Scalable tensor methods for multi-relational learning across graphs

Zhuliu Li

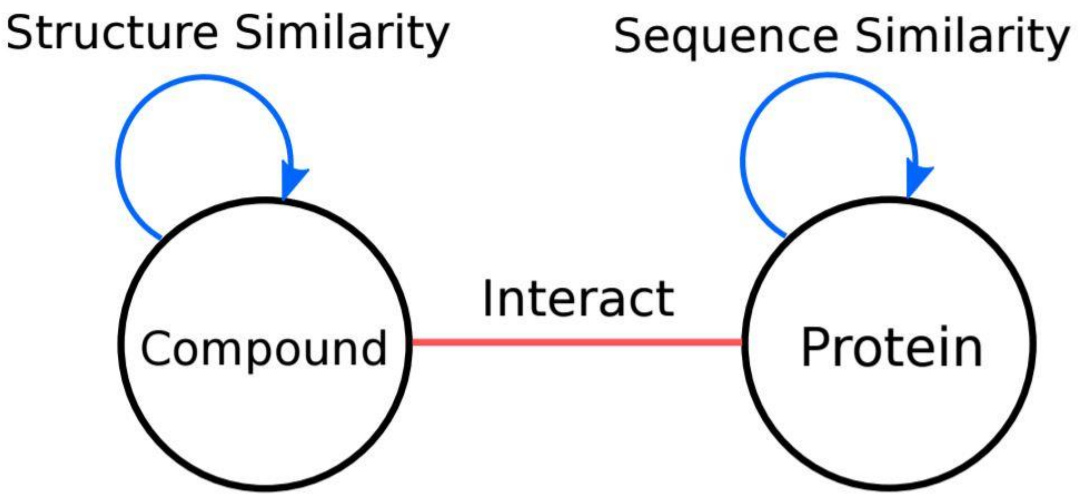
Department of Computer Science and Engineering

Outline

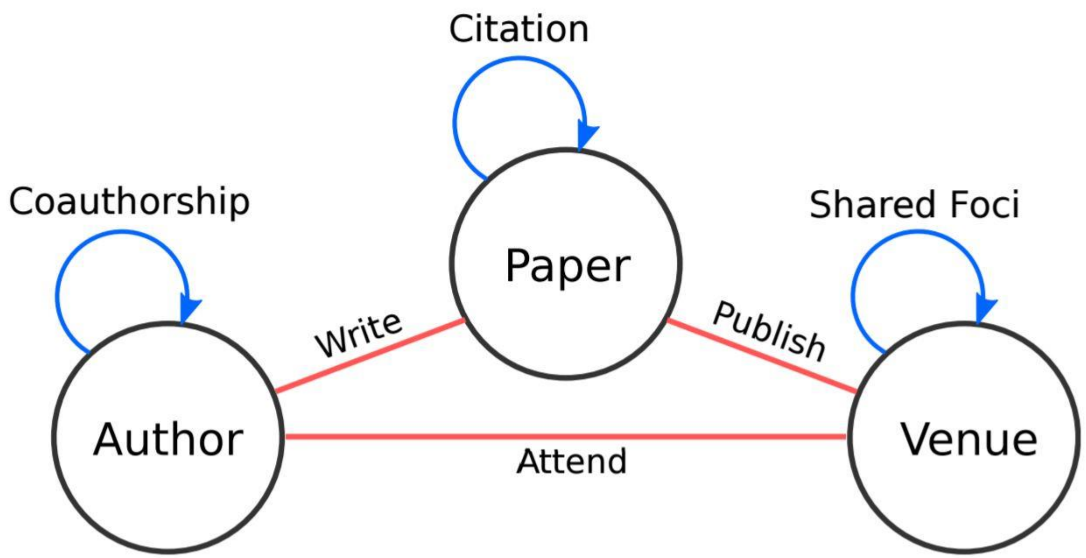
- Task description: multi-relational learning among heterogeneous graphs
    - Examples
    - Formal definition
- Review: label propagation and tensor product graph
    - Label propagation algorithm
    - Tensor product graph
- Scalable tensor methods for multi-relational learning
    - Low-rank label propagation algorithm on tensor product graph
    - Sparse label propagation algorithm on tensor product graph
    - Tensor decomposition with graph constraint
- Experiments: Aligning multiple PPI networks, CT scans and DBLP data

Task description: examples

Liu, H., & Yang, Y. (ICML, 2016)



Enzyme data



DBLP data

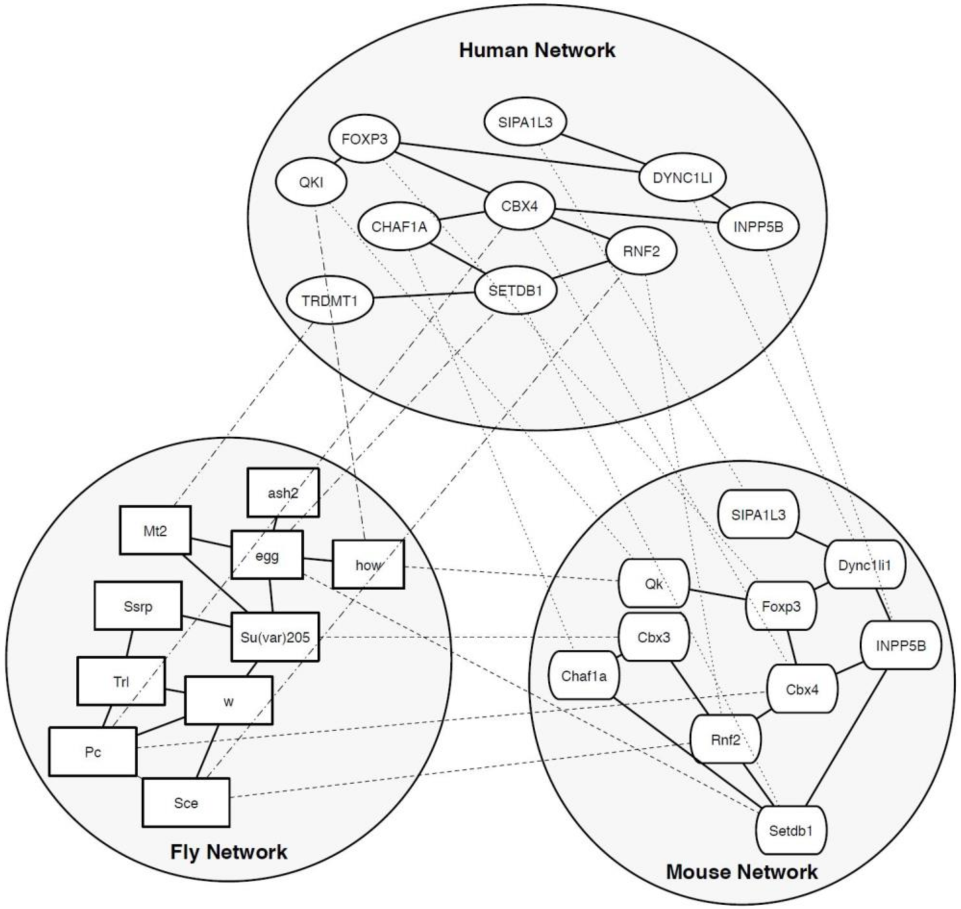
- Blue edges: within graph interactions
- Red edges: cross graph interactions

"Chemical compound (drug) A is targeting on protein B." (Compound: A, Protein: B)

"John publish a reinforcement learning paper at ICML." (Author: John, Paper: RL, Venue: ICML)

