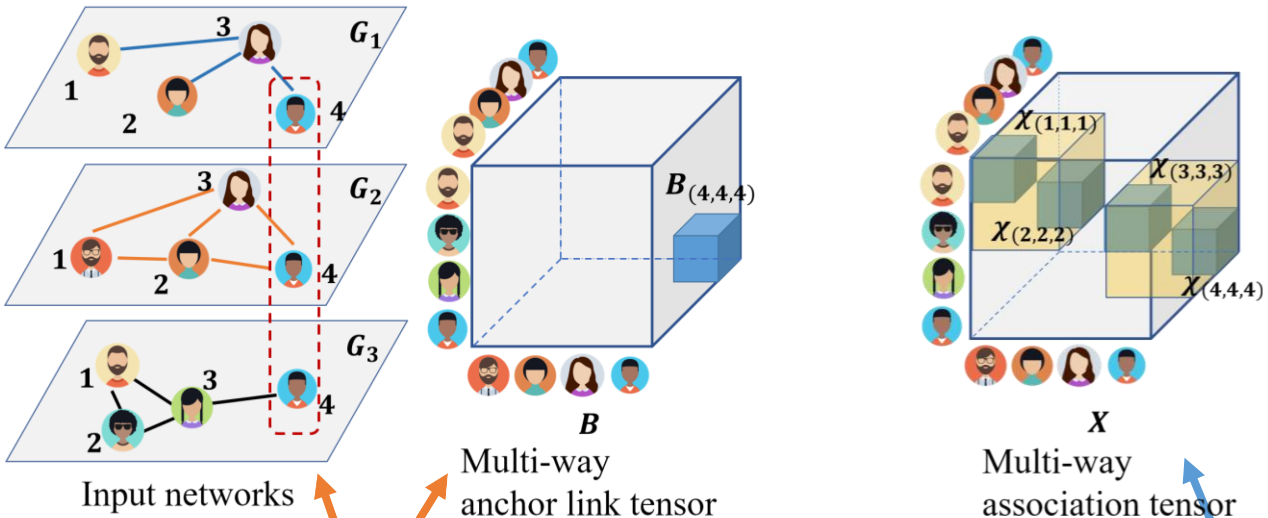
Multi-network Mining Examples (cont'd)

- Recommend items, activities and locations to a user simultaneously.
- Assign team members with the right skills to the right teams.

- Meng Jiang, Peng Cui, Fei Wang, Qiang Yang, Wenwu Zhu, and Shiqiang Yang. 2012. Social recommendation across multiple relational domains. In Proceedingsof the 21st ACM international conference on Information and knowledge management.
- Liangyue Li, Hanghang Tong, Nan Cao, Kate Ehrlich, Yu-Ru Lin, and Norbou Buchler. 2015. Replacing the irreplaceable: Fast algorithms for team member recommendation. In Proceedings of the 24th International Conference on World Wide Web.

Multi-way Association

- Aims to discover the collective association w.r.t. to ***a set of*** nodes.
- Identifies strongly correlated nodes from multiple networks.



Red: known to be associated with each other a priori

Blue blocks: indicate the inferred strongly associated users

Roadmap

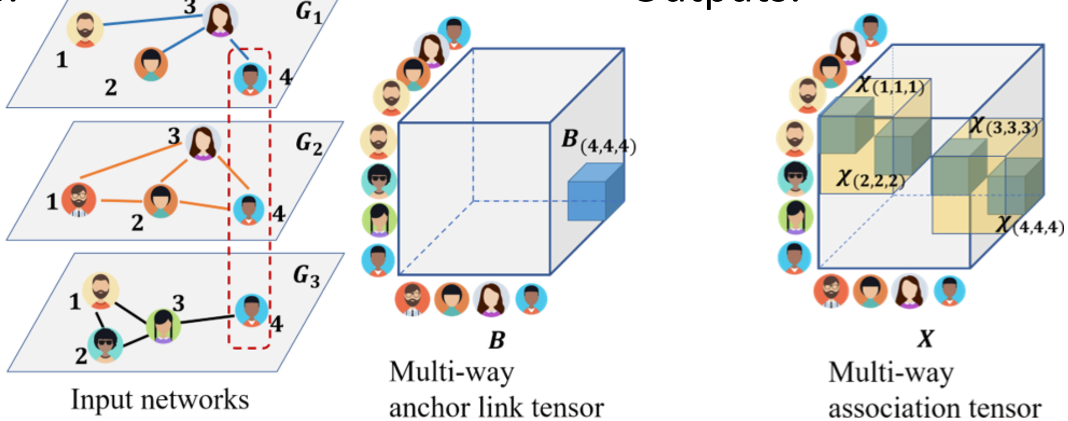
- Motivation ✓
- Problem Definition ←
- Formulation
- Proposed Algorithm
- Experimental Results
- Conclusion

Problem Definition

- Given:
  - A set of K networks $\{G_k\ (k = 1, \dots, K)\}$ (with node number n_k).
  - A multi-way anchor association tensor $\boldsymbol{\mathcal{B}}$.
- Output: Multi-way association tensor $\boldsymbol{\mathcal{X}}$
  - Entries of $\boldsymbol{\mathcal{X}}$: the strength of multi-way association.
  - Multi-way association: collective association of a node set.

Inputs:

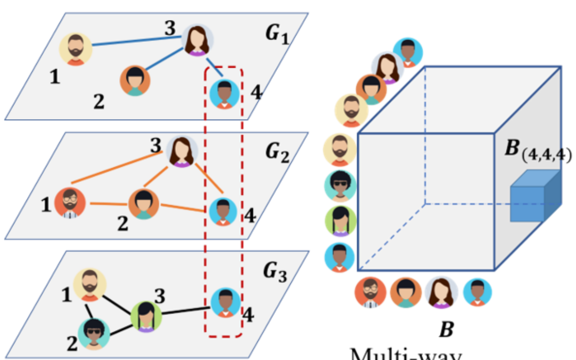
Outputs:



Challenges

- C1. Problem formulation:
  - How to generalize the consistency principle to multi-way association?
- C2. Algorithms:
  - How to solve the formulation in terms of its optimality and sensitivity?
- C3. Scalability:
  - How to deal with the significantly large solution space?

Inputs:

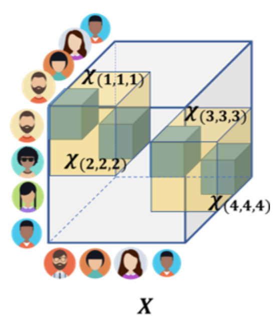


Input networks

Multi-way anchor link tensor

Assuming n nodes in each input network

Outputs:



Multi-way association tensor

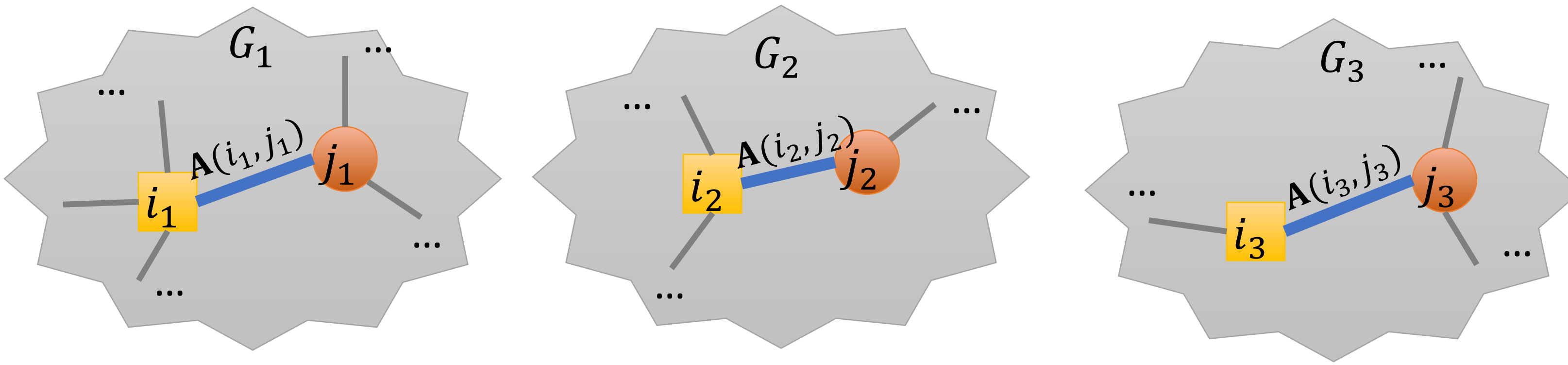
For K input networks, the number of multi-way association is n^K.

