Partial Gromov-Wasserstein Learning for Partial Graph Matching

Weijie Liu, Chao Zhang, Jiahao Xie, Zebang Shen, Hui Qian, Nenggan Zheng

2021.3.5

Outlines

Graph Matching (Alignment)

- Graph matching finds the correspondence of nodes across two graphs.
- Numerous existing methods match **every** node in one graph to one node in the other graph.
- But two graphs usually overlap partially in many real-world applications.

Full Matching Problem

Graph matching aims to find a matching matrix $\mathbf{T}$ between two graphs $\mathcal{G}^s$ and $\mathcal{G}^t$.
Assuming $\mathcal{G}^s$ has fewer nodes than $\mathcal{G}^t$, $\mathbf{T}$ is the solution to the following optimization formulation

$$\min_{\mathbf{T}\in\mathcal{P}^f} \left[\sum_{ii'} k_{ii'} T_{ii'} + \sum_{ii'jj'} d_{ii'jj'} T_{ii'} T_{jj'} \right] \tag{1}$$

subject to the feasible domain

$$\mathcal{P}^f = \left\{ \mathbf{T} \mid \mathbf{T} \in \{0,1\}^{m\times n}, \mathbf{T}1 = \mathbf{1}, \mathbf{T}^\top \mathbf{1} \leq \mathbf{1} \right\} \tag{2}$$

$\mathbf{T} = [T_{ii'}]$: $T_{ii'} = 1$ if node i in $\mathcal{G}^s$ maps to node i' in $\mathcal{G}^t$, and $T_{ii'} = 0$ otherwise.
$k_{ii'}$: value of the cost function for the unary assignment $i \to i'$.
$d_{ii'jj'}$: cost for the pairwise assignment $(i,j) \to (i',j')$.

Partial Matching

$$\min_{\mathbf{T}\in\mathcal{P}^p}\left[\sum_{ii'} k_{ii'}T_{ii'} + \sum_{ii'jj'} d_{ii'jj'}T_{ii'}T_{jj'}\right] \tag{3}$$

subject to the relaxed feasible domain

$$\mathcal{P}^p = \left\{\mathbf{T} \mid \mathbf{T}\in\{0,1\}^{m\times n}, \mathbf{T1}\leq\mathbf{1}, \mathbf{T}^\top\mathbf{1}\leq\mathbf{1}\right\} \tag{4}$$

Limitations:

- limit the choice of cost functions;
- can lead to extreme cases(degraded matching $\mathbf{T}=\mathbf{0}$ or still full matching with $\mathbf{T1}=\mathbf{1}$).

Solution: A novel partial graph matching framework based on partial OT.

Let $\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^m$ and $\mathcal{Y} = \left\{\mathbf{y}_j\right\}_{j=1}^n$ be two sample spaces.
We assume two discrete probability distributions over $\mathcal{X}$ and $\mathcal{Y}$ respectively, i.e.

$$\mathbf{p} = \sum_{i=1}^{m} p_i \delta(\mathbf{x}_i) \text{ and } \mathbf{q} = \sum_{j=1}^{n} q_j \delta\left(\mathbf{y}_j\right), \text{ s.t. } \mathbf{p} \in \Sigma^m, \mathbf{q} \in \Sigma^n$$

The p -Wasserstein distance between $\mathbf{p}$ and $\mathbf{q}$:

$$W_p^p(\mathbf{p}, \mathbf{q}) = \min_{\mathbf{T} \in \Pi(\mathbf{p}, \mathbf{q})} \langle \mathbf{K}^p, \mathbf{T} \rangle \tag{5}$$

where cost matrix $\mathbf{K}^p = \left[K_{ij}^p\right]$ and transport matrix $\mathbf{T} = \left[T_{ij}\right]$.
Gromov-Wasserstein (GW) distance:

$$GW_p^p(\mathbf{p}, \mathbf{q}) = \min_{\mathbf{T} \in \Pi(\mathbf{p}, \mathbf{q})} \sum_{i,j=1}^{m} \sum_{i',j'=1}^{n} D_{ii'jj'}^p T_{ii'} T_{jj'} \tag{6}$$

Partial OT

The partial OT problem focuses on transporting only a fraction $0 \leq b \leq 1$ of the mass with minimum transportation costs. So the set of admissible couplings is

$$\Pi^b(\mathbf{p}, \mathbf{q}) = \left\{\mathbf{T} \mid \mathbf{T} \in \mathbb{R}_+^{m \times n}, \mathbf{T}1 \leq \mathbf{p}, \mathbf{T}^\top \mathbf{1} \leq \mathbf{q}, \mathbf{1}^\top \mathbf{T}\mathbf{1} = b\right\}. \tag{7}$$

Partial Wasserstein distance and partial GW distance:

$$PW_p^p(\mathbf{p}, \mathbf{q}) = \min_{\mathbf{T} \in \Pi^b(\mathbf{p}, \mathbf{q})} \langle \mathbf{K}^p, \mathbf{T} \rangle,$$

$$PGW_p^p(\mathbf{p}, \mathbf{q}) = \min_{\mathbf{T} \in \Pi^b(\mathbf{p}, \mathbf{q})} \sum_{i,j=1}^{m} \sum_{i',j'=1}^{n} D_{ii'jj'}^p T_{ii'} T_{jj'}$$

Notations for Graph

We consider undirected and attributed graphs as $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{X}, \mathbf{W}, f)$.

