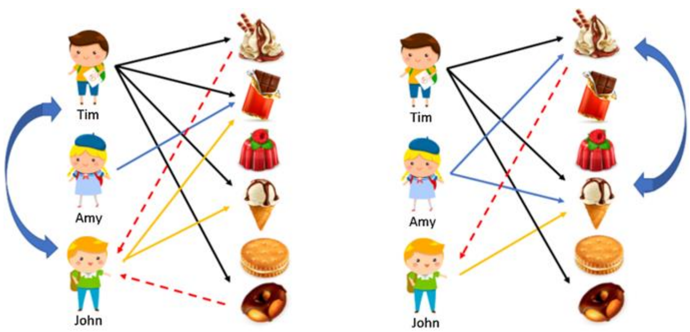
Graph 15. Learning with Heterogeneous Graphs (II)

Product Graphs [H Liu & Y Yang. ICML 2015/2016]

1

Motivating Example 1:

Recommender System over a Bipartite Graph



(a) User-based filtering **(b)** Item-based filtering

Cold Start Issue: Sparse observations on new users/items

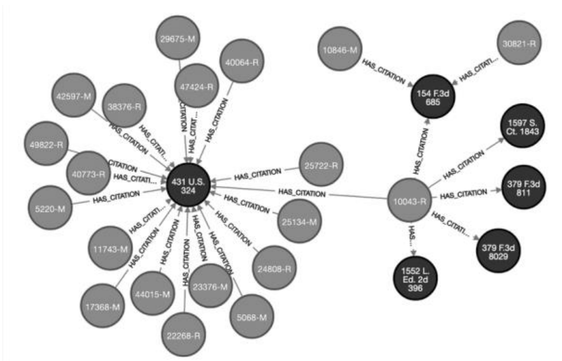
2

Motivating Example 1 (cont'd)

- **Challenge**: Cold-start issue in recommender systems
- **Remedy**: Using additional information about users and items
  - If user A and user B are friends, they may have similar tastes in watching movies.
  - If movie A and movie B share the same director and/or leading actress, they may be favored by the same user or user group.
- **Graph-based Reasoning**
  - Given a user-user similarity graph (G1), an item-item similarity graph (G2) and a partially observed bipartite graph (B), predict the missing links in B.

3

Motivating Example 2:
Publication-related Graphs

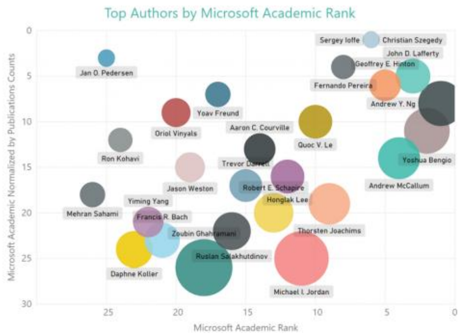


Citation Graph (useful for node classification)

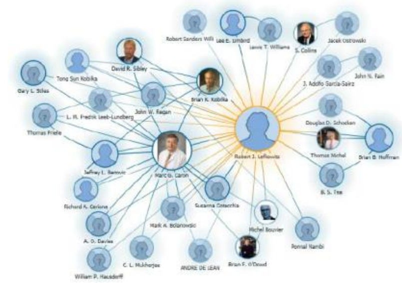
4

Motivating Example 2 (cont'd)

Publication-related Graphs



Author-Rank Graph

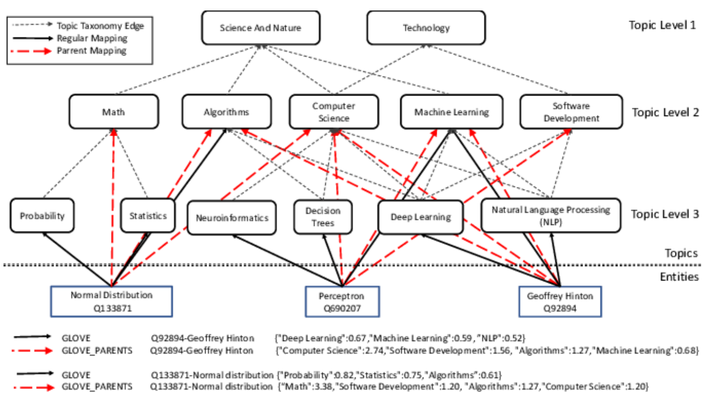


Co-author Graph

5

Motivating Example 2 (cont'd)