Roadmap

- Motivation ✓
- Problem Definition ✓
- Formulation ←
- Proposed Algorithm
- Experimental Results
- Conclusion

Intuition: Topology Consistency

- Multi-way association $\mathcal{X}(i_K, \ldots, i_1)$ is close to $\mathcal{X}(j_K, \ldots, j_1)$ if node set $\{i_1, \ldots, i_K\}$ and $\{j_1, \ldots, j_K\}$ satisfy:
  - (1) Two sets of nodes are strongly connected



- Large $\mathbf{A_1}(i_1, j_1)$, $\mathbf{A_2}(i_2, j_2)$ and $\mathbf{A_3}(i_3, j_3)$
- Large $\mathcal{X}(i_3, i_2, i_1)$

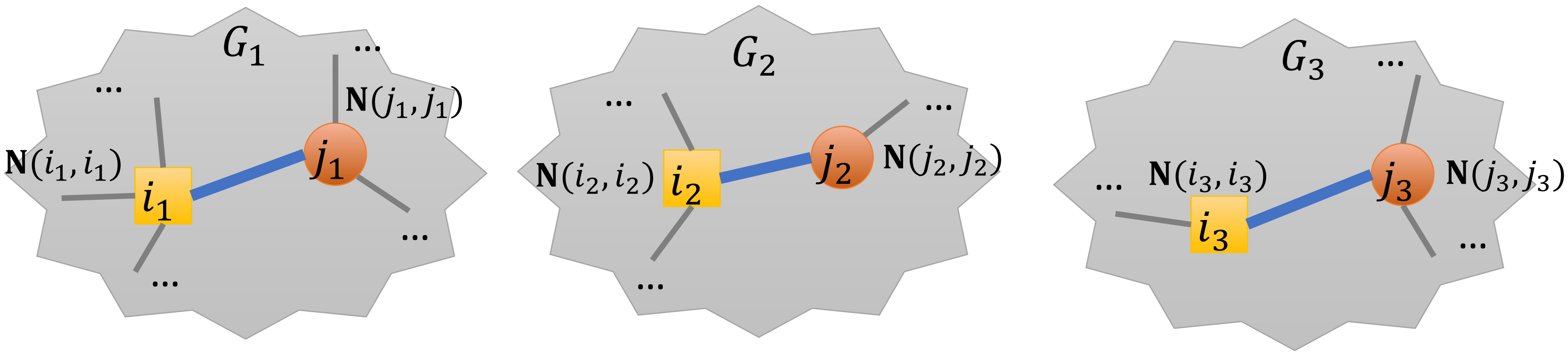
→ Large $\mathcal{X}(j_3, j_2, j_1)$

- Mathematically, $min\left(\mathcal{X}(i_3, i_2, i_1) - \mathcal{X}(j_3, j_2, j_1)\right)^2 \mathbf{A_1}(i_1, j_1)\mathbf{A_2}(i_2, j_2)\mathbf{A}_3(i_3, j_3)$

- Si Zhang and Hanghang Tong. 2016. Final: Fast attributed network alignment. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, 1345–1354.
- Rohit Singh, Jinbo Xu, and Bonnie Berger. 2008. Global alignment of multiple protein interaction networks with application to functional orthology detection. Proceedings of the National Academy of Sciences 105, 35 (2008), 12763–12768

Intuition: Node Attribute Consistency

- Multi-way association $\mathcal{X}(i_K, \ldots, i_1)$ is close to $\mathcal{X}(j_K, \ldots, j_1)$ if node set $\{i_1, \ldots, i_K\}$ and $\{j_1, \ldots, j_K\}$ satisfy:
  - (2) Nodes in each of the two sets share the same attribute respectively



- Large $\mathcal{X}(i_3, i_2, i_1)$
- Large $\mathcal{X}(j_3, j_2, j_1)$

⇒ $\{i_1, i_2, i_3\}$, $\{j_1, j_2, j_3\}$: same node attribute

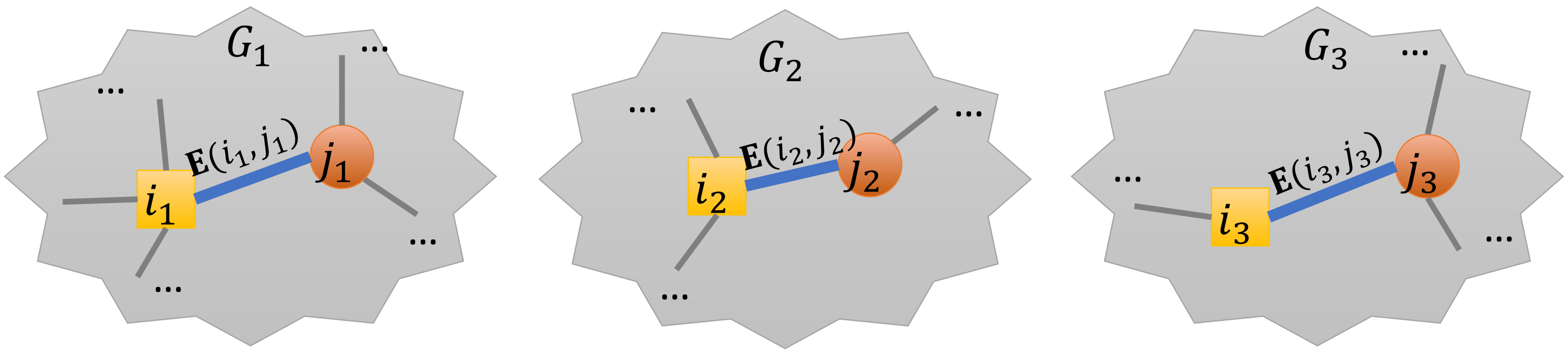
- Mathematically, $min\big(\mathcal{X}(i_3, i_2, i_1) - \mathcal{X}(j_3, j_2, j_1)\big)^2 \mathbf{A_1}(i_1, j_1)\mathbf{A_2}(i_2, j_2)\mathbf{A}_3(i_3, j_3)$

if $\mathbf{N}(i_1, i_1) = \mathbf{N}(i_2, i_2) = \mathbf{N}(i_3, i_3)$

- Si Zhang and Hanghang Tong. 2016. Final: Fast attributed network alignment. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, 1345–1354.
- Rohit Singh, Jinbo Xu, and Bonnie Berger. 2008. Global alignment of multiple protein interaction networks with application to functional orthology detection. Proceedings of the National Academy of Sciences 105, 35 (2008), 12763–12768

Intuition: Edge Attribute Consistency

- Multi-way association $\mathcal{X}(i_K, \dots, i_1)$ is close to $\mathcal{X}(j_K, \dots, j_1)$ if node set $\{i_1, \dots, i_K\}$ and $\{j_1, \dots, j_K\}$ satisfy:
  - (3) Nodes from node sets are connected by the same edge attribute