- $\mathcal{V}$ and $\mathcal{E}$ are the sets of vertices and edges of the graph, respectively.
- Node features are summarized in a $|\mathcal{V}| \times D$ matrix $\mathbf{X}$.
- $\mathbf{W} \in \mathbb{R}^{|\mathcal{V}| \times |\mathcal{V}|}$ assigns each observed edge $\left(u_i, u_j\right) \in \mathcal{E}$ for $u_i, u_j \in \mathcal{V}$ a weight w_{ij} .
- $f : \mathbf{X} \rightarrow \mathbf{Z}$ associates each vertex u_i with some representation vector $\mathbf{z}_i$.
- Cost matrix $\mathbf{C} = \left[c_{ij}\right] \in \mathbb{R}^{|\mathcal{V}| \times |\mathcal{V}|}$ (Each c_{ij} characterizing the dissimilarity between vertices i and j.)

Given a source graph $\mathcal{G}^s$ and a target graph $\mathcal{G}^t$ ($|\mathcal{V}^s| \le \left|\mathcal{V}^t\right|$), we assign two distributions $\boldsymbol{\mu}^s = \left[\mu_i^s\right]$ and $\boldsymbol{\mu}^t = \left[\mu_i^t\right]$ to the two graphs, where

$$\mu_i^z = \frac{\sum_j w_{ij}^z}{\sum_{ij} w_{ij}^z}, for \quad z = s, t. \tag{8}$$

GW Distance for Graph Matching

The GW distance between $\mathcal{G}^s$ and $\mathcal{G}^t$:

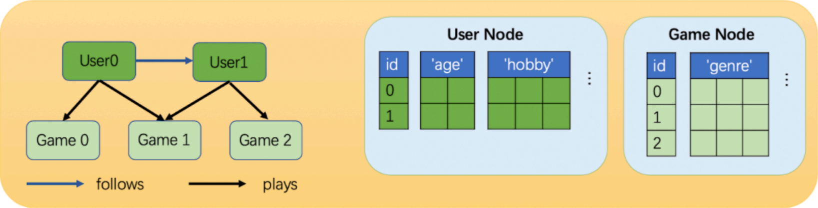
$$\begin{aligned} GW\left(\boldsymbol{\mu}^s, \boldsymbol{\mu}^t\right) &= \min_{\mathbf{T}\in\Pi\left(\boldsymbol{\mu}^s,\boldsymbol{\mu}^t\right)} \sum_{i,j,i',j'} \ell\left(c^s_{ij}, c^t_{i'j'}\right) T_{ii'} T_{jj'} \\ &= \min_{\boldsymbol{T}\in\Pi\left(\boldsymbol{\mu}^s,\boldsymbol{\mu}^t\right)} \left\langle \mathbf{L}\left(\mathbf{C}^s, \mathbf{C}^t, \mathbf{T}\right), \mathbf{T}\right\rangle \end{aligned} \tag{9}$$

- $\ell\left(c^s_{ij}, c^t_{i'j'}\right)$ measures the distance between two scalars c^s_{ij} and $c^t_{i'j'}$.
- $\mathbf{L}\left(\mathbf{C}^s, \mathbf{C}^t, \mathbf{T}\right) = \left[l_{jj'}\right] \in \mathbb{R}^{|\mathcal{V}^s|\times|\mathcal{V}^t|}$
- $l_{jj'} = \sum_{i,i'} \ell\left(c^s_{ij}, c^t_{i'j'}\right) T_{ii'}$.

The optimal transport map indicates the correpsondence of vertices across the two graphs.

Heterogeneous Graphs

A heterogeneous graph contains multiple types of nodes and edges.



Neural networks like LSTM can generate node embeddings.

$$\max_f \sum_{u \in \mathcal{V}} \log \mathbb{P}\left(\mathcal{N}_u \mid f(u)\right) \tag{10}$$

$\mathcal{N}_u$: neighborhood of vertices u.
$f : \mathcal{V} \to \mathbb{R}^d$: the neural network to be trained.