Publication-related Graphs



Topic Graph

6

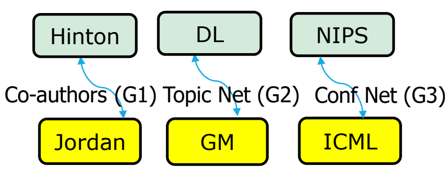
Motivating Example 2 (cont'd)

- **Challenge:** Discover the missing ones in publication records
- **Intuitions**
  - If authors A and B had many joint publications in the past, they may have shared interests in future publications.
  - If authors A and B have overlapped a lot on research topics, they are likely to attend the same conferences.
- **Graph-based Reasoning**
  - Given a co-author graph graph (G1), a citation graph (G2), a conference-conference topic-overlap graph (G3), and a set of observed multi-relational patterns (author-topic-conference tuples) , predict the tuples which are missing or likely to happen in the future.

Generalizing from Link Prediction (Example 1) to Multi-relational Prediction (Example 2)

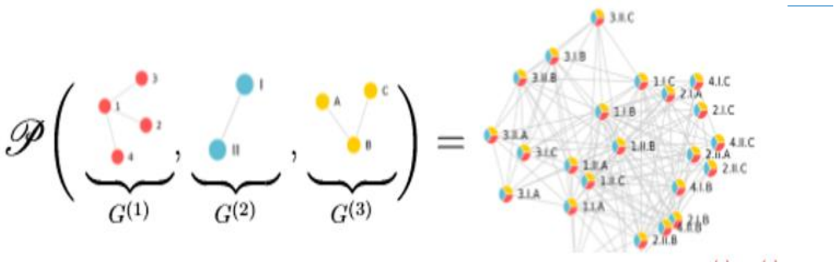
- **k-tuples (in a knowledge graph)**

  E.g., Author-Topic-Conference tuples (k=3)



- **Graph-based prediction Task**
  - Given a training set of semantically valid tuples and graphs $G_1, G_2, G_3, \cdots$ in addition, predict the unobserved semantically valid tuples.

Product Graph Construction



- Each node in $\mathcal{P}$ is a tuple with one node from each input graph.
- Each edge in $\mathcal{P}$ combines the edge strengths in input graphs as
  - **Soft AND (Tensor Product):** If all the corresponding links from tuple A to tuple B are strong, then the combined link from A to B in $\mathcal{P}$ is strong (multiplicative);
  - **Soft OR (Cartesian Product):** If any of the corresponding links from tuple A to tuple B are strong, then the combined link from A to B in $\mathcal{P}$ is strong (additive).

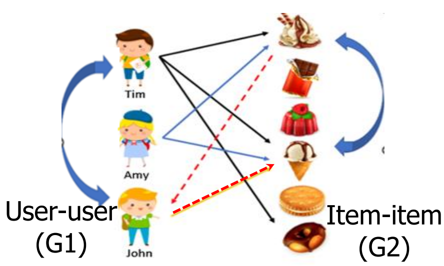
use soft And/or to build the links.

4/22/2024 @Yiming Yang, 11-741 Lecture on Product Graphs 9

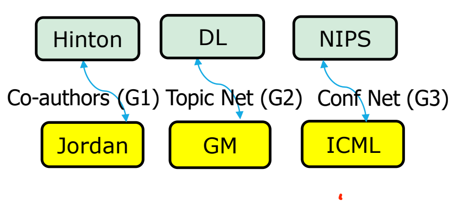
9

Combine Evidence in Tuple Predictions

Ex 1. Link Prediction (k=2)



Ex 2. Triplet Prediction (k=3)