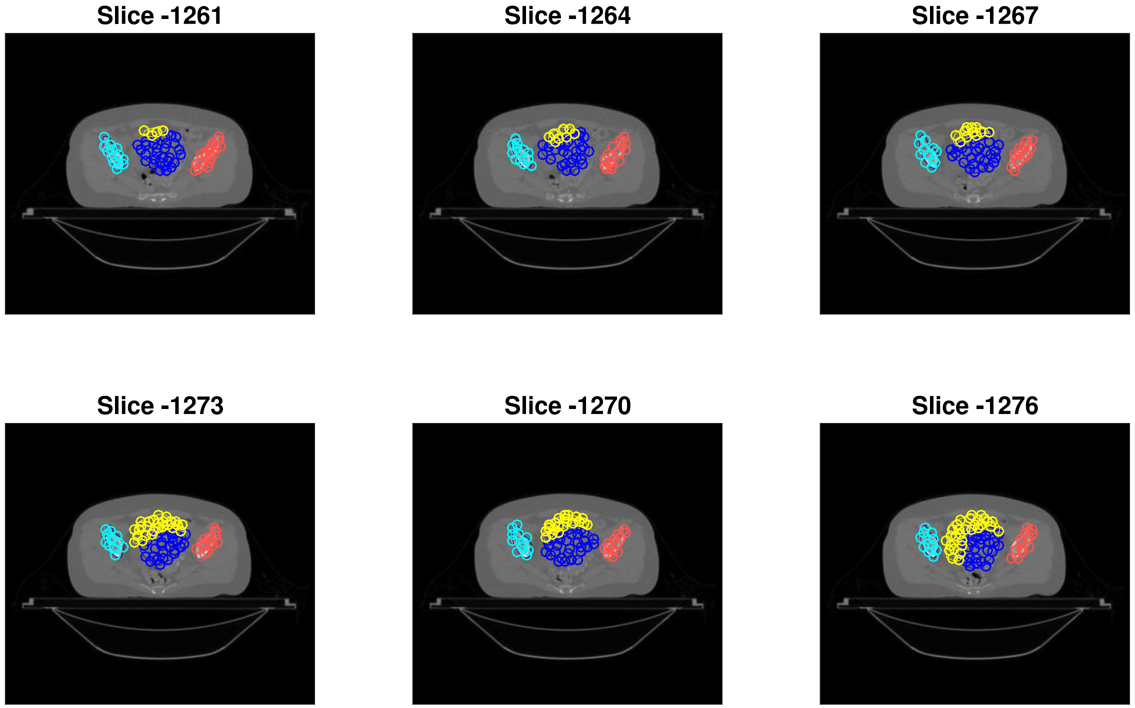
Task description: examples

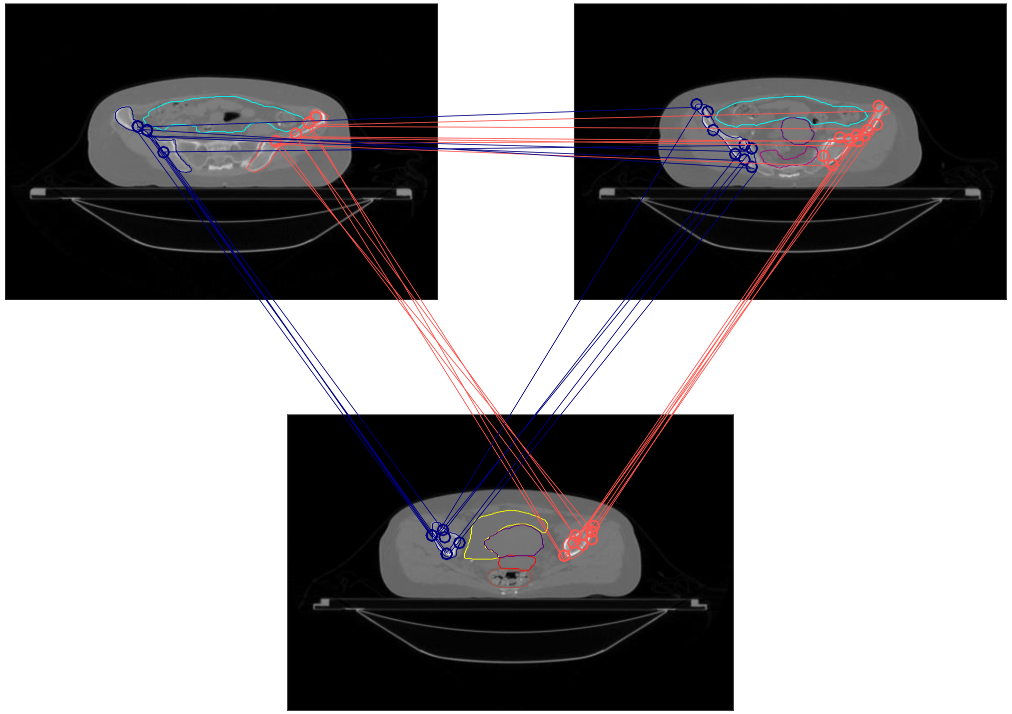
- Aligning protein-protein interaction networks across species.
- Nodes are proteins
- Edges connect interacting proteins.
- The relations are evolutionary relations.