- Large $\mathcal{X}(i_3, i_2, i_1)$
- Large $\mathcal{X}(j_3, j_2, j_1)$

⇒ $\{i_1, j_1\}, \{i_2, j_2\}, \{i_3, j_3\}$: same edge attribute

- Mathematically, $min\left(\mathcal{X}(i_3, i_2, i_1) - \mathcal{X}(j_3, j_2, j_1)\right)^2 \mathbf{A_1}(i_1, j_1)\mathbf{A_2}(i_2, j_2)\mathbf{A}_3(i_3, j_3)$

if $\mathbf{E}(i_1, j_1) = \mathbf{E}(i_2, j_2) = \mathbf{E}(i_3, j_3)$

- Si Zhang and Hanghang Tong. 2016. Final: Fast attributed network alignment. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, 1345–1354.
- Rohit Singh, Jinbo Xu, and Bonnie Berger. 2008. Global alignment of multiple protein interaction networks with application to functional orthology detection. Proceedings of the National Academy of Sciences 105, 35 (2008), 12763–12768

Formulation

- Objective function:

$$J(\boldsymbol{\mathcal{X}}) = \sum_{\substack{i_1,\dots,i_K \\ j_1,\dots,j_K}} [\beta \left(\frac{\boldsymbol{\mathcal{X}}(i_K,\dots,i_1)}{\sqrt{d(i_1,\dots,i_K)}} - \frac{\boldsymbol{\mathcal{X}}(j_K,\dots,j_1)}{\sqrt{d(j_1,\dots,j_K)}} \right)^2 * t(A_1,\dots,A_K) * f(i_k)f(j_k) * g(i_k,j_k) + \gamma\left(\boldsymbol{\mathcal{X}}(i_K,\dots,i_1) - \boldsymbol{\mathcal{B}}(i_K,\dots,i_1)\right)^2]$$

Normalized association smoothness preserver

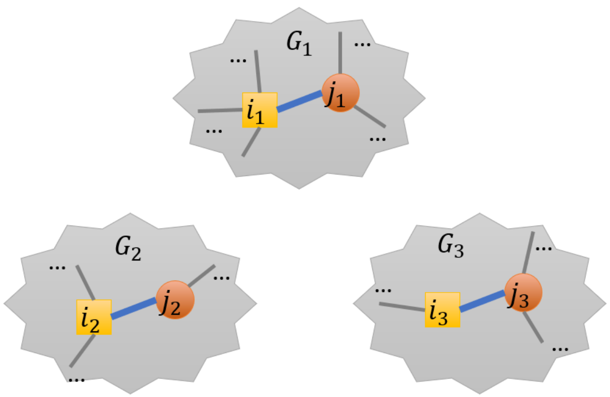
Topology consistency

Edge attribute consistency

Node attribute consistency

Anchor association regularizer

- Details:
  - $t(\mathbf{A}_1,\dots,\mathbf{A}_K) = \mathbf{A}_1(i_1,j_1)\cdots\mathbf{A}_K(i_K,j_K)$
  - $f(i_k) = \mathbb{1}(\mathbf{N}_1(i_1,i_1) = \cdots = \mathbf{N}_K(i_K,i_K))$
  - $g(i_k,j_k) = \mathbb{1}(\mathbf{E}_1(i_1,j_1) = \cdots = \mathbf{E}_K(i_K,j_K))$
  - $d(i_1,\dots,i_K) = \sum_{j_1,\dots,j_K} \mathbf{A}_1(i_1,j_1)\cdots\mathbf{A}_K(i_K,j_K)$
  - β,γ: weighting parameters



Sylvester Tensor Equation

- On plain networks:

$$\boldsymbol{\mathcal{X}} - \alpha \boldsymbol{\mathcal{X}} \times_1 \tilde{\mathbf{A}}_K \times_2 \cdots \times_K \tilde{\mathbf{A}}_1 - (1-\alpha)\boldsymbol{\mathcal{B}} = \mathbf{0}$$

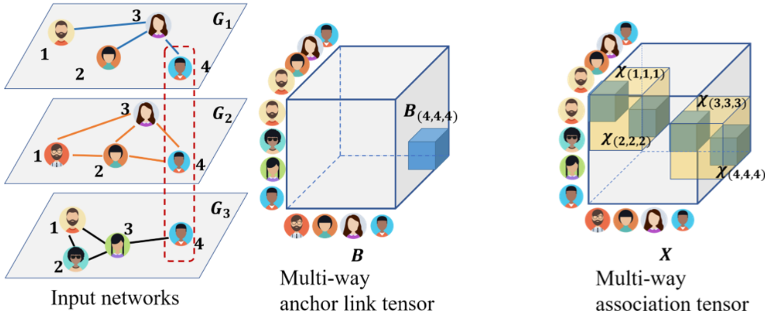
  - where $\tilde{\mathbf{A}}_i = \left(\mathbf{D}_i^{-1/2}\right) \mathbf{A}_i \left(\mathbf{D}_i^{-1/2}\right)$. → Normalization

- Corresponding linear system:
  - $\left(\mathbf{I} - \tilde{\mathbf{A}}_1 \otimes \cdots \otimes \tilde{\mathbf{A}}_K\right)\mathbf{x} = \mathbf{b}$ → $\mathbf{x} = \mathrm{vec}(\boldsymbol{\mathcal{X}}), \mathbf{b} = \mathrm{vec}(\boldsymbol{\mathcal{B}})$

- Explanation:
  - $\boldsymbol{\mathcal{X}} = \underbrace{\alpha \boldsymbol{\mathcal{X}} \times_1 \tilde{\mathbf{A}}_K \times_2 \cdots \times_K \tilde{\mathbf{A}}_1}_{\text{Multi-way association aggregation}} + \underbrace{(1-\alpha)\boldsymbol{\mathcal{B}}}_{\text{Update}}$

Multi-way association aggregation

Update



Sylvester Tensor Equation

- On attributed networks:

$$\boldsymbol{\mathcal{X}} - \alpha \sum_{o,p,q} \boldsymbol{\mathcal{X}} \times_1 \widetilde{\mathbf{A}}_K^{(o,p,q)} \times_2 \cdots \times_K \widetilde{\mathbf{A}}_1^{(o,p,q)} - (1-\alpha)\boldsymbol{\mathcal{B}} = \mathbf{0}$$

  - where $\widetilde{\mathbf{A}}_i^{(o,p,q)} = (\mathbf{D}_i^{-\frac{1}{2}}\mathbf{N}_i^p)(\mathbf{E}_i^o \odot \mathbf{A}_i)(\mathbf{D}_i^{-\frac{1}{2}}\mathbf{N}_i^q)$. → Normalization with node/edge attribute filtering
  - $\mathbf{N}_i^p$: diagonal node attribute matrix for attribute *p*
  - $\mathbf{E}_i^o$: edge attribute matrix for attribute *o*
- Corresponding linear system:
  - $\left(\mathbf{I} - \sum_{o,p,q} \widetilde{\mathbf{A}}_1^{(o,p,q)} \otimes \cdots \otimes \widetilde{\mathbf{A}}_K^{(o,p,q)}\right)\mathbf{x} = \mathbf{b}$
- Q: How to solve the equations accurately and efficiently?

