Cross-Graph Learning of Multi-Relational Associations

Hanxiao Liu, Yiming Yang
Carnegie Mellon University
{hanxiaol, yiming}@cs.cmu.edu

June 22, 2016

Outline

Task Description

New Contributions

Framework

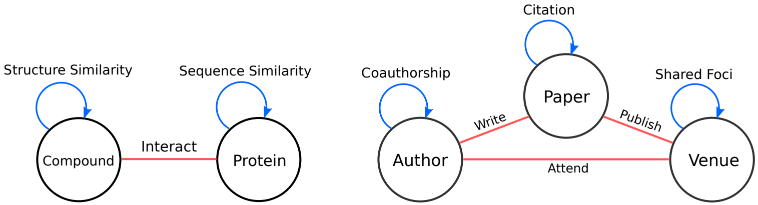
Scalable Inference

Empirical Evaluation

Summary

Task Description

Goal: Predict associations among heterogeneous graphs.



(a) Drug-Target Interaction (b) Citation Network

"John publish a reinforcement learning paper at ICML."

`(John,RL_Paper,ICML)`

Outline

Task Description

New Contributions

Framework

Scalable Inference

Empirical Evaluation

Summary

New Contributions

- A unified framework to integrating heterogeneous information in multiple graphs.
- Transductive learning to leverage both labeled data (sparse) and unlabeled data (massive).
- A convex approximation for the scalable inference over the combinatorial number of possible tuples.

Outline

Task Description

New Contributions

Framework

Scalable Inference

Empirical Evaluation

Summary

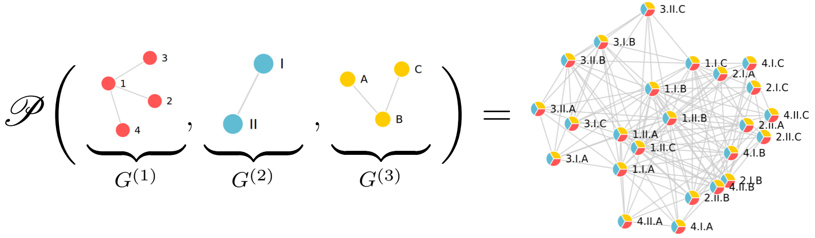
Framework

Notation

- $G^{(1)}, G^{(2)}, \ldots, G^{(J)}$ are individual graphs;
- n_j is the #nodes in $G^{(j)}$;
- $(i_1, i_2, \ldots, i_J)$ is a tuple (multi-relation);
- $f_{i_1, i_2, \ldots, i_J}$ is the predicted score for the tuple;
- f is a tensor in $\mathbb{R}^{n_1 \times n_2 \times \cdots \times n_J}$.

Framework

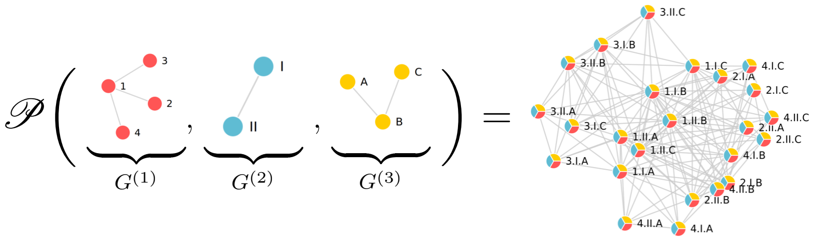
Product Graph ($\mathscr{P}$) induced from $G^{(1)}, \ldots, G^{(J)}$.



Tensor product: $\mathscr{P}(G^{(1)}, G^{(2)}, G^{(3)}) = G^{(1)} \otimes G^{(2)} \otimes G^{(3)}$

Framework

Product Graph ($\mathscr{P}$) induced from $G^{(1)}, \ldots, G^{(J)}$.



Tensor product: $\mathscr{P}(G^{(1)}, G^{(2)}, G^{(3)}) = G^{(1)} \otimes G^{(2)} \otimes G^{(3)}$

Framework

Why product graph?

- Mapping heterogeneous graphs onto a unified graph for label propagation (transductive learning).