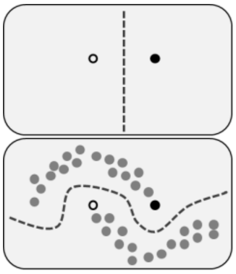
- **Soft AND** (Tensor Product) or Soft OR (**Cartesian Product)**

4/22/2024 10
@Yiming Yang, 11-741 Lecture on Product Graphs

10

Recap: SSL for Classification

- With or without Semi-Supervised Learning (SSL)



(from Wikipedia)

- Conventional SVM

$$\min_f \mathcal{L}_O(f) + c\|f\|_K^2$$

- Laplacian SVM

$$\min_f \mathcal{L}_O(f) + c_1\|f\|_K^2 + c_2\frac{l}{(l+u)^2} f^T L f$$

GCN

- Smoothness Penalty: $f^T L f$ forces well-connected nodes to have similar scores in f.

11

Our Objective for Optimization

- Define our objective as

$$\min_f \; l_O(f) + C f^T A^{-1} f$$

where $f \in R^n$ consists of the system-assigned scores to nodes in $\mathcal{P}$.

- Term $f^T A^{-1} f$ means to assume a Gaussian prior for f

$$f \sim N(0, A), \qquad log P(f) \propto -f^T A^{-1} f$$

is proportional

- Minimizing $f^T A^{-1} f$ is equivalent to maximizing the log-likelihood of the system-predicted scores based on the Gaussian prior.
- **Intuition:** we want similar scores to nearby nodes based on MD distance (instead of Euclidian distance).

12

Related Concept: Mahalanobis Distance (MD)
https://en.wikipedia.org/wiki/Mahalanobis_distance

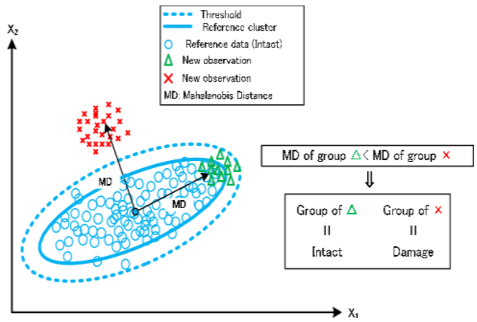
MD Definitions:

$$d(x, y) = \sqrt{(x-y)^T \Sigma^{-1} (x-y)}$$

$$\Updownarrow$$

$$d(x, \mu) = \sqrt{(x-\mu)^T \Sigma^{-1} (x-\mu)}$$

where Σ is the covariance matrix of some underlying Gaussian distribution.



https://www.researchgate.net/profile/Chul_Woo_Kim/pub668639590088740@1536427509947/Concept-of-Mahalanobis-distance-MD.pnglication/275701517/figure/fig7/AS:

13

Two Basic Product Graphs

Let $v = (v_1, v_2, \cdots, v_J)$ and $v' = (v'_1, v'_2, \cdots, v'_J)$ are two nodes in $\mathcal{P}_\otimes$.

- **Tensor Product** (mimicking SOFT AND)

$$\mathcal{P}_\otimes = G_1 \otimes G_2 \cdots \otimes G_J; \quad A_\otimes(v, v') = \prod_{i=1}^{J} A_{G_j}(v_j, v_j{}')$$

  - link weight $A_\otimes(v, v')$ is strong iff $A_{G_j}(v_j, v_j{}')$ is strong for all G's.

- **Cartesian Product** (mimicking SOFT OR).

$$\mathcal{P}_\oplus = G_1 \oplus G_2 \cdots \oplus G_J; \quad A_\oplus(v, v') = \sum_{j=1}^{J} A_{G_j}(v_j, v_j{}')$$

  - link weight $A_\oplus(v, v')$ is strong iff $A_{G_j}(v_j, v_j{}')$ is strong for any G's.