Roadmap

- Motivation
- Problem Definition
- Formulation
- **Proposed Algorithm**
- Experimental Results
- Conclusion

Complexity Summary

Algorithm	Time Complexity	Space Complexity
Fixed Point (FP) [Vishwanathan et al' 10]	$O(n^{3K})$	$O(m^{2K})$
Conjugate Gradient (CG) [Y Saad et al' 03]	$O(n^{3K})$	$O(m^{2K})$
Basic algorithm	$O(cm^{\frac{K}{2}}n^{\frac{K}{2}} + n^K)$	$O(m^{2K})$
*SyTE-Fast-P**	$O(c_1 n^K)$	$O(c_1 m + c_2 n)$
*SyTE-Fast-A**	$O((K + c_1)m + n + c_2 n^K)$	$O(c_1 m + c_2 n)$
SyTE-Fast-P	$O(sKlm + sl^K)$	$O(Km + l^{2K} + Kln)$
SyTE-Fast-A	$O((K + c_1)m + n + c_2 l^K)$	$O(PKm + Kln + l^{2K})$

- Yellow: traditional solver
- Blue: proposed baselines
- Red: proposed algorithm

- n: # of nodes in input network
- m: # of edges in input network
- Other scalars: small constants

- Saad, Yousef. *Iterative methods for sparse linear systems*. Vol. 82. siam, 2003.
- U Kang, Hanghang Tong, Jimeng Sun: Fast Random Walk Graph Kernel. SDM 2012: 828-838

Key Ideas: Plain Networks

- Decompose the equation into a series of subsystems
  - Utilize the sparsity of $\boldsymbol{\mathcal{B}}$ to solve a small number of subsystems
  - Each subsystem is relatively easier to solve by fast algorithm
- Solve each subsystem by a Tensorized Krylov subspace method
  - Tensorized Krylov subspace vs. traditional Krylov subspace: $O(m^K) \rightarrow O(sKlm)$
  - Solve each subsystem by generalized minimal residual method

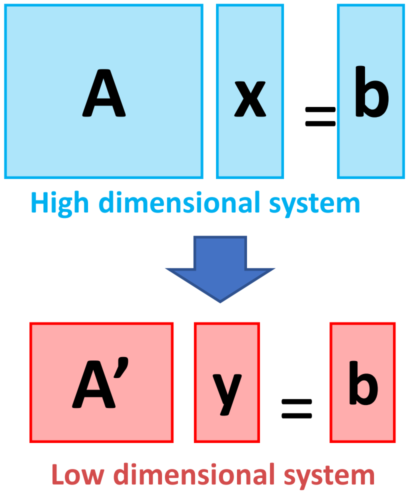
$$(\mathbf{I} - \widetilde{\mathbf{A}}_1 \otimes \cdots \otimes \widetilde{\mathbf{A}}_K)\mathbf{x} = \mathbf{b}$$

Decompose

$$\sum_i (\mathbf{I} - \widetilde{\mathbf{A}}_1 \otimes \cdots \otimes \widetilde{\mathbf{A}}_K)\mathbf{x}_i = \sum_i \otimes_{j=1}^{K} \mathbf{b}_{\mathrm{j}}^{(i)}$$

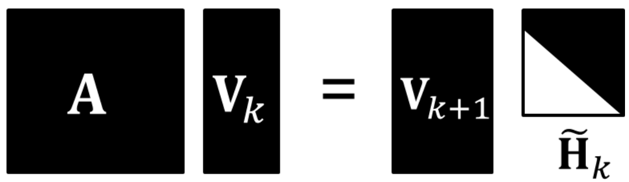
Tensorized Krylov subspace

$$(\otimes_{i=1}^{K} \mathbf{I}_{l_i+1,l_i} - \otimes_{i=1}^{K} \widetilde{\mathbf{H}}_i)\mathbf{y} = \otimes_{i=1}^{K} \mathbf{U}_{l_i+1}^{T} \mathbf{r}_0$$



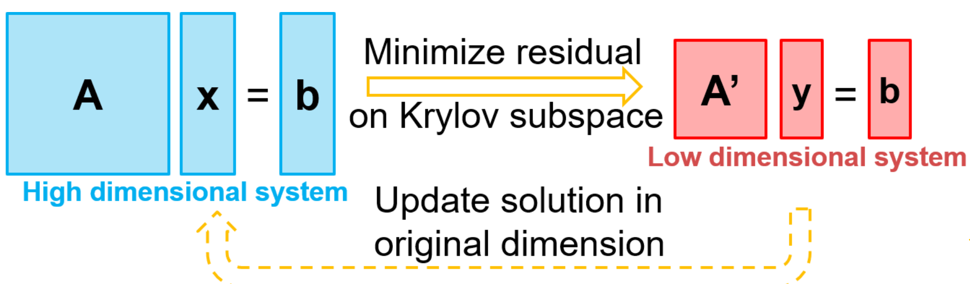
Preliminaries

- Krylov subspace:
  - $K_k(\mathbf{A}, \mathbf{r_0}) = span\{\mathbf{r_0}, \mathbf{A}\mathbf{r_0}, \mathbf{A}^2\mathbf{r_0}, \dots, \mathbf{A}^{k-1}\mathbf{r_0}\}$;
  - Arnoldi process outputs i orthonormal basis: $\mathbf{V}_i = [\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_i], i \in \{k, k+1\}$
  - $\mathbf{A}\mathbf{V}_k = \mathbf{V}_{k+1}\widetilde{\mathbf{H}}_k$



Upper-Hessenberg matrix

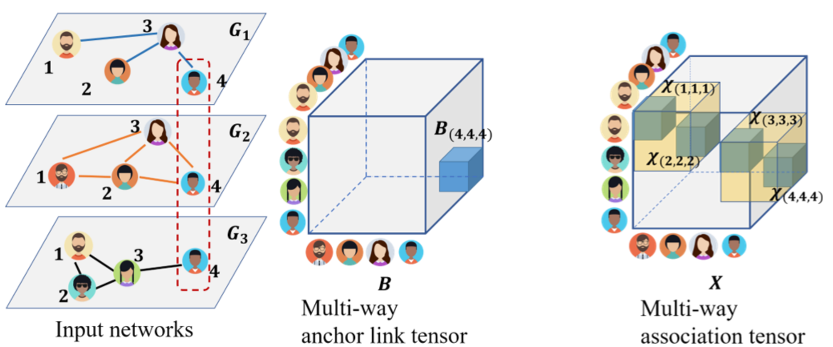
- Krylov subspace-based Minimal Residual method:
  - Extract solution from k-dimensional Krylov subspace (let $K_k = K_k(\mathbf{A}, \mathbf{r_0})$);
  - Minimize the residual $\mathbf{r}$ and update solution at every iteration.



optional

Algorithm: SyTE-Fast-P

- The original linear system for plain networks:
  - $(\mathbf{I} - \widetilde{\mathbf{A}}_1 \otimes \cdots \otimes \widetilde{\mathbf{A}}_K)\mathbf{x} = \mathbf{b}$
  - Weighting parameters absolved into matrices for brevity
- Decomposed subsystems:
  - $\sum_i (\mathbf{I} - \widetilde{\mathbf{A}}_1 \otimes \cdots \otimes \widetilde{\mathbf{A}}_K)\mathbf{x}_i = \sum_i \otimes_{j=1}^{K} \mathbf{b}_{\mathrm{j}}^{(i)}$
  - Since $\mathbf{b} = \sum_{i=1}^{s} \mathbf{b}_1^{(i)} \otimes \cdots \otimes \mathbf{b}_K^{(i)}$, s is the number of non-zeros in $\boldsymbol{\mathcal{B}}$.