Task description: examples

- Align the same organ (color) across the CT scans of human body

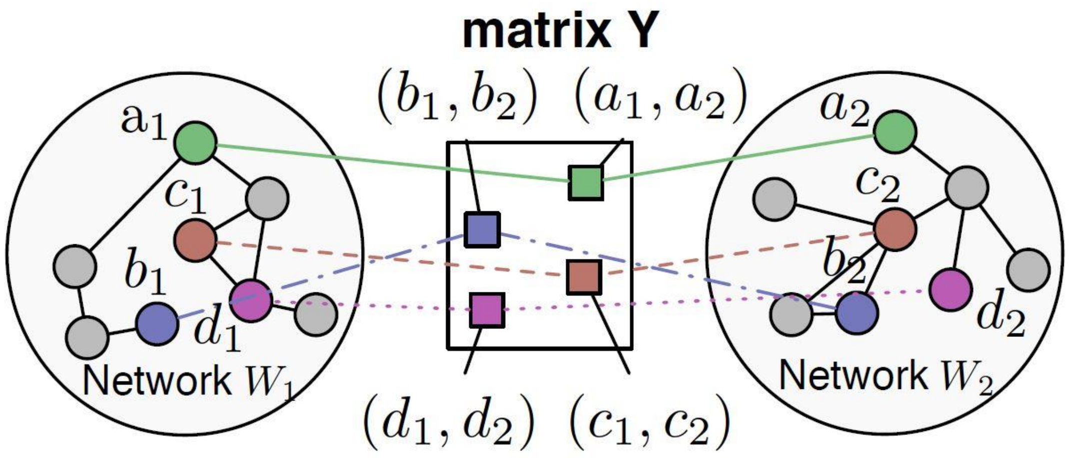


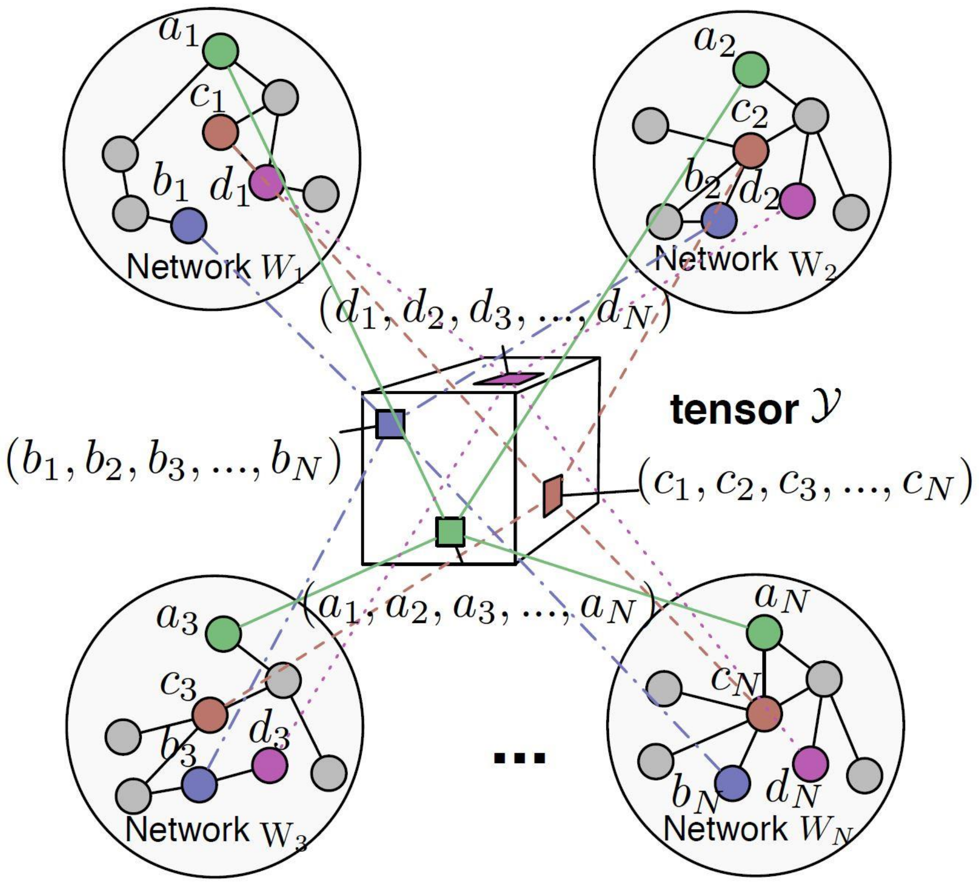


Task description: definition

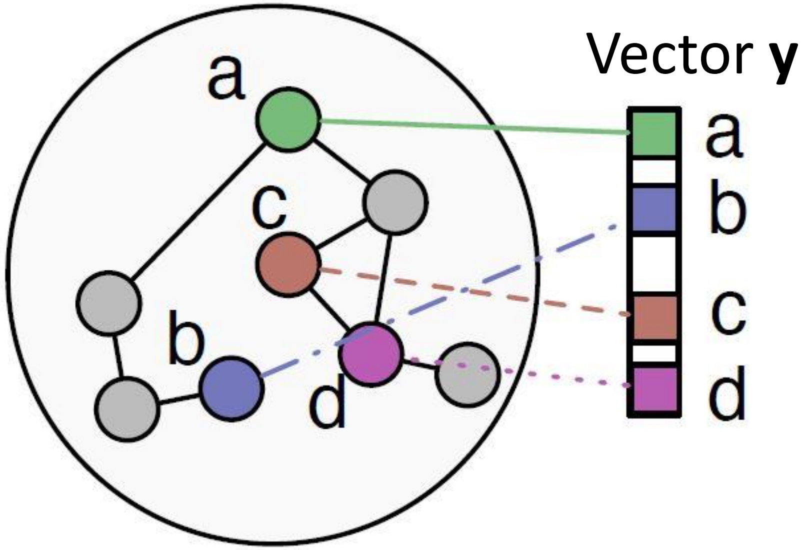
- Given:
  - Individual graph $W^{(i)} \in \mathbb{R}^{I_i \times I_i}, i = 1, \ldots n$.
  - A tensor $\mathcal{Y} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_n}$ storing (parts of) the initial scores of the multi-relations (tuples) between the nodes from different graphs.
- Task:
  - Predict the scores of the unscored tuples.
  - Correct the scores of the scored tuples based on the graph structures.

Task description: definition



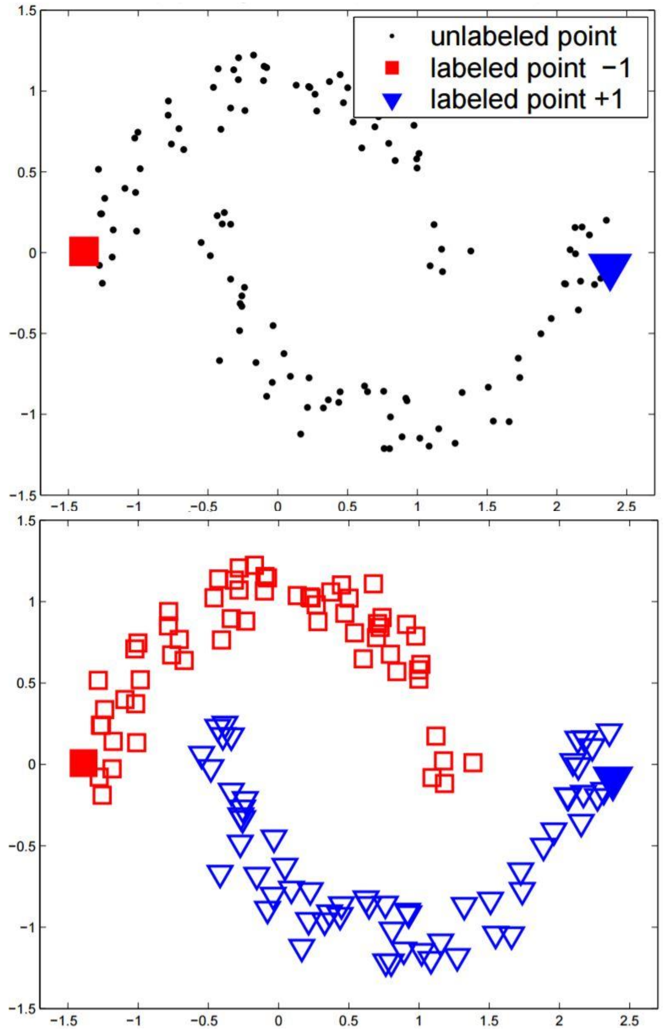


Background: label propagation



$$y^{t+1} = \alpha S y^t + (1-\alpha) y^0$$

$$S = D^{-\frac{1}{2}} W D^{-\frac{1}{2}}$$



Zhou, Denny, et al. (NIPS, 2004)

Background: label propagation

- Objective function

$$\mathcal{J}(y) = \sum_{i,j} w_{ij}\left(\frac{y_i}{\sqrt{d_{ii}}} - \frac{y_j}{\sqrt{d_{jj}}}\right)^2 + \mu||y - y^0||^2$$

$$= y^T L y + \mu||y - y^0||^2,$$

where $\mu = \frac{1-\alpha}{\alpha}$, $L = I - S$.

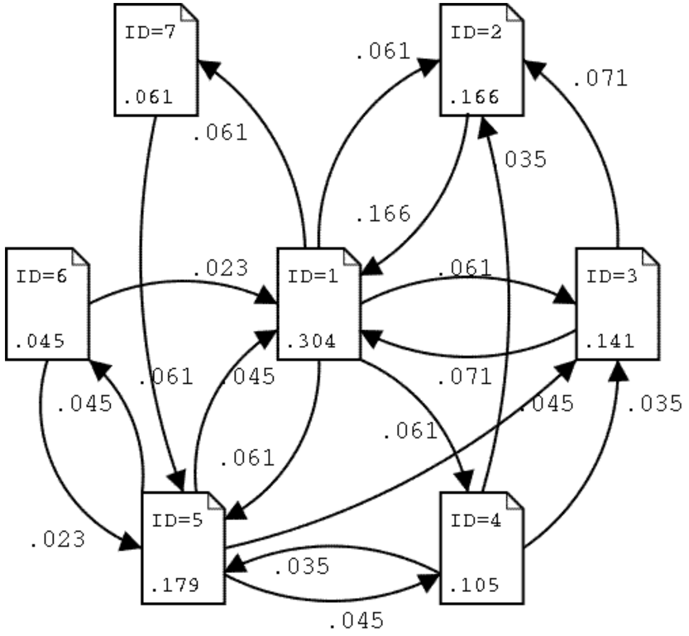
- Closed-form solution

$$y^* = (1-\alpha)(I - \alpha S)^{-1} y^0$$

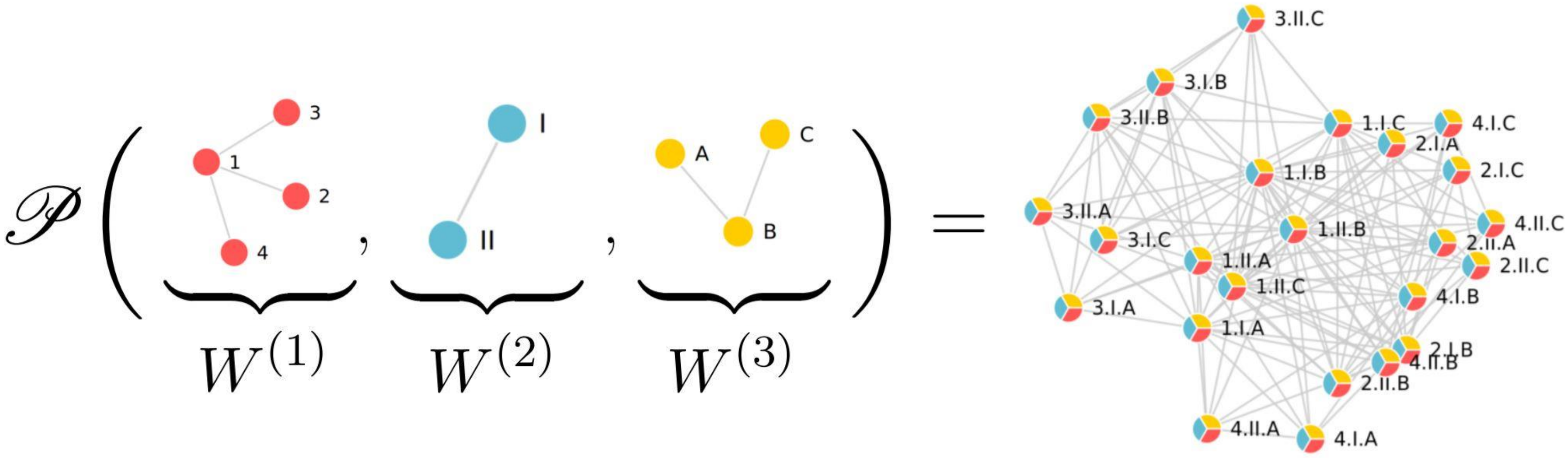
PageRank (random walk with restart)

- Define a Markov Chain to represent the web surfing jumping probability
- Start with a random page and take random walks.
- The stationary distribution of the Markov Chain gives the probability of stopping at a particular page (a good rank!)
- Label propagation is a very similar algorithm.