14

Enriched Formulation: $A \rightarrow \kappa(A)$

Basic Objective: $\min_f \; l_O(f) + \lambda f^T A^{-1} f$

Enriched Objective: $\min_f \; l_O(f) + \lambda f^T \kappa(A)^{-1} f$

Enhancing smoothness via multi-hop propagation:

$\kappa_{rw}(A) = A^k$ for k−step random walk

$\kappa_{von\_Neumann}(A) = (I - A)^{-1} = \mathrm{I} + \mathrm{A} + A^2 + \cdots$ for infinite walk

$\kappa_{heat}(A) = e^A = I + A + \frac{1}{2!}A^2 + \frac{1}{3!}A^3 + \cdots$ for weight-decayed walk

…

15

Spectral Graph Product (SGP)

- Use the eigen-systems of the individual input graphs to construct the eigen-system of $\mathcal{P}$

$\mathcal{P}$ is the new graph

  - The eigenvectors

$$\{v\} = \left\{ v_i^{(G_1)} \otimes v_j^{(G_2)} \otimes v_k^{(G_3)} \cdots \right\}$$

  - The eigenvalues

$$\{\lambda\} = \left\{ \kappa\left(\lambda_i^{(G_1)}, \lambda_j^{(G_2)}, \lambda_k^{(G_3)}, \cdots \right) \right\}$$

  - The adjacency matrix

$$A_\kappa = \sum_{i_1, i_2, \cdots, i_J = 1}^{n_1, n_2, \cdots, n_J} \kappa\left(\lambda_i^{(G1)}, \lambda_j^{(G2)}, \lambda_k^{(G_3)}, \cdots \right) \left\{ v_i^{(G_1)} \otimes v_j^{(G_2)} \otimes v_k^{(G_3)} \cdots \right\}$$

16

Different kernels in SGP

- Denote mapping function κ as

$$\kappa\colon \left\{\lambda_i^{(G1)}, \lambda_j^{(G2)}, \lambda_k^{(G_3)}, \cdots\right\} \to \mathbb{R}$$

  - $\kappa_{\otimes}(.) = \lambda_i^{(G1)} \times \lambda_j^{(G2)} \times \cdots \; for \; \mathcal{P}_{\otimes} = G_1 \otimes G_2 \otimes \cdots$ (Tensor Product)
  - $\kappa_{\oplus}(.) = \lambda_i^{(G1)} + \lambda_j^{(G2)} + \cdots \; for \; \mathcal{P}_{\oplus} = G_1 \oplus G_2 \oplus \cdots$ (Cartesian Product)
  - For other graph products … (next slide)

Different kernel maps in SGP

- **k−step Random Walk**

$$\kappa_{rw}(A) \triangleq A^k; \qquad \lambda_i \to \lambda_i^k$$

- **Infinite Walk (von Neumann Kernel)**

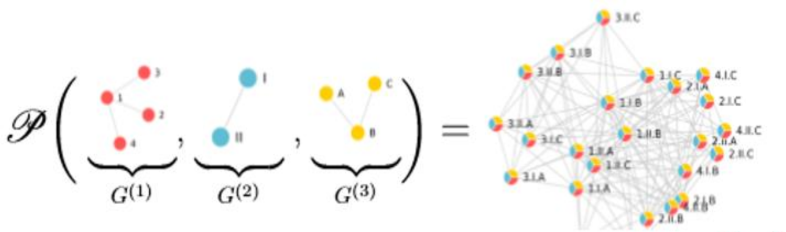
$$\kappa_{von\_Neumann}(A) \triangleq (I - A)^{-1} = \mathrm{I} + \mathrm{A} + A^2 + \cdots; \quad \lambda_i \to (1 - \lambda_i)^{-1}$$

- **Wight-decayed Walk (Heat Diffusion Kernel)**

$$\kappa_{heat}(A) \triangleq e^A = I + A + \frac{1}{2!}A^2 + \frac{1}{3!}A^3 + \cdots; \qquad \lambda_i \to e^{\lambda_i}$$

Operator over eigenvalues can be multiplicative (for $\mathcal{P}_{\otimes}$) or additive (for $\mathcal{P}_{\oplus}$).

Computational Tractability?