$\boldsymbol{\mathcal{B}} = [0,0,0,1]^T \circ [0,0,0,1]^T \circ [0,0,0,1]^T$

$\mathbf{b} = [0,0,0,1]^T \otimes [0,0,0,1]^T \otimes [0,0,0,1]^T$

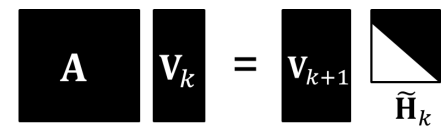
One subsystem: $(\mathbf{I} - \widetilde{\mathbf{A}}_1 \otimes \widetilde{\mathbf{A}}_2 \otimes \widetilde{\mathbf{A}}_3)\mathbf{x} = \mathbf{b}$

Notation:
$\otimes_{j=1}^{K} \mathbf{b}_{\mathrm{j}}^{(i)} = \mathbf{b}_1^{(i)} \otimes \cdots \otimes \mathbf{b}_K^{(i)}$

Algorithm: SyTE-Fast-P (cont'd)

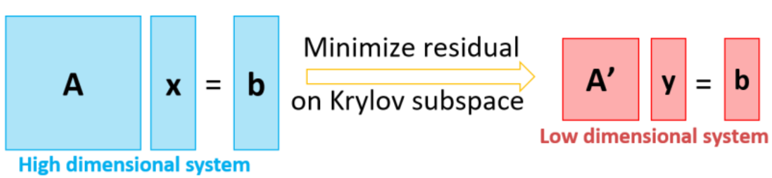
- The Tensorized Krylov subspace:
  - $\mathcal{K}_L^{\otimes}(\mathbf{A}_\times, \mathbf{b}) = span(\mathcal{K}_{l_1}(\widetilde{\mathbf{A}}_1, \mathbf{b}_1) \otimes \cdots \otimes \mathcal{K}_{l_K}(\widetilde{\mathbf{A}}_K, \mathbf{b}_K))$
  - E.g., For $\mathcal{K}_{l_i}(\widetilde{\mathbf{A}}_i, \mathbf{b}_i)$: $\mathbf{U}_{l_i+1}^T \widetilde{\mathbf{H}}_i = \widetilde{\mathbf{A}}_i \mathbf{U}_{l_i}$

Complexity: $O(sKlm)$



- Properties of Tensorized Krylov subspace:
  - $\otimes_{i=1}^K \mathbf{U}_{l_i}^{(i)}$ forms the orthonormal basis of $\mathcal{K}_L^{\otimes}(\mathbf{A}_\times, \mathbf{b})$
  - The original Krylov subspace $\mathcal{K}_L\ (\mathbf{A}_\times, \mathbf{b})$ is contained in $\mathcal{K}_L^{\otimes}(\mathbf{A}_\times, \mathbf{b})$
- The small-scaled system:
  - $(\otimes_{i=1}^K \mathbf{I}_{l_i+1,l_i} - \otimes_{i=1}^K \widetilde{\mathbf{H}}_i)\mathbf{y} = \otimes_{i=1}^K \mathbf{U}_{l_i+1}^T \mathbf{r}_0$
  - Coefficient matrix: Hessenberg -> back-substitution

Complexity: $O(sl^K)$



Notation:
$\mathbf{A}_\times = \mathbf{I} - \widetilde{\mathbf{A}}_1 \otimes \cdots \otimes \widetilde{\mathbf{A}}_K$

Algorithm: SyTE-Fast-P (cont'd)

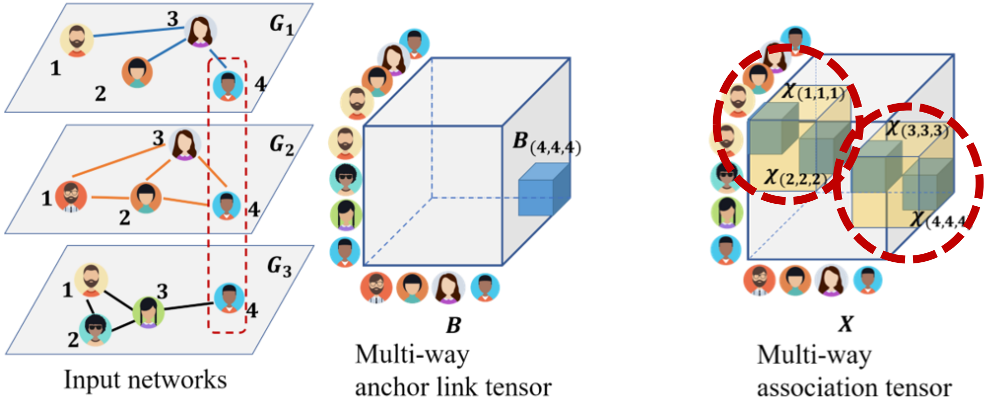
- Solution:
  - Solve the small-scaled system $(\otimes_{i=1}^{K} \mathbf{I}_{l_i+1,l_i} - \otimes_{i=1}^{K} \widetilde{\mathbf{H}}_i)\mathbf{y} = \otimes_{i=1}^{K} \mathbf{U}_{l_i+1}^{T} \mathbf{r}_0$
  - For all subsystems $\left(\mathbf{I} - \widetilde{\mathbf{A}}_1 \otimes \cdots \otimes \widetilde{\mathbf{A}}_K\right)\mathbf{x}_i = \otimes_{j=1}^{K} \mathbf{b}_{\mathrm{j}}^{(i)}$
  - $\mathbf{x} = \mathbf{x}_1 + \cdots \mathbf{x}_K$, where $\mathbf{x}_i = \otimes_{j=1}^{K} \mathbf{U}_{l_j}^{(i)} \mathbf{y}_i$
- Complexity:
  - Time: $O(sKlm + sl^K)$
  - Space: $O(Km + l^{2K} + Kln)$
- Observation:
  - Significantly smaller complexity
  - Linear w.r.t. the number of nodes/edges in each input network

Recall:
- Traditional Krylov subspace: $O(m^K)$
- Solution space bottleneck: $O(n^K)$

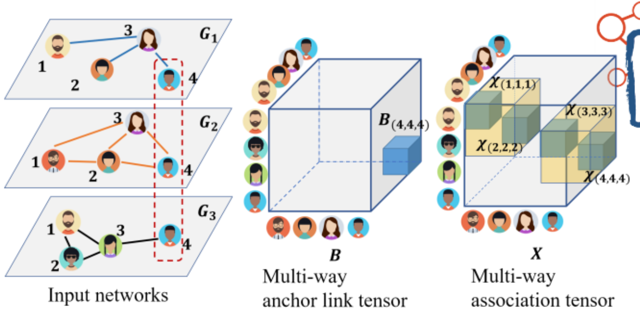
Typical magnitude:
- n, m: 10,000+
- l: 10
- K: 4

Key Ideas: Attributed Networks

- Decompose the equation by node attributes
  - The solution tensor has a block-diagonal structure
- Solve the diagonal tensors by block coordinate descent (BCD)
  - For diagonal block variables
- Adopt approximation in BCD for faster computation
  - Faster computation



Algorithm: SyTE-Fast-A



- Original equation:
  - $\boldsymbol{\mathcal{X}} - \alpha \sum_{o,p,q} \boldsymbol{\mathcal{X}} \times_1 \widetilde{\mathbf{A}}_K^{(o,p,q)} \times_2 \cdots \times_K \widetilde{\mathbf{A}}_1^{(o,p,q)} - (1-\alpha)\boldsymbol{\mathcal{B}} = \mathbf{0}$
- Decomposition:
  - $\boldsymbol{\mathcal{X}}^{i\ldots i} - \sum_{j=1} \boldsymbol{\mathcal{X}}^{j\ldots j} \times_1 \widetilde{\mathbf{A}}_1^{ij} \ldots \times_K \widetilde{\mathbf{A}}_K^{ij} = \boldsymbol{\mathcal{B}}^{i\ldots i}$ (for each node attribute *i*)
- Example:
  - $\boldsymbol{\mathcal{X}}^{1,1,1} - [\boldsymbol{\mathcal{X}}^{1,1,1} \times_1 \tilde{A}_1^{11} \cdots \times_3 \tilde{A}_3^{11} + \underbrace{\boldsymbol{\mathcal{X}}^{2,2,2} \times_1 \tilde{A}_1^{12} \cdots \times_3 \tilde{A}_3^{12}}_{C_1}] = \boldsymbol{\mathcal{B}}^{1,1,1}$
  - $\boldsymbol{\mathcal{X}}^{2,2,2} - [\boldsymbol{\mathcal{X}}^{2,2,2} \times_1 \tilde{A}_1^{22} \cdots \times_3 \tilde{A}_3^{22} + \underbrace{\boldsymbol{\mathcal{X}}^{1,1,1} \times_1 \tilde{A}_1^{21} \cdots \times_3 \tilde{A}_3^{21}}_{C_2}] = \boldsymbol{\mathcal{B}}^{2,2,2}$
  - C_1, C_2: the adjusted anchor multi-way association for each subsystem
  - Approximation: drop C_1, C_2 for solving each subsystem separately