$$\mathbf{R} = \begin{bmatrix} (1-d)/N \\ (1-d)/N \\ \vdots \\ (1-d)/N \end{bmatrix} + d \begin{bmatrix} \ell(p_1,p_1) & \ell(p_1,p_2) & \cdots & \ell(p_1,p_N) \\ \ell(p_2,p_1) & \ddots & & \\ \vdots & & \ell(p_i,p_j) & \\ \ell(p_N,p_1) & & & \ell(p_N,p_N) \end{bmatrix} \mathbf{R}$$



Background: tensor product graph (TPG)



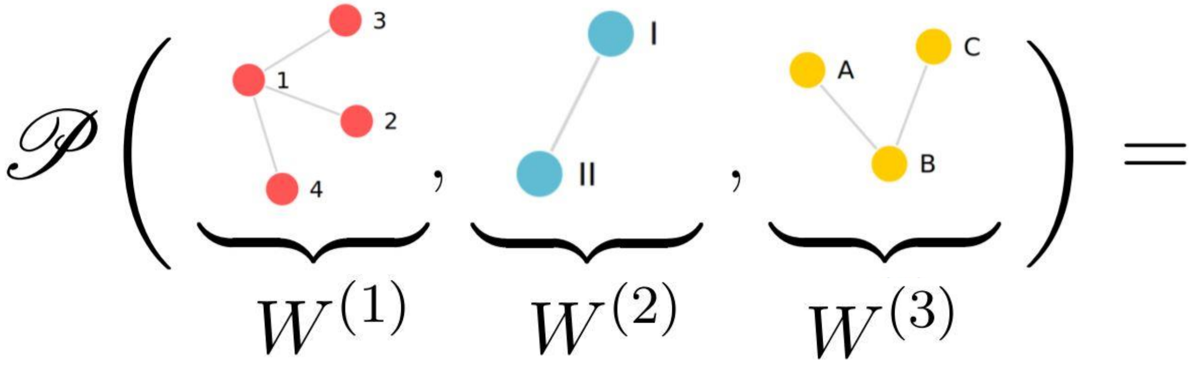
$$W = W^{(1)} \otimes W^{(2)} \otimes W^{(3)} \in \mathbb{R}^{I_1 I_2 I_3 \times I_1 I_2 I_3}$$

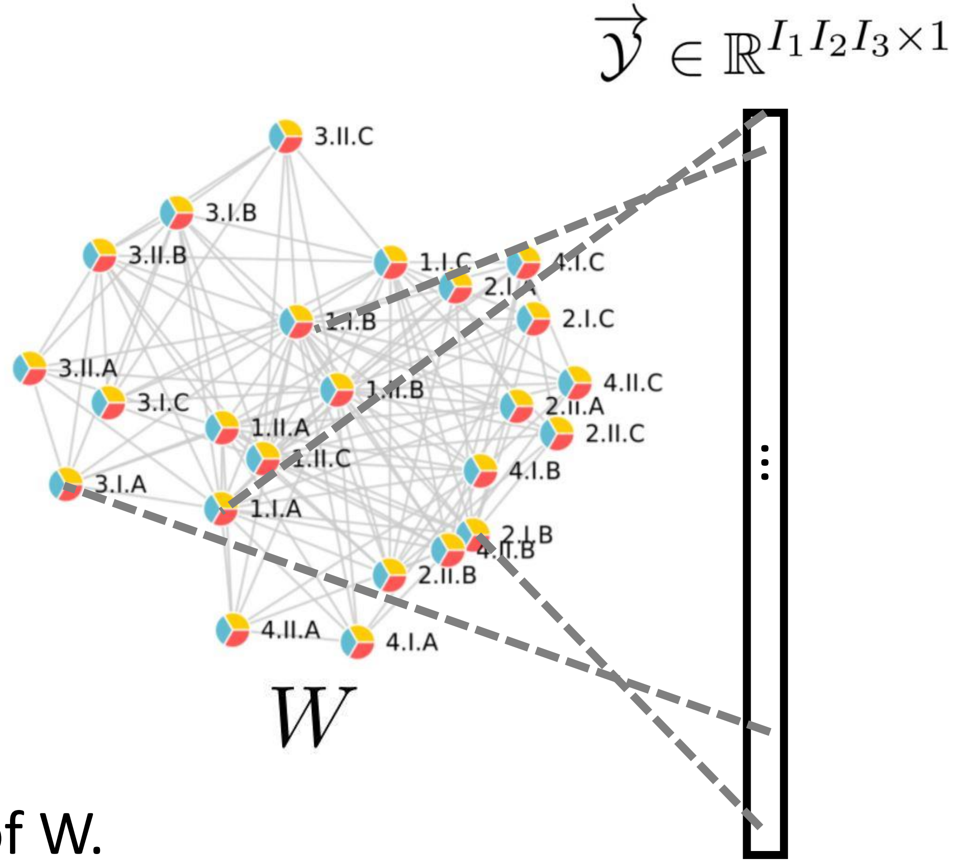
$$W_{(1,I,A),(2,II,B)} = W^{(1)}_{1,2} W^{(2)}_{I,II} W^{(3)}_{A,B}$$

Revisit the learning task

- Given:
  - Individual graph $W^{(i)} \in \mathbb{R}^{I_i \times I_i}, i = 1, \ldots n$.
  - A tensor $\mathcal{Y} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_n}$ storing (parts of) the initial scores of the multi-relations (tuple) between the nodes from different graphs.
- Task:
  - Predict the scores of the unscored tuples.
  - Correct the scores of the scored tuples based on the graph structures.

Revisit the learning task

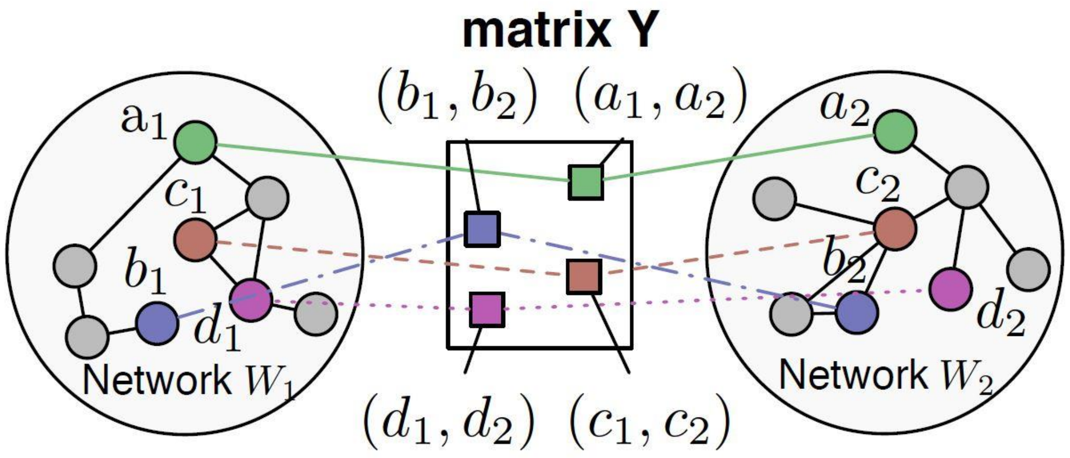




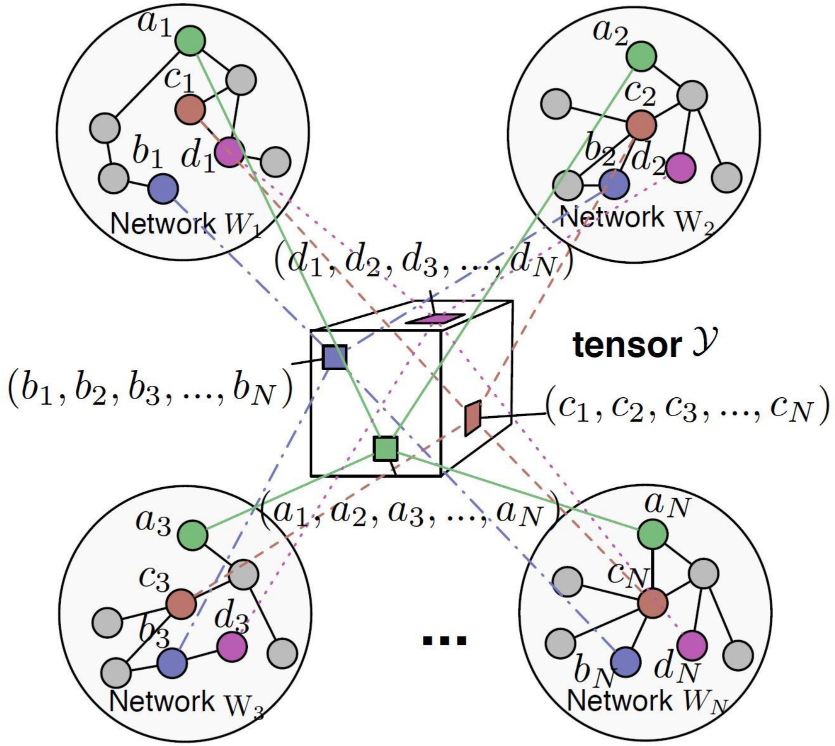
- Task:
  - Predict the scores of the unscored tuples =
  - Predict the labels of the unlabeled nodes of W.