Framework

Assuming

$$vec(f) \sim \mathcal{N}\left(0, \mathscr{P}\right) \tag{1}$$

which implies:

$$-\log p\left(f|\mathscr{P}\right) \propto vec(f)^\top \mathscr{P}^{-1} vec(f) := \|f\|_{\mathscr{P}}^2 \tag{2}$$

Optimization problem

$$\min_f \ \ell_{\mathcal{O}}(f) + \frac{\gamma}{2}\|f\|_{\mathscr{P}}^2 \tag{3}$$

Framework

Assuming

$$vec(f) \sim \mathcal{N}\left(0, \mathscr{P}\right) \tag{1}$$

which implies:

$$-\log p\left(f|\mathscr{P}\right) \propto vec(f)^\top \mathscr{P}^{-1} vec(f) := \|f\|_{\mathscr{P}}^2 \tag{2}$$

Optimization problem

$$\min_f \; \ell_{\mathcal{O}}(f) + \frac{\gamma}{2}\|f\|_{\mathscr{P}}^2 \tag{3}$$

Framework

Assuming

$$vec(f) \sim \mathcal{N}\left(0, \mathscr{P}\right) \tag{1}$$

which implies:

$$-\log p\left(f|\mathscr{P}\right) \propto vec(f)^\top \mathscr{P}^{-1} vec(f) := \|f\|_{\mathscr{P}}^2 \tag{2}$$

Optimization problem

$$\min_f \ \ell_{\mathcal{O}}(f) + \frac{\gamma}{2}\|f\|_{\mathscr{P}}^2 \tag{3}$$

Framework

For computational tractability, we focus on the spectral graph product family of $\mathscr{P}$.

Spectral Graph Product (SGP)
The eigensystem of $\mathscr{P}_\kappa\left(G^{(1)}, \ldots, G^{(J)}\right)$ is parametrized by the eigensystems of individual graphs, i.e.,

$$\left\{\kappa\big(\lambda_{i_1}, \ldots, \lambda_{i_J}\big), \bigotimes_j v_{i_j}\right\}_{i_1,\ldots,i_J} \tag{4}$$

λ_{i_j}/v_{i_j} is the i_j-th eigenvalue/eigenvector of the j-th graph.

Framework

Nice properties of SGP:

Subsuming basic operations

$$\kappa(x,y) = x \times y \implies \mathscr{P}_\kappa(G,H) = G \otimes H \quad \text{Tensor} \tag{5}$$
$$\kappa(x,y) = x + y \implies \mathscr{P}_\kappa(G,H) = G \oplus H \quad \text{Cartesian} \tag{6}$$

Supporting graph diffusions

$$\sigma_{Heat}(\mathscr{P}_\kappa) = I + \mathscr{P}_\kappa + \frac{1}{2}\mathscr{P}_\kappa^2 + \cdots = \mathscr{P}_{e^\kappa} \tag{7}$$
$$\sigma_{von-Neumann}(\mathscr{P}_\kappa) = I + \mathscr{P}_\kappa + \mathscr{P}_\kappa^2 + \cdots = \mathscr{P}_{\frac{1}{1-\kappa}} \tag{8}$$

Order-insensitive: If κ is commutative, then SGP is commutative (up to graph isomorphism).

Framework

Nice properties of SGP:

Subsuming basic operations

$$\kappa(x,y) = x \times y \implies \mathscr{P}_\kappa(G,H) = G \otimes H \quad \text{Tensor} \tag{5}$$
$$\kappa(x,y) = x + y \implies \mathscr{P}_\kappa(G,H) = G \oplus H \quad \text{Cartesian} \tag{6}$$

Supporting graph diffusions

$$\sigma_{Heat}(\mathscr{P}_\kappa) = I + \mathscr{P}_\kappa + \frac{1}{2}\mathscr{P}_\kappa^2 + \cdots = \mathscr{P}_{e^\kappa} \tag{7}$$
$$\sigma_{von-Neumann}(\mathscr{P}_\kappa) = I + \mathscr{P}_\kappa + \mathscr{P}_\kappa^2 + \cdots = \mathscr{P}_{\frac{1}{1-\kappa}} \tag{8}$$

Order-insensitive: If κ is commutative, then SGP is commutative (up to graph isomorphism).