Algorithm: SyTE-Fast-A (cont'd)

- Solution:
  - Construct blocks $\widetilde{\mathbf{A}}_1^{ii}, \dots, \widetilde{\mathbf{A}}_K^{ii}$.
  - Solve $\boldsymbol{\mathcal{X}}^{i\dots i} - \boldsymbol{\mathcal{X}}^{i\dots i} \times_1 \widetilde{\mathbf{A}}_1^{ii} \dots \times_K \widetilde{\mathbf{A}}_K^{ii} = \boldsymbol{\mathcal{B}}^{i\dots i}$ with SyTE-Fast-P.
  - Obtain implicit solution for each diagonal blocks.
- Complexity:
  - Time: $O((K + c_1)m + n + c_2 l^K)$
  - Space: $O(PKm + Kln + l^{2K})$
- Observations:
  - Significantly smaller complexity
  - Linear w.r.t. the number of nodes/edges in each input network

Recall:
- Traditional Krylov subspace: $O(m^K)$
- Solution space bottleneck: $O(n^K)$

Typical magnitude:
- n, m: 10,000+
- l: 10
- K: 4

Roadmap

- Motivation ✓
- Problem Definition ✓
- Formulation ✓
- Proposed Algorithm ✓
- Experimental Results ←
- Conclusion

Experimental Settings

- Datasets:

Dataset Name	Category	# of Nodes	# of Edges
Arxiv	Academic network	2,908	3,551
DBLP	Co-authorship	1,013	3,244
Douban	User relationship	3,384	6,556
Aminer	Academic network	1,274,360	4,756,194

Dataset Name	# of Users	# of Artists	# of Tags
LastFm	15,154	2,982	4,144

- Evaluation tasks for effectiveness:
  - T1. Multi-network alignment (one-to-one)
  - T2. Multi-network node retrieval
  - T3. High-order recommendation

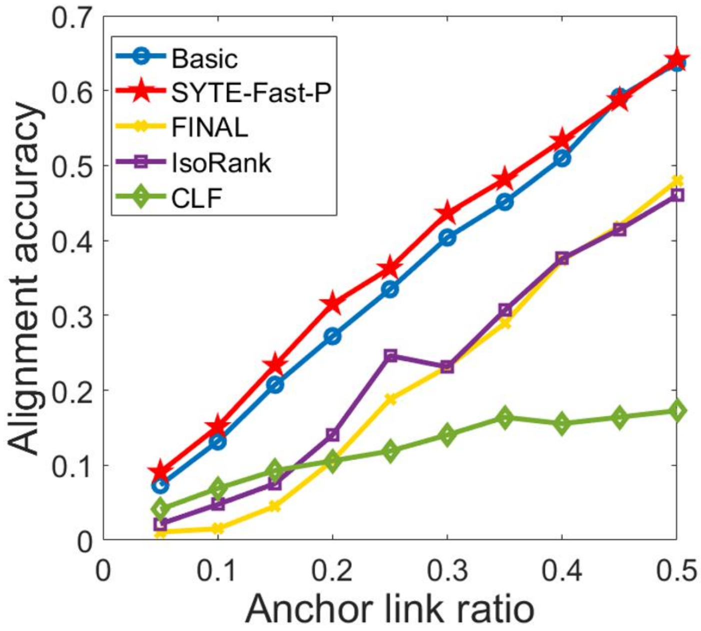
Experimental Settings (cont'd)

- Existing baseline methods:
  - T1. Multi-network alignment (one-to-one):
    - *CLF, FINAL, IsoRank*
  - T2. Multi-network node retrieval:
    - *REGAL, CrossMNA, FINAL, IsoRank*
  - T3. High-order recommendation:
    - *nNTF (non-negative tensor factorization), NTF (Neural Tensor Factorization), wiZAN-Dual*
  - Scalability:
    - *FP (Fixed Point method) and CG (Conjugate Gradient method)*
- Proposed baseline methods:
  - For plain networks:
    - *STYE-Fast-P*, Basic algorithm*
  - For attributed networks:
    - *SYTE-Fast-A*, Basic algorithm, SYTE-BCD*

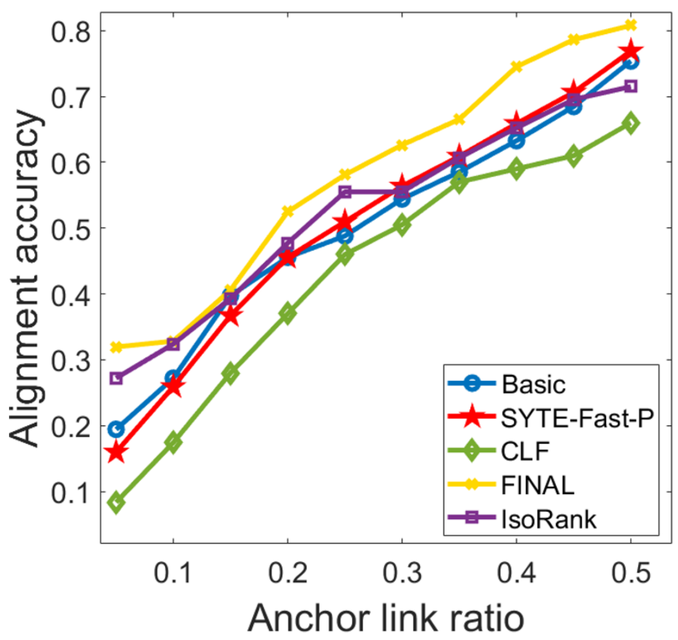
T1. Multi-network Alignment

- On plain networks:

By high-order metric



By pairwise metric

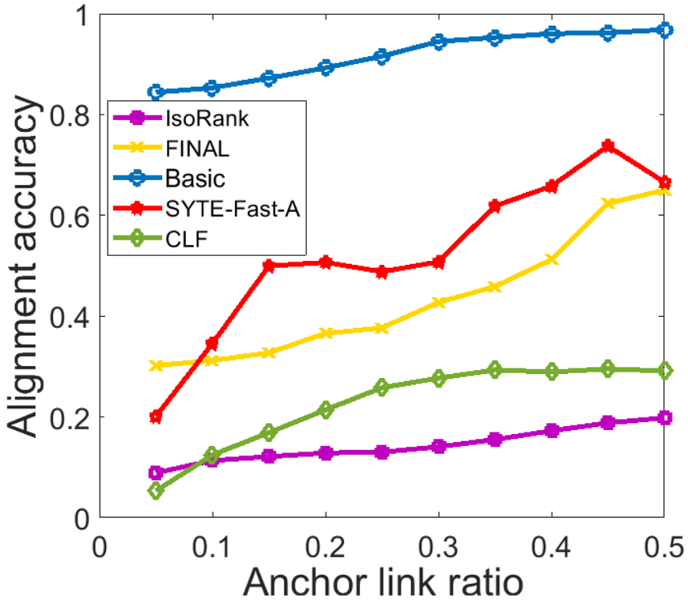


- Observations:
  - Both basic algorithm and SyTE-Fast-P outperform baseline methods with high-order metric.