  - Correct the scores of the scored tuples based on the graph structures =
  - Correct the labels of the labeled nodes of W based on the graph structures

Revisit the learning task

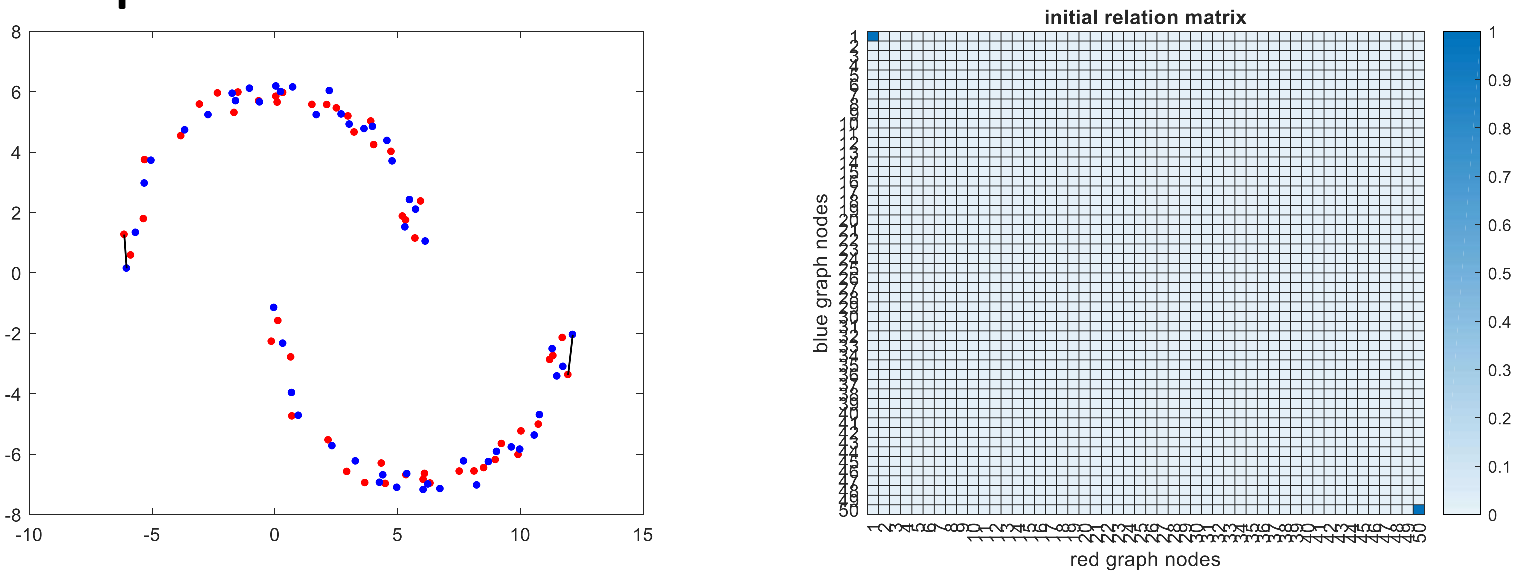


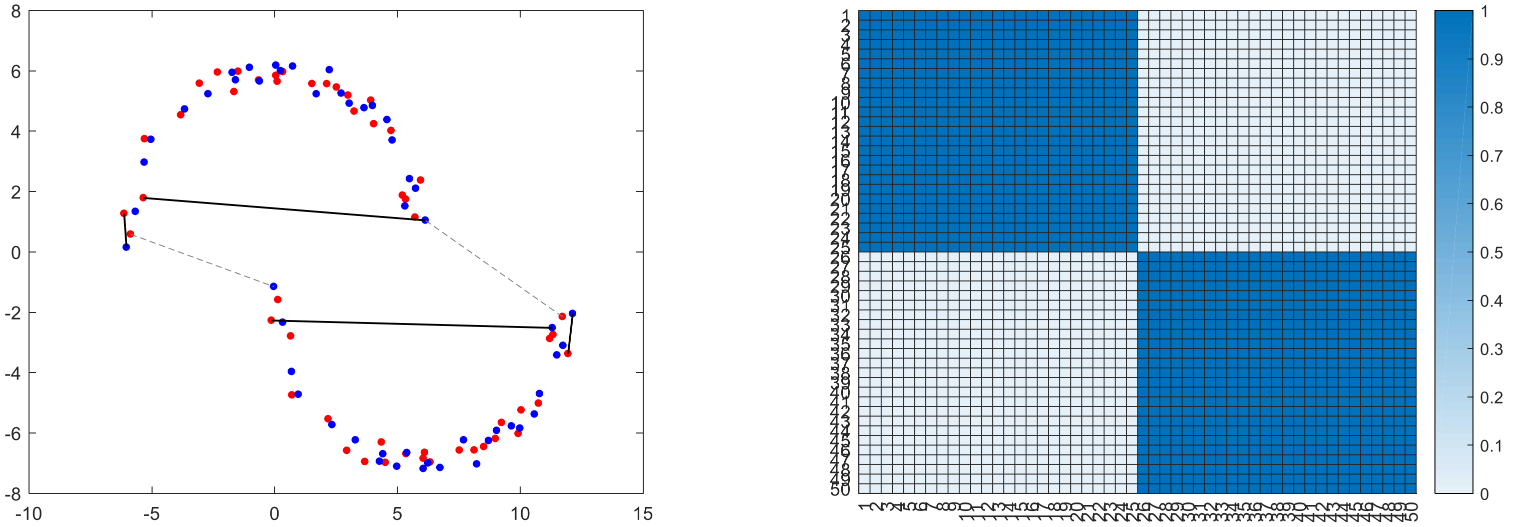
$$\vec{Y}^{t+1} = \alpha(S^{(1)} \otimes S^{(2)})\vec{Y}^{t} + (1-\alpha)\vec{Y}^{0}$$
$$Y^{t+1} = \alpha S^{(2)} Y^{t} S^{(1)} + (1-\alpha) Y^{0}$$



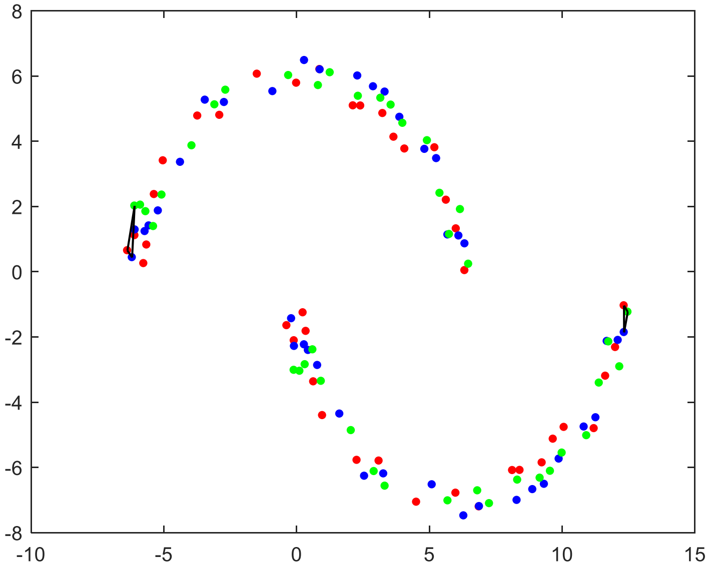
$$\vec{\mathcal{Y}}^{t+1} = \alpha(\otimes_{i=1}^{n} S^{(i)})\vec{\mathcal{Y}}^{t} + (1-\alpha)\vec{\mathcal{Y}}^{0}$$
$$\mathcal{Y}^{t+1} = \alpha \mathcal{Y}^{t} \times_1 S^{(n)} \times_2 S^{(n-1)} \cdots \times_n S^{(1)} + (1-\alpha)\mathcal{Y}^{0}$$