- # of nodes in $\mathcal{P}$ is $N = n_1 \times n_2 \times n_3 \cdots$
- # of edges is N^2

Computational Tractability

- We need to solve the problem *without explicitly* constructing A of $\mathcal{P}$
- Divide-&-conquer Strategy
  - Solving the eigen-decomposition of each graph independently
  - Approximation f *via Low-rank Tensor Decomposition*

https://bnl.zoomgov.com/j/1605669017?pwd=bWpJZ0YvZ2gzMkZ0L1pTbDI0VlkxUT09&from=addon

Meeting ID: 160 566 9017
Passcode: 748677

Calculating smoothness penalty

$A = V\Lambda V^T = \sum_{i=1}^{N} \lambda_i v_i v_i^T$

$A^{-1} = (V^T)^{-1}\Lambda^{-1}V^{-1} = V\Lambda^{-1}V^T = \sum_{i=1}^{N} \frac{v_i v_i^T}{\lambda_i}$

$\kappa(A)^{-1} = \sum_{i=1}^{N} \frac{v_i v_i^T}{\kappa(\lambda_1,\lambda_2,\cdots,\lambda_N)}$

$f^T \kappa(A)^{-1} f = \sum_{i=1}^{N} \frac{f^T v_i v_i^T f}{\kappa(\lambda_1,\lambda_2,\cdots,\lambda_N)} = \sum_{i=1}^{N} \frac{\|f^T v_i\|^2}{\kappa(\lambda_1,\lambda_2,\cdots,\lambda_N)}$

$$= \sum_{i_1,i_2,\cdots,i_J=1}^{n_1,n_2,\cdots,n_J} \frac{\left(TensorProd\left(f;\ v_{i_1}^{(G1)}, v_{i2}^{(G2)}, \cdots, v_{i_J}^{(G_J)}\right)\right)^2}{\kappa\left(\lambda_{i_1}^{(G1)},\ \lambda_{,i_2}^{(G2)}, \cdots, \lambda_{i_J}^{(G_J)}\right)}$$

21

Calculating the Smoothness Penalty

$$\boldsymbol{f}^T A^{-1} \boldsymbol{f} = \sum_{i_1,i_2,\cdots,i_J=1}^{n_1,n_2,\cdots,n_J} \frac{\left(TensorProd\left(f; v_{i_1}^{(G1)}, v_{i2}^{(G2)}, \cdots, v_{i_J}^{(G_J)}\right)\right)^2}{\kappa\left(\lambda_{i_1}^{(G1)}, \lambda_{,i_2}^{(G2)}, \cdots, \lambda_{i_J}^{(G_J)}\right)}$$

$$\approx \sum_{i_1,i_2,\cdots,i_J=1}^{k_1,k_2,\cdots,k_J} \frac{\left(TensorProd\left(\alpha; v_{i_1}^{(G1)}, v_{i2}^{(G2)}, \cdots, v_{i_J}^{(G_J)}\right)\right)^2}{\kappa\left(\lambda_{i_1}^{(G1)}, \lambda_{,i_2}^{(G2)}, \cdots, \lambda_{i_J}^{(G_J)}\right)}$$

J-mode Product on the full tensor, taking time of $O\left(\left(\sum_{j=1}^{J} n_j\right)\prod_{j=1}^{J} n_j\right)$

J-mode Product on the core tensor with $k_i = d, \forall i$, taking time of $O(d^{J+1})$, $d \ll n_j$

22

Low-rank Tensor Approximation

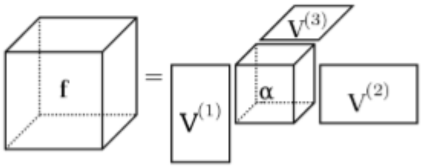


Figure: Tucker Decomposition, where α is the core tensor.

- Tensor algebras are carried out on GPU.

Smoothing Effect (a simulated example)

Low-rank approx. prunes off the high-volatility factors (eigen-vectors)

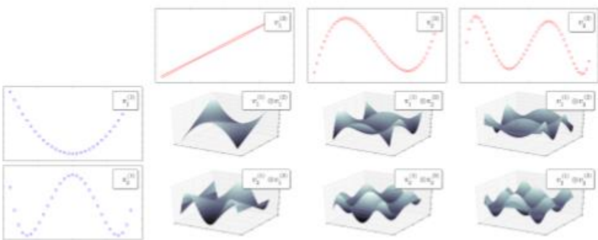


Figure: Eigenvectors of G_1 (blue), G_2 (red) and $\mathscr{P}(G_2)$.