Framework

Nice properties of SGP:

Subsuming basic operations

$$\kappa(x,y) = x \times y \implies \mathscr{P}_\kappa(G,H) = G \otimes H \quad \text{Tensor} \tag{5}$$
$$\kappa(x,y) = x + y \implies \mathscr{P}_\kappa(G,H) = G \oplus H \quad \text{Cartesian} \tag{6}$$

Supporting graph diffusions

$$\sigma_{Heat}(\mathscr{P}_\kappa) = I + \mathscr{P}_\kappa + \frac{1}{2}\mathscr{P}_\kappa^2 + \cdots = \mathscr{P}_{e^\kappa} \tag{7}$$
$$\sigma_{von-Neumann}(\mathscr{P}_\kappa) = I + \mathscr{P}_\kappa + \mathscr{P}_\kappa^2 + \cdots = \mathscr{P}_{\frac{1}{1-\kappa}} \tag{8}$$

Order-insensitive: If κ is commutative, then SGP is commutative (up to graph isomorphism).

Outline

Task Description

New Contributions

Framework

Scalable Inference

Empirical Evaluation

Summary

Scalable Inference

For general GP, the semi-norm is computed as

$$\|f\|^2_{\mathscr{P}} = vec(f)^\top \mathscr{P}^{-1} vec(f) \tag{9}$$

For SGP, $\mathscr{P}_\kappa$ no longer has to be explicitly computed.

$$\|f\|^2_{\mathscr{P}_\kappa} = \sum_{i_1, i_2, \ldots, i_J}^{n_1, n_2, \ldots, n_J} \frac{f\left(v_{i_1}, \ldots, v_{i_J}\right)^2}{\kappa\left(\lambda_{i_1}, \ldots, \lambda_{i_J}\right)} \tag{10}$$

- $f(v_{i_1}, v_{i_2}, \ldots, v_{i_J}) = f \times_1 v_{i_1} \times_2 v_{i_2} \cdots \times_J v_{i_J}$
- However, even evaluating (10) is expensive.

Scalable Inference

For general GP, the semi-norm is computed as

$$\|f\|_{\mathscr{P}}^2 = vec(f)^\top \mathscr{P}^{-1} vec(f) \tag{9}$$

For SGP, $\mathscr{P}_\kappa$ no longer has to be explicitly computed.

$$\|f\|_{\mathscr{P}_\kappa}^2 = \sum_{i_1, i_2, \ldots, i_J}^{n_1, n_2, \ldots, n_J} \frac{f\left(v_{i_1}, \ldots, v_{i_J}\right)^2}{\kappa\left(\lambda_{i_1}, \ldots, \lambda_{i_J}\right)} \tag{10}$$

- $f(v_{i_1}, v_{i_2}, \ldots, v_{i_J}) = f \times_1 v_{i_1} \times_2 v_{i_2} \cdots \times_J v_{i_J}$
- However, even evaluating (10) is expensive.

Scalable Inference

For general GP, the semi-norm is computed as

$$\|f\|^2_{\mathscr{P}} = vec(f)^\top \mathscr{P}^{-1} vec(f) \tag{9}$$

For SGP, $\mathscr{P}_\kappa$ no longer has to be explicitly computed.

$$\|f\|^2_{\mathscr{P}_\kappa} = \sum_{i_1, i_2, \ldots, i_J}^{n_1, n_2, \ldots, n_J} \frac{f\left(v_{i_1}, \ldots, v_{i_J}\right)^2}{\kappa\left(\lambda_{i_1}, \ldots, \lambda_{i_J}\right)} \tag{10}$$

- $f(v_{i_1}, v_{i_2}, \ldots, v_{i_J}) = f \times_1 v_{i_1} \times_2 v_{i_2} \cdots \times_J v_{i_J}$
- However, even evaluating (10) is expensive.

Scalable Inference

Using low-rank SGP

- f lies in the linear span of the eigenvectors of $\mathscr{P}$.
- Eigenvectors of high volatility can be pruned away.

Scalable Inference

Using low-rank SGP

- f lies in the linear span of the eigenvectors of $\mathscr{P}$.
- Eigenvectors of high volatility can be pruned away.

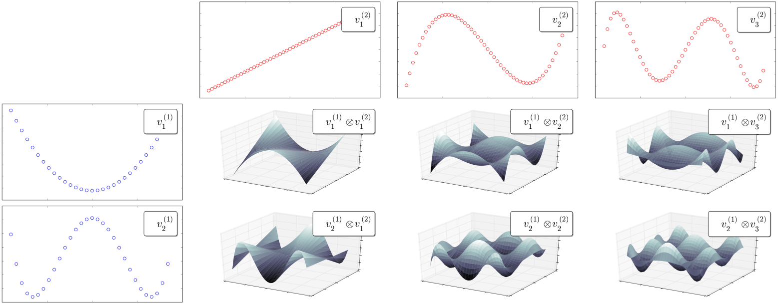


Figure : Eigenvectors of G (blue), H (red) and $\mathscr{P}(G, H)$.