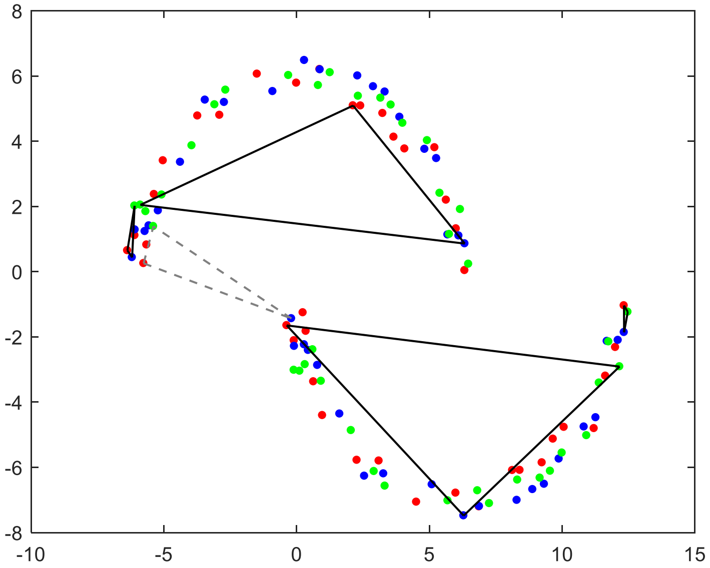
A simple simulation

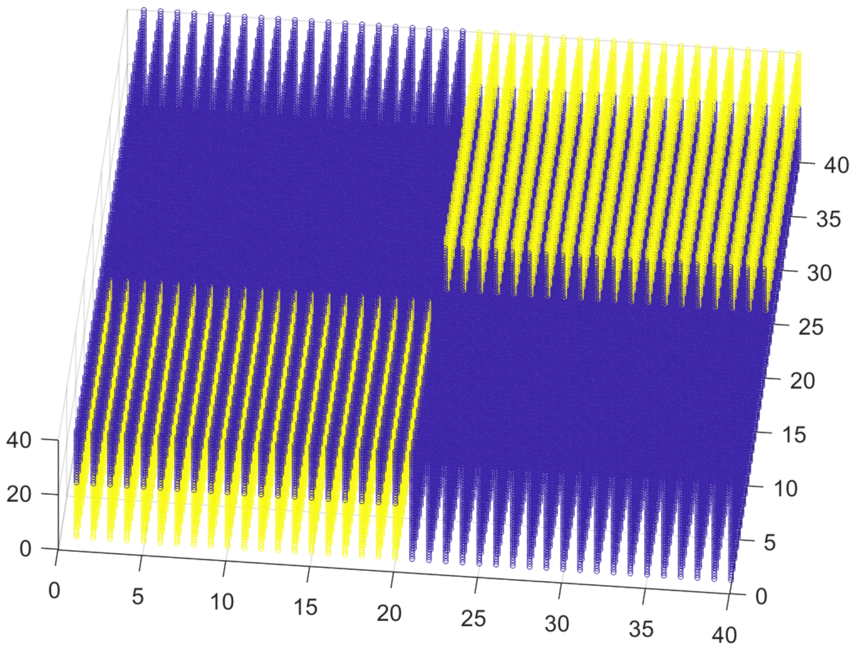




A simple simulation







Scalability issue

- Iterative approach:

$$\vec{\mathcal{Y}}^{t+1} = \alpha(\otimes_{i=1}^{n} S^{(i)})\vec{\mathcal{Y}}^{t} + (1-\alpha)\vec{\mathcal{Y}}^{0}$$

$$\mathcal{Y}^{t+1} = \alpha\mathcal{Y}^{t} \times_1 S^{(n)} \times_2 S^{(n-1)} \cdots \times_n S^{(1)} + (1-\alpha)\mathcal{Y}^{0}$$

  - Time complexity of one iteration $O((\prod_{i=1}^{n} I_i)(\sum_{i=1}^{n} I_i))$
  - Space complexity $O(\prod_{i=1}^{n} I_i)$

- Closed form solution:

$$\begin{aligned}\overrightarrow{\mathcal{Y}}^{*} &= (1-\alpha)(I-\alpha S)^{-1}\overrightarrow{\mathcal{Y}}^{0} \\ &= (1-\alpha)(\otimes_{i=1}^{n} Q^{(i)})(I-\alpha(\otimes_{i=1}^{n}\Lambda^{(i)}))^{-1}(\otimes_{i=1}^{n} Q^{(i)T})\overrightarrow{\mathcal{Y}^{0}}\end{aligned}$$

Proposition 1: Low rank approximation of TPG

- Idea:

$$\underset{S_k}{\text{minimize}} \quad ||(I - \alpha S)^{-1} - (I - \alpha S_k)^{-1}||_{2,F}$$

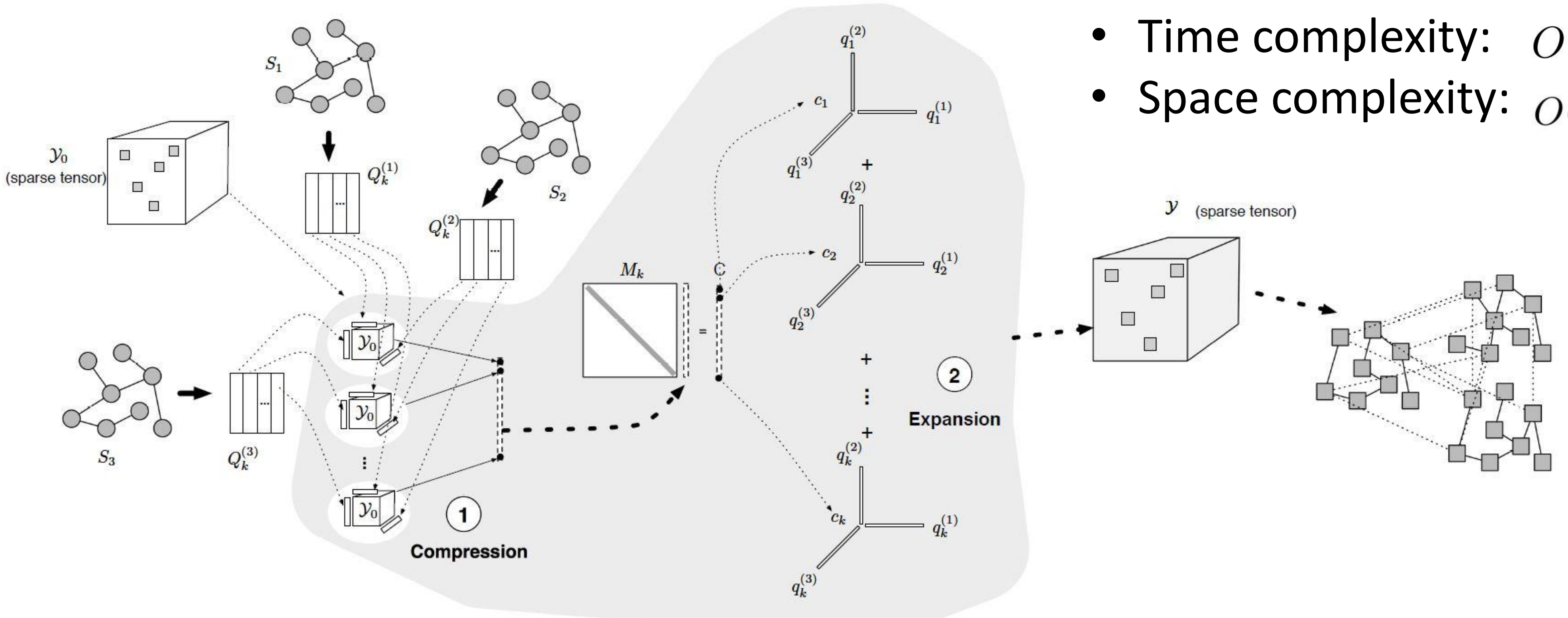
$$\text{subject to} \quad \text{rank}(S_k) = k$$

Solved by eigenvalue selection method

Lemma. *Let $\lambda_1, \ldots, \lambda_n$ be eigenvalues of A with corresponding eigenvectors $x_1, \ldots, x_n$, and let $\mu_1, \ldots, \mu_m$ be eigenvalues of B with corresponding eigenvectors $y_1, \ldots, y_m$. Then the eigenvalues and eigenvectors of $A \otimes B$ are $\lambda_i \mu_j$ and $x_i \otimes y_j$, $i = 1, \ldots, n$, $j = 1, \ldots, m$.*

Proposition 1: Low rank approximation of TPG

- Time complexity: $O(mnk)$
- Space complexity: $O(\max(k\sum_{i=1}^{n} I_i, m))$