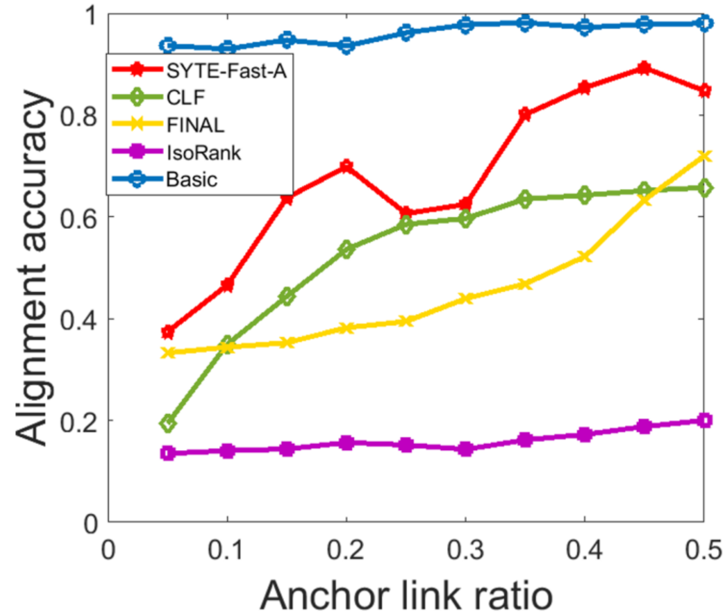
T1. Multi-network Alignment (cont'd)

- On attributed networks:

By high-order metric



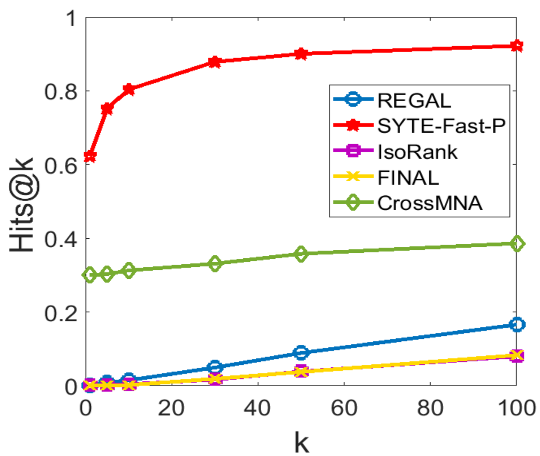
By pairwise metric

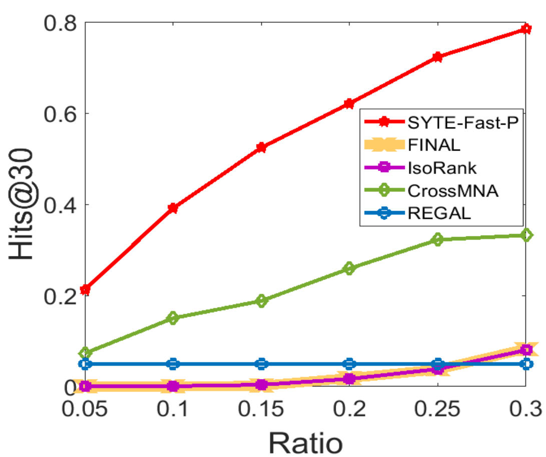


- Observations:
  - Both basic algorithm and SyTE-Fast-A outperform baseline methods.
  - Performance drop compared with basic method

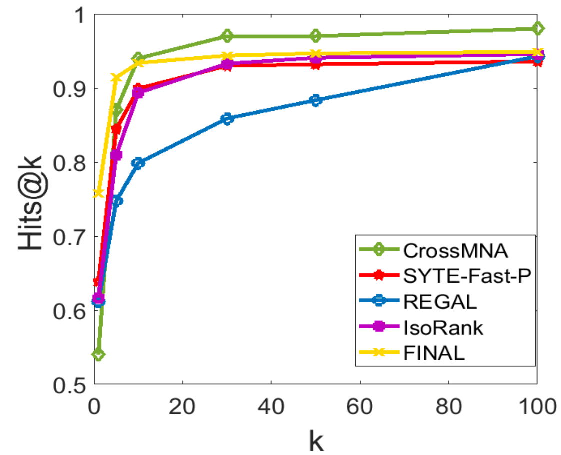
T2. Multi-network Node Retrieval

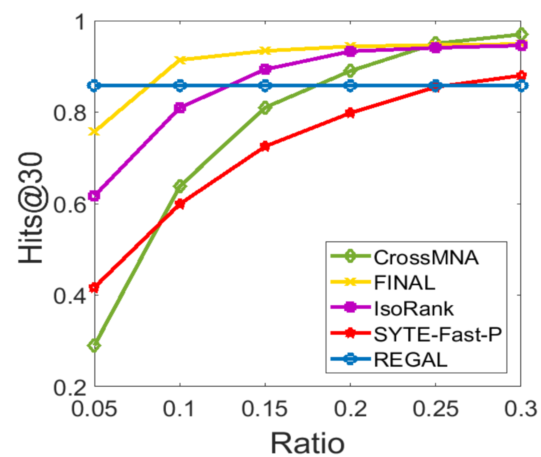
By high-order metric





By pairwise metric

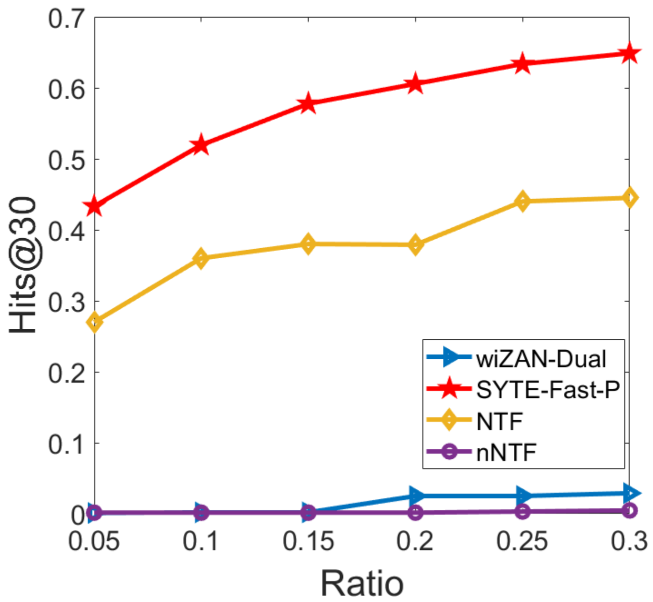




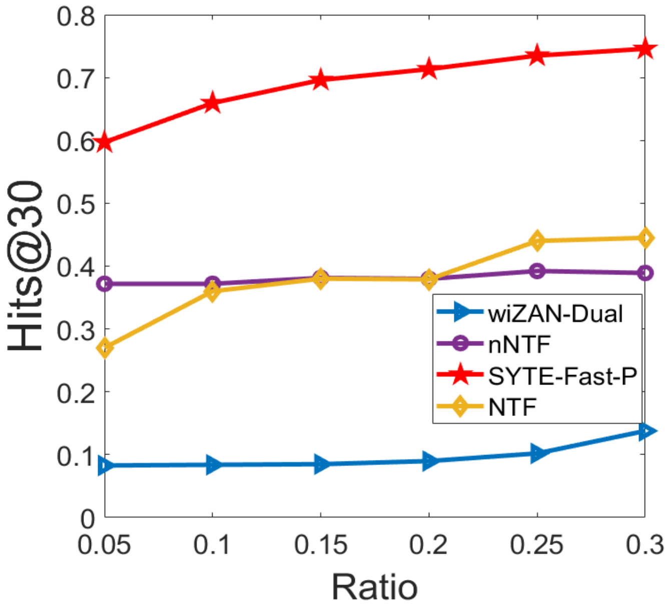
T3. High-order Recommendation

- Hits@30 vs. ratio of known recommendation:

By high-order metric



By pairwise metric

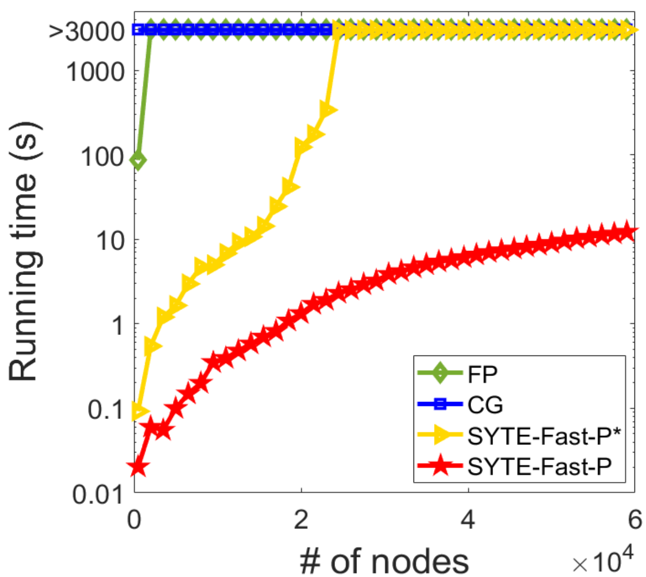


- Observations:
  - SyTE-Fast-P outperforms baselines in terms of both high-order and pair-wise metrics.

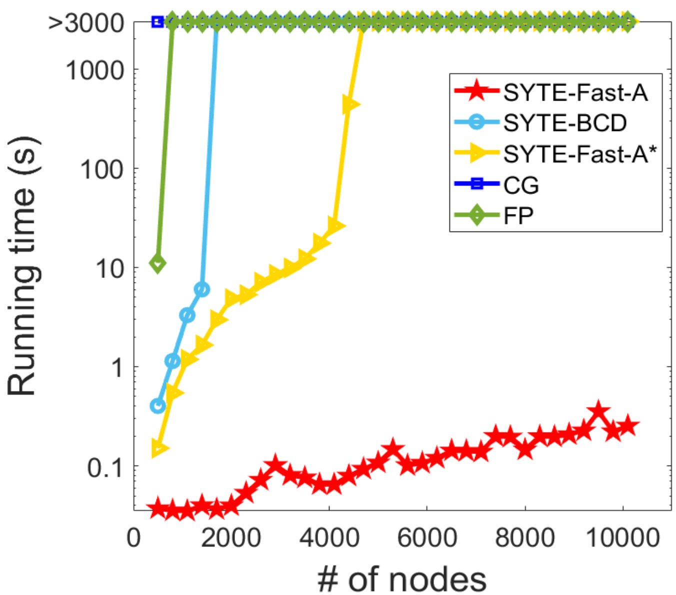
Scalability

- Runtime vs. # of nodes in each network:

On plain networks



On attributed networks

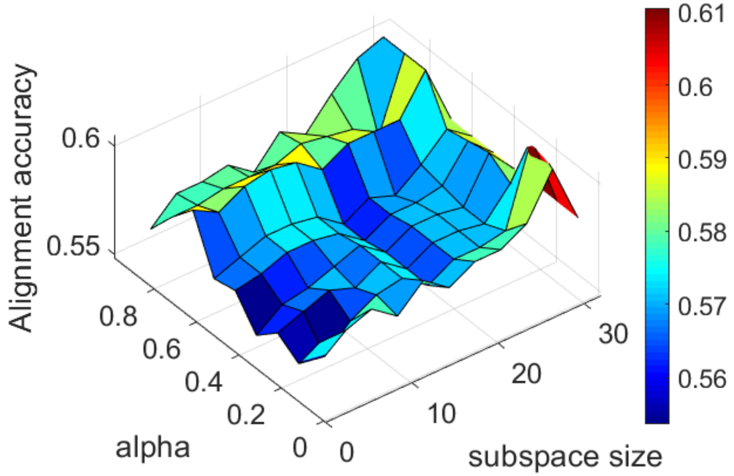


- Observations:
  - SyTE-Fast-P/A exhibits a linear scalability w.r.t. the # of nodes of the input networks

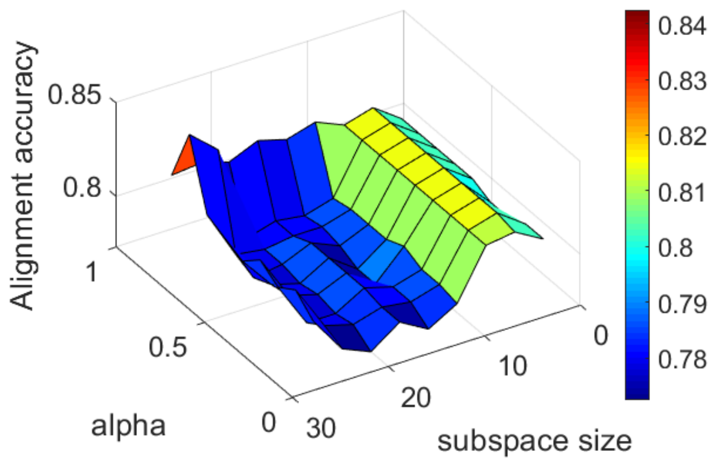
Parameter Sensitivity

- Multi-network alignment accuracy vs. model parameters:

By high-order metric



By pairwise metric

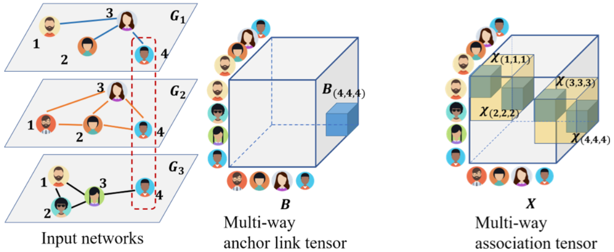


- Observations:
  - Stable in a relatively large range of parameter space.

Roadmap

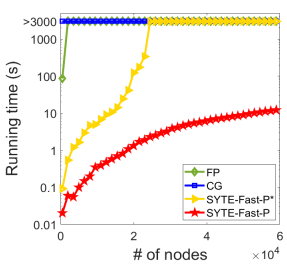
- Motivation ✓
- Problem Definition ✓
- Formulation ✓
- Proposed Algorithm ✓
- Experimental Results ✓
- Conclusion ←

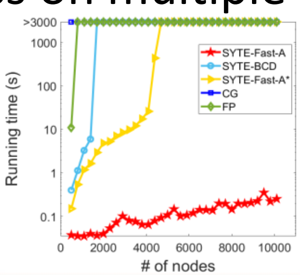
Conclusion

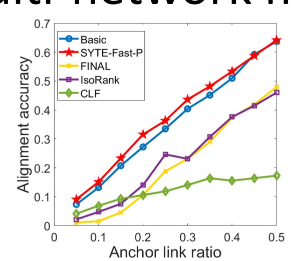


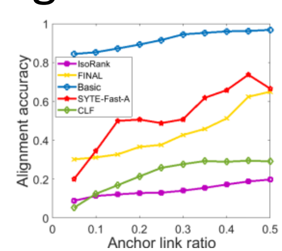
- Goal:
  - Fast algorithms for multi-way association inference
- Contribution:
  - Optimization formulation solved by Sylvester tensor equation
  - Fast algorithm on plain/attributed networks
  - Theoretical analysis on model optimality, sensitivity
- Evaluation results:
  - *Linear* scalability w.r.t the input graph size
  - Significant speedup against traditional methods
  - Effectiveness on multiple multi-network mining tasks

More in the paper:
- Model variants
- Sensitivity analysis
- Additional experiments









…

Thank you!

Q&A

- *Code:* https://github.com/boxindu/SYTE
- *Contact:* *boxindu2@illinois.edu*
  http://boxindu2.web.illinois.edu/