Scalable Inference

Restrict f in the linear span of "smooth" bases of $\mathscr{P}$.

$$f(\alpha) = \sum_{i_1, i_2, \cdots, i_J = 1}^{d_1, d_2, \cdots, d_J} \alpha_{i_1, i_2, \cdots, i_J} \bigotimes_j v_{i_j} \tag{11}$$

where the *core tensor* $\alpha \in \mathbb{R}^{d_1 \times d_2 \times \cdots \times d_J}$, $d_j \ll n_j$.

The semi-norm becomes

$$\|f(\alpha)\|^2_{\mathscr{P}_\kappa} = \sum_{i_1, i_2, \ldots, i_J = 1}^{d_1, d_2, \cdots, d_J} \frac{\alpha^2_{i_1, i_2, \cdots, i_J}}{\kappa\big(\lambda_{i_1}, \lambda_{i_2}, \ldots, \lambda_{i_J}\big)} \tag{12}$$

We then optimize w.r.t. α instead of f. Parameter size: $\prod_j n_j \to \prod_j d_j$.

Scalable Inference

Restrict f in the linear span of "smooth" bases of $\mathscr{P}$.

$$f(\alpha) = \sum_{i_1,i_2,\cdots,i_J=1}^{d_1,d_2,\cdots,d_J} \alpha_{i_1,i_2,\cdots,i_J} \bigotimes_j v_{i_j} \tag{11}$$

where the *core tensor* $\alpha \in \mathbb{R}^{d_1 \times d_2 \times \cdots \times d_J}$, $d_j \ll n_j$.

The semi-norm becomes

$$\|f(\alpha)\|^2_{\mathscr{P}_\kappa} = \sum_{i_1,i_2,\ldots,i_J=1}^{d_1,d_2,\cdots,d_J} \frac{\alpha^2_{i_1,i_2,\cdots,i_J}}{\kappa\big(\lambda_{i_1}, \lambda_{i_2}, \ldots, \lambda_{i_J}\big)} \tag{12}$$

We then optimize w.r.t. α instead of f. Parameter size: $\prod_j n_j \to \prod_j d_j$.

Scalable Inference

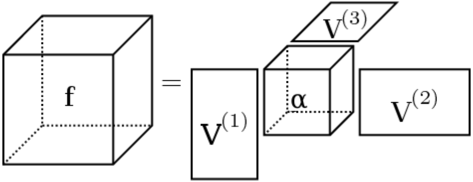


Figure : Tucker Decomposition, where α is the core tensor.

Scalable Inference

Revised optimization objective

$$\min_{\alpha \in \mathbb{R}^{d_1 \times d_2 \cdots \times d_J}} \quad \ell_{\mathcal{O}}\left(f(\alpha)\right) + \frac{\gamma}{2}\|f(\alpha)\|^2_{\mathscr{P}_\kappa} \tag{13}$$

Ranking loss function

$$\ell_{\mathcal{O}}(f) = \frac{\sum_{\substack{(i_1,\ldots,i_J) \in \mathcal{O} \\ (i'_1,\ldots,i'_J) \in \bar{\mathcal{O}}}} \left(f_{i_1 \ldots i_J} - f_{i'_1 \ldots i'_J}\right)_+^2}{|\mathcal{O} \times \bar{\mathcal{O}}|} \tag{14}$$

$$\nabla_\alpha = \frac{\partial \ell_{\mathcal{O}}}{\partial f}\left(\frac{\partial f_{i_1,\ldots,i_J}}{\partial \alpha} - \frac{\partial f_{i'_1,\ldots,i'_J}}{\partial \alpha}\right) + \gamma \alpha \oslash \kappa \tag{15}$$

Tensor algebras are carried out on GPU.