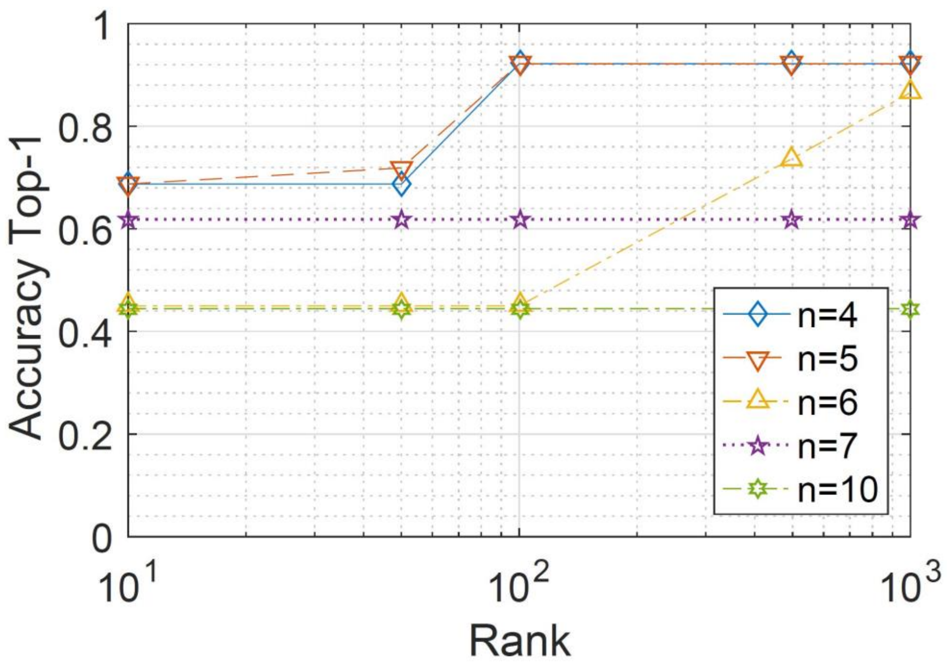


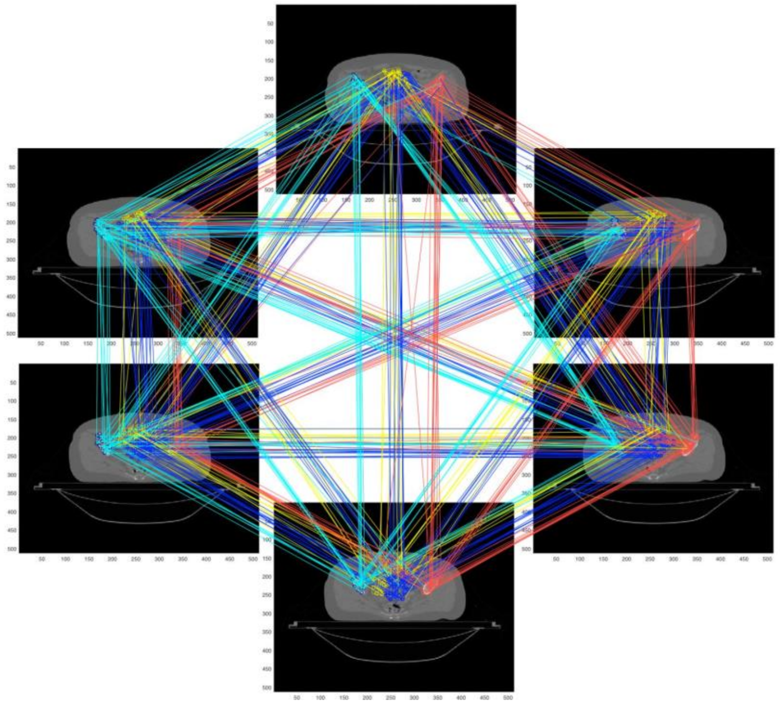
$$\overrightarrow{\mathcal{Y}^*} = (1-\alpha)(Q^{(1)}_{\text{select}} \odot \cdots \odot Q^{(n)}_{\text{select}}) M (Q^{(1)}_{\text{select}} \odot \cdots \odot Q^{(n)}_{\text{select}})^T \overrightarrow{\mathcal{Y}^0} + (1-\alpha)\overrightarrow{\mathcal{Y}^0}$$

Expansion — Compression

Experiment 1 – CT Scan Image Alignment

- 134 CT Scan images of size 512 x 512.
- Each image was segmented into regions (features).
- Sampled spots are aligned across the images.





Proposition 2: keep the tensor sparse

- Tensor becomes dense after every propagation step
- Adding L1-norm regularizer to keep the sparsity

$$\mathcal{J}_{L1}(\overrightarrow{\mathcal{Y}}) = \mathcal{J}(\overrightarrow{\mathcal{Y}}) + \beta \sum_{i=1}^{N} |\overrightarrow{\mathcal{Y}}_i|.$$

- Apply FISTA (fast iterative shrinkage-thresholding algorithm)

$$\mathcal{Y}^{t+1} = \frac{1}{L_J}(\mathcal{Y}^t \times_1 S^{(n)} \times_2 S^{(n-1)} \cdots \times_n S^{(1)}) + (1 - \frac{1+\mu}{L_J})\mathcal{Y}^t + \frac{\mu}{L_J}\mathcal{Y}^0.$$

where the step size $L_J = 1 + \mu - \min(\otimes_{i=1}^{n}[\lambda_{\min}(S^{(i)}), \lambda_{\max}(S^{(i)})])$

- Use METTM (Memory-Efficient Tensor Times Matrix) for matrix-tensor multiplication

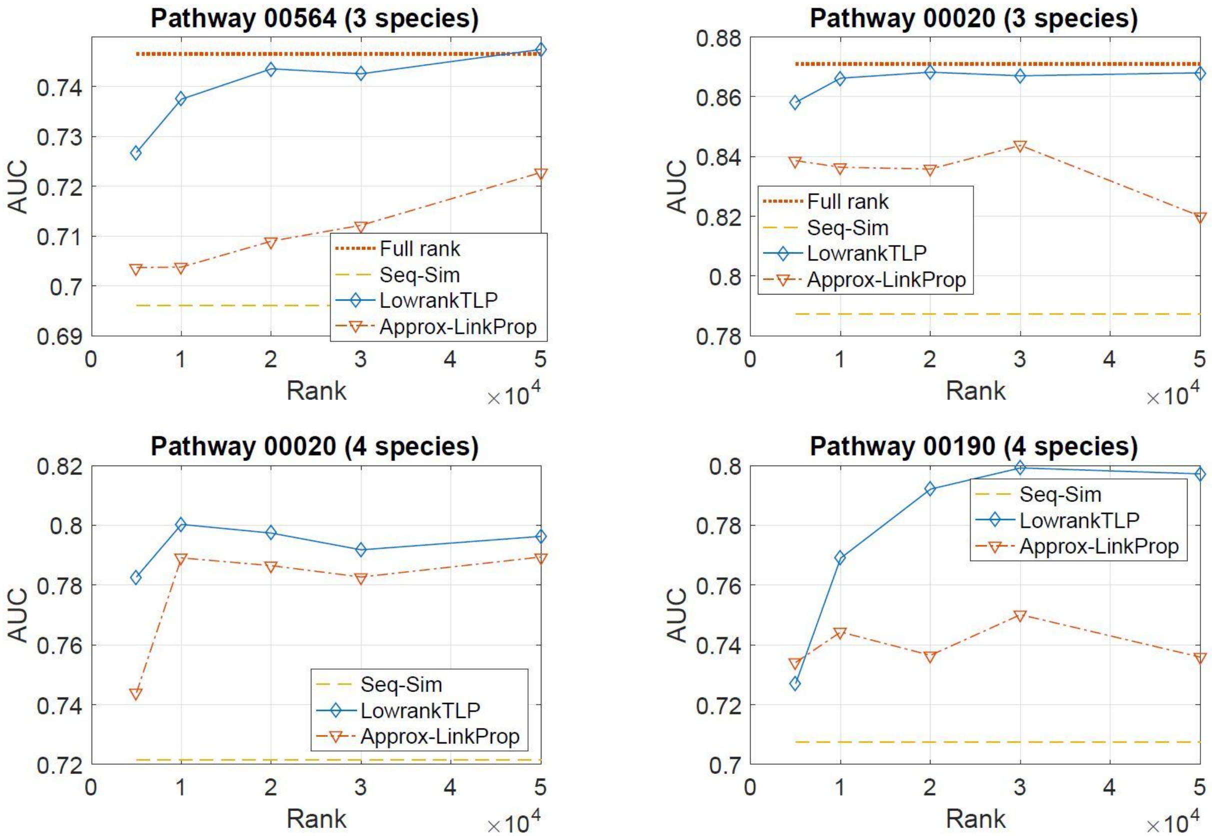
Experiment 2 – PPI Network Alignment

- PPI subnetworks of four species Human (HSA), Mouse (MMU), Fly (DME) and Yeast (SCE).
- Four pathways (subnetworks) are tested.