Evaluation

Datasets

Enzyme 445 compounds, 664 proteins.

DBLP 34K authors, 11K papers, 22 venues.

Representative Baselines

TF/GRTF Tensor Factorization/Graph-Regularized TF

NN One-class Nearest Neighbor

RSVM Ranking SVMs

LTKM Low-Rank Tensor Kernel Machines

State-of-the art (SOTA) Performance

Our method: "TOP" (blue).

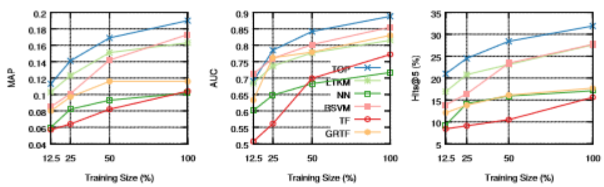
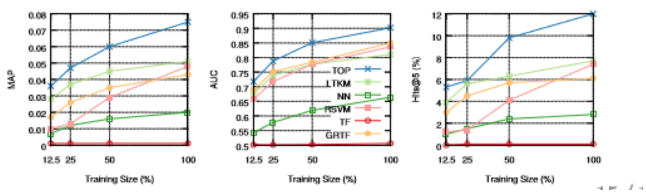


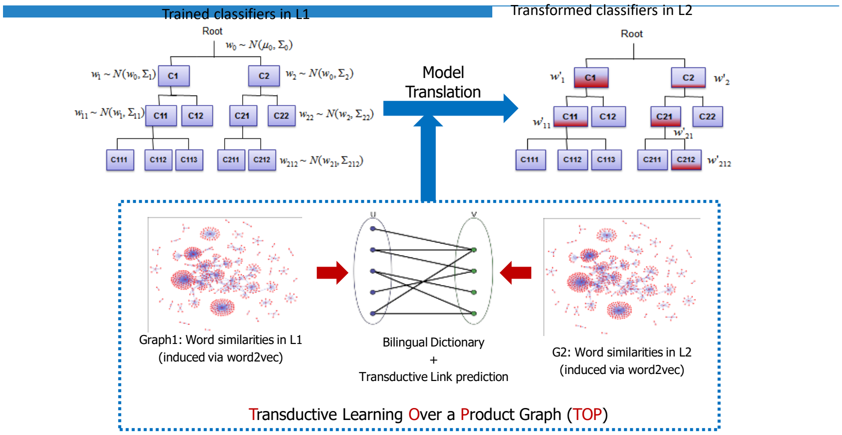
Figure: Performance on Enzyme (above) and DBLP (below).



Enzyme: predicting compound-protein interactions (link prediction)
DBLP: predicting Author-Paper- Conference triplets

Cross-Language Topic Model Translation

(R Xu et al., ACL'16)



27

Concluding Remarks

- **Challenge in multi-relational prediction**
  - Lack of sufficient labeled training data for link/tuple prediction
- **Remedy**
  - Leveraging heterogeneous graphs to enable SSL
- **Contributions**
  - Novel unified framework for integrating heterogeneous types of graph into a product graph
  - Scalable algorithms for extreme-scale convex optimization via tensor-based approximation
  - State-of-the-art performance on evaluation benchmarks
- **Limitation:** No learning of edge weights; manually specified kernels

28

References

- Hanxiao Liu and Yiming Yang. Bipartite edge prediction via transductive learning over product graphs. [pdf] *ICML* 2015.
- Hanxiao Liu and Yiming Yang. Cross-Graph Learning of Multi-Relational Associations [pdf] *ICML 2016.*
- T. G. Kolda, B. W. Bader. Tensor Decompositions and Applications. *SIAM Review*, Vol. 51, No. 3, pp. 455-500, 2009. https://doi.org/10.1137/07070111X