Scalable Inference

Revised optimization objective

$$\min_{\alpha \in \mathbb{R}^{d_1 \times d_2 \cdots \times d_J}} \quad \ell_{\mathcal{O}}\left(f(\alpha)\right) + \frac{\gamma}{2} \|f(\alpha)\|^2_{\mathscr{P}_\kappa} \tag{13}$$

Ranking loss function

$$\ell_{\mathcal{O}}(f) = \frac{\sum_{\substack{(i_1, \ldots, i_J) \in \mathcal{O} \\ (i'_1, \ldots, i'_J) \in \bar{\mathcal{O}}}} \left(f_{i_1 \ldots i_J} - f_{i'_1 \ldots i'_J}\right)^2_+}{|\mathcal{O} \times \bar{\mathcal{O}}|} \tag{14}$$

$$\nabla_\alpha = \frac{\partial \ell_{\mathcal{O}}}{\partial f} \left(\frac{\partial f_{i_1, \ldots, i_J}}{\partial \alpha} - \frac{\partial f_{i'_1, \ldots, i'_J}}{\partial \alpha} \right) + \gamma \alpha \oslash \kappa \tag{15}$$

Tensor algebras are carried out on GPU.

Scalable Inference

Revised optimization objective

$$\min_{\alpha \in \mathbb{R}^{d_1 \times d_2 \cdots \times d_J}} \quad \ell_{\mathcal{O}}\left(f(\alpha)\right) + \frac{\gamma}{2}\|f(\alpha)\|^2_{\mathscr{P}_\kappa} \tag{13}$$

Ranking loss function

$$\ell_{\mathcal{O}}(f) = \frac{\sum_{\substack{(i_1,\ldots,i_J) \in \mathcal{O} \\ (i'_1,\ldots,i'_J) \in \bar{\mathcal{O}}}} \left(f_{i_1 \ldots i_J} - f_{i'_1 \ldots i'_J}\right)^2_+}{|\mathcal{O} \times \bar{\mathcal{O}}|} \tag{14}$$

$$\nabla_\alpha = \frac{\partial \ell_{\mathcal{O}}}{\partial f}\left(\frac{\partial f_{i_1,\ldots,i_J}}{\partial \alpha} - \frac{\partial f_{i'_1,\ldots,i'_J}}{\partial \alpha}\right) + \gamma \alpha \oslash \kappa \tag{15}$$

Tensor algebras are carried out on GPU.

Outline

Task Description

New Contributions

Framework

Scalable Inference

Empirical Evaluation

Summary

Empirical 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

Empirical 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

Empirical Evaluation

Our method: “TOP” (blue).

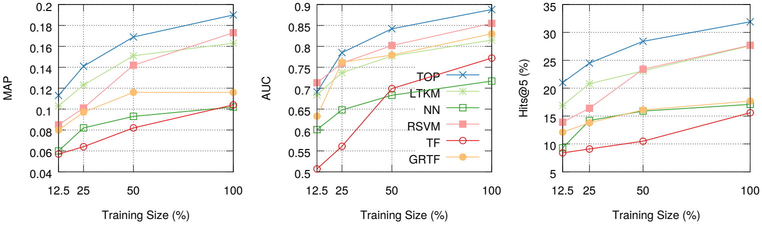
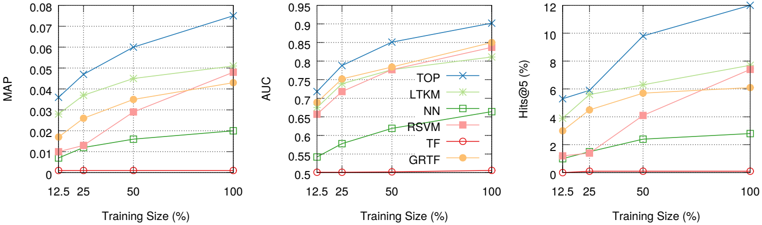


Figure : Performance on Enzyme (above) and DBLP (below).



Outline

Task Description

New Contributions

Framework

Scalable Inference

Empirical Evaluation

Summary

Summary

Contribution

- A unified framework to integrating heterogeneous information in multiple graphs.
- Transductive learning to leverage both labeled data (sparse) and unlabeled data (massive).
- A convex approximation for the scalable inference over the combinatorial number of possible tuples.

Future/On-going Work

- Learning structured associations.
- Larger problems: Microsoft Academic Graph (37 GB).

Summary

Contribution

- A unified framework to integrating heterogeneous information in multiple graphs.
- Transductive learning to leverage both labeled data (sparse) and unlabeled data (massive).
- A convex approximation for the scalable inference over the combinatorial number of possible tuples.

Future/On-going Work

- Learning structured associations.
- Larger problems: Microsoft Academic Graph (37 GB).

Thank You