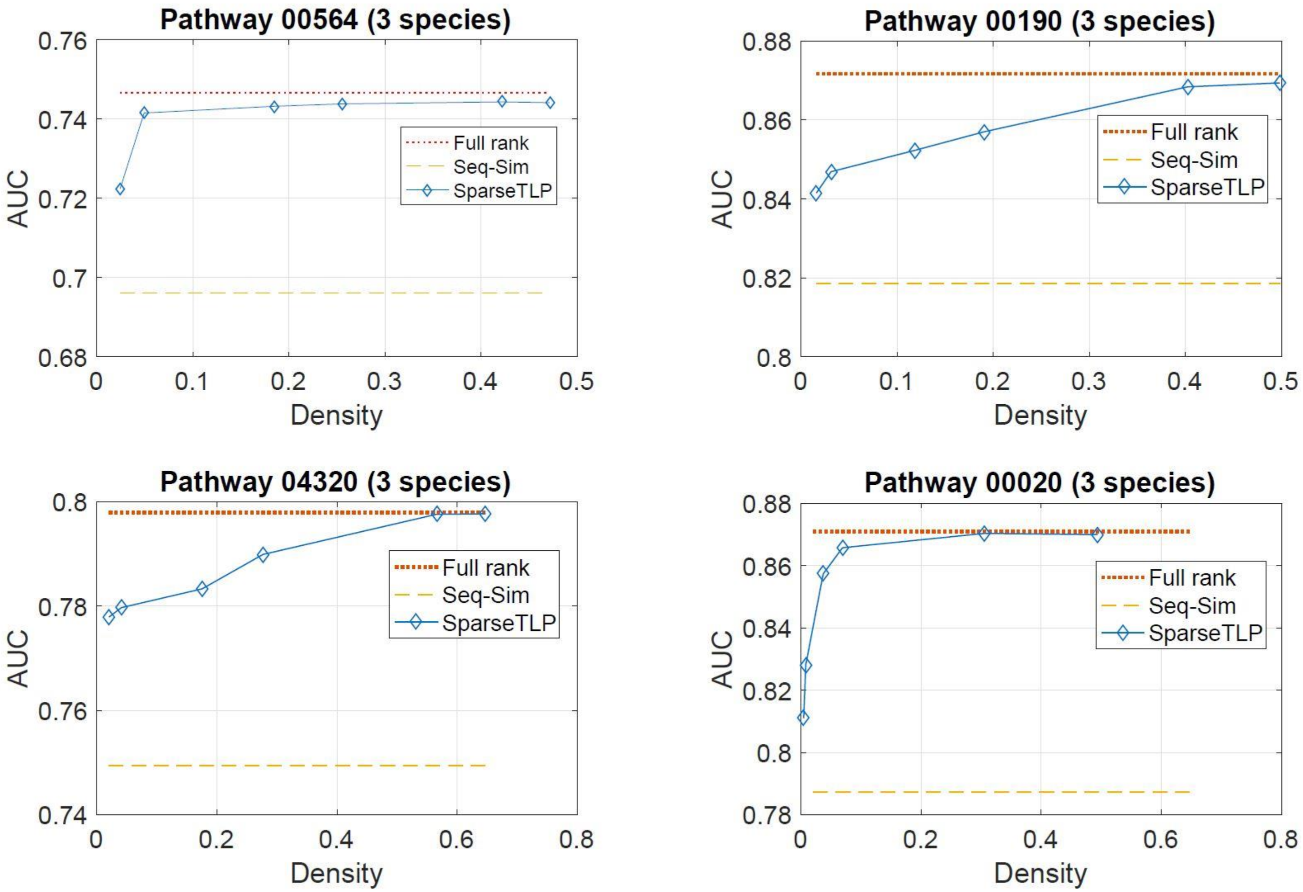
Pathway	DME	HSA	MMU	SCE
00020	43/332	30/257	32/312	32/265
00190	144/208	133/292	134/315	72/284
00564	63/151	95/318	94/187	18/192
04320	28/226	28/300	26/228	-

- **Validation**:
  Checking the agreement of the alignment scores and the protein functional similarities returned by Gene Ontology *(Consortium et al., 2015)*

Experiment 2 – PPI Network Alignment



Experiment 2 – PPI Network Alignment



Future work: tensor decomposition

- Revisit objective function

$$\mathcal{J}(y) = \sum_{i,j} w_{ij}\left(\frac{y_i}{\sqrt{d_{ii}}} - \frac{y_j}{\sqrt{d_{jj}}}\right)^2 + \mu||y - y^0||^2$$
$$= y^T L y + \mu||y - y^0||^2,$$

where $\mu = \frac{1-\alpha}{\alpha}$, $L = I - S$.

Future work: tensor decomposition

- CPD form assumption

$$\mathcal{J}(A,B,C) = vec([\![A,B,C]\!])^T L vec([\![A,B,C]\!]) + \mu||[\![A,B,C]\!] - \mathcal{Y}^0||^2$$
$$= \mathbf{1}^T (C \odot B \odot A)^T L (C \odot B \odot A)\mathbf{1} + \mu||\mathcal{Y}^0 - [\![A,B,C]\!]||^2$$

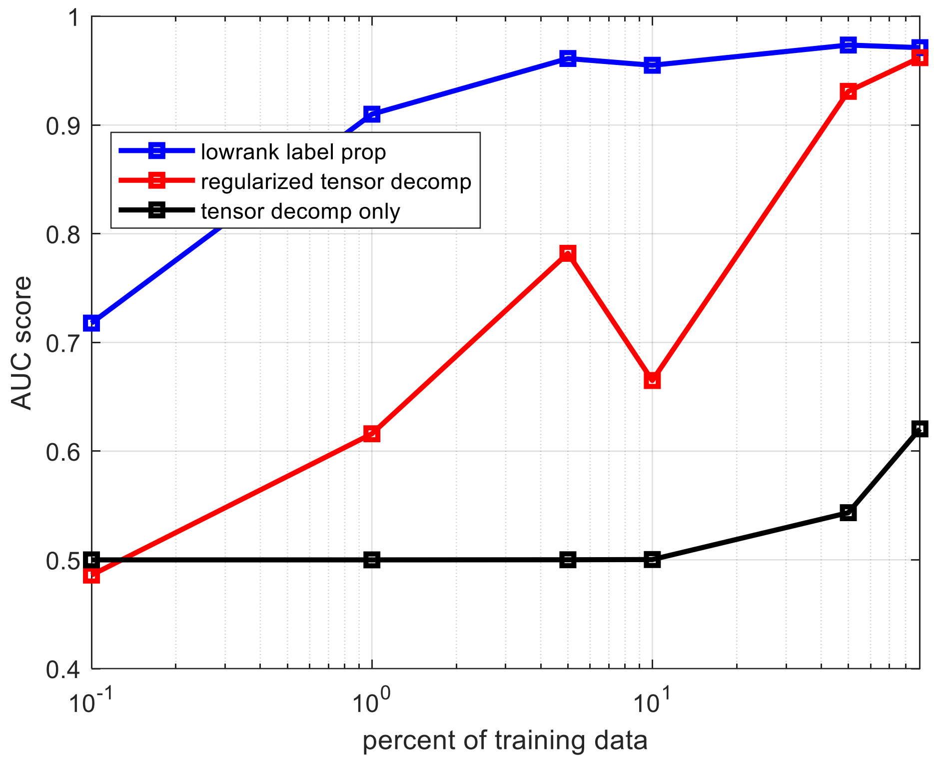
$$\frac{\partial \mathcal{J}}{\partial A} = 2(AM_1 - \mu S^{(1)} AM_2 - (1-\mu)M_3)$$

$M_1 = C^T C * B^T B$, $M_2 = C^T S^{(3)} C * B^T S^{(2)} B$ and $M_3 = Y^0_{(1)}(C \odot B)$.

Experiment 3 – DBLP dataset

- 13823 Authors, 11372 papers and 10167 venues.
- 12066 tuples.
- Experiment setting: randomly sampling 0.1%, 1%, 5%, 10%, 50% and 90% of tuples to be the training data. The rest are testing data.

Liu, H., & Yang, Y. (ICML, 2016)



References

- Zhou, Denny, et al. "Learning with local and global consistency." *NIPS (2004).*
- Liu, Hanxiao, Yuexin Wu, and Yiming Yang. "Analogical Inference for Multi-Relational Embeddings." *ICML* (2017).
- Kashima, Hisashi, et al. "Link propagation: A fast semi-supervised learning algorithm for link prediction." *SDM (*2009).
- Narita, Atsuhiro, et al. "Tensor factorization using auxiliary information." *Data Mining and Knowledge Discovery (2012).*

THANKS !