Proposed Model

We simultaneously learn

- the transport map $\mathbf{T}$(indicating the correpsondence between graphs) and
- the node embeddings $\mathbf{Z}^s$ and $\mathbf{Z}^t$ that are parameterized by $\boldsymbol{\theta}^s$ and $\boldsymbol{\theta}^t$ respectively.

$$\begin{aligned}&\min_{\mathbf{T},\boldsymbol{\theta}^s,\boldsymbol{\theta}^t} \underbrace{\left\langle \mathbf{L}\left(\mathbf{C}^s(\boldsymbol{\theta}^s), \mathbf{C}^t\left(\boldsymbol{\theta}^t\right), \mathbf{T}\right), \mathbf{T}\right\rangle}_{\text{Gromov-Wasserstein discrepancy}} \\ &+ \underbrace{\alpha\left\langle \mathbf{K}\left(\mathbf{Z}^s(\boldsymbol{\theta}^s), \mathbf{Z}^t\left(\boldsymbol{\theta}^t\right)\right), \mathbf{T}\right\rangle}_{\text{Wasserstein discrepancy}} + \underbrace{\alpha_1 R\left(\mathbf{Z}^s(\boldsymbol{\theta}^s), \mathbf{Z}^t\left(\boldsymbol{\theta}^t\right)\right)}_{\text{prior information}}.\end{aligned} \tag{11}$$

subject to $\mathbf{T} \in \Pi^b\left(\boldsymbol{\mu}^s, \boldsymbol{\mu}^t\right)$, where

$$\mathbf{C}^z(\boldsymbol{\theta}^z) = (1-\alpha)\mathbf{G}^z + \alpha\mathbf{K}\left(\mathbf{Z}^z(\boldsymbol{\theta}^z), \mathbf{Z}^z(\boldsymbol{\theta}^z)\right) \tag{12}$$

$\mathbf{G}^z = \left[g\left(w_{ij}^z\right)\right] \in \mathbb{R}^{|\mathcal{V}^z|\times|\mathcal{V}^z|}$,
$\mathbf{K}\left(\mathbf{Z}^z(\boldsymbol{\theta}^z), \mathbf{Z}^z(\boldsymbol{\theta}^z)\right) = \left[k\left(\mathbf{z}_i(\boldsymbol{\theta}^z), \mathbf{z}_j(\boldsymbol{\theta}^z)\right)\right] \in \mathbb{R}^{|\mathcal{V}^z|\times|\mathcal{V}^z|}$.

Given $\boldsymbol{\theta}_m^s$ and $\boldsymbol{\theta}_m^t$, we solve the following sub-problem

$$\min_{\mathbf{T}} \left\langle \mathbf{L}\left(\mathbf{C}^s\left(\boldsymbol{\theta}_m^s\right), \mathbf{C}^t\left(\boldsymbol{\theta}_m^t\right), \mathbf{T}\right) + \alpha \mathbf{K}\left(\mathbf{Z}^s\left(\boldsymbol{\theta}_m^s\right), \mathbf{Z}^t\left(\boldsymbol{\theta}_m^t\right)\right), \mathbf{T}\right\rangle \tag{13}$$

subject to $\mathbf{T} \in \Pi^b\left(\boldsymbol{\mu}^s, \boldsymbol{\mu}^t\right)$.

Proximal Point Method to Learn OT

In the n -th iteration, we update $\mathbf{T}$ by solving

$$\begin{aligned}\min_{\mathbf{T}\in\Pi^b(\boldsymbol{\mu}^s,\boldsymbol{\mu}^t)} A_n(\mathbf{T}) := \Big\langle \mathbf{L}\left(\mathbf{C}^s\left(\boldsymbol{\theta}_m^s\right),\mathbf{C}^t\left(\boldsymbol{\theta}_m^t\right),\mathbf{T}\right) \\ +\alpha\mathbf{K}\left(\mathbf{Z}^s\left(\boldsymbol{\theta}_m^s\right),\mathbf{Z}^t\left(\boldsymbol{\theta}_m^t\right)\right),\mathbf{T}\Big\rangle + \gamma\mathrm{KL}\left(\mathbf{T}\|\mathbf{T}_n\right)\end{aligned} \tag{14}$$

which is solved via projected gradient descent

$$\mathbf{T}_{l+1,n} = \mathrm{Proj}^{\mathrm{KL}}_{\mathbf{T}\in\Pi^b(\boldsymbol{\mu}^s,\boldsymbol{\mu}^t)}\left(\mathbf{T}_{l,n}\odot e^{-\tau\nabla A_n(\mathbf{T}_{l,n})}\right) \tag{15}$$

where the projection according to the KL divergence onto a convex set C is defined as $\mathrm{Proj}^{\mathrm{KL}}_{\mathbf{T}\in C}(\mathbf{A}) = \mathrm{argmin}_{\mathbf{T}'\in C}\,\mathrm{KL}\left(\mathbf{T}'\|\mathbf{A}\right)$.

Complexity for Learning OT

- $\mathbf{L}\left(\mathbf{C}^s, \mathbf{C}^t, \mathbf{T}\right)$: $\mathcal{O}\left(V^3\right)$, where $V = \max\left\{|\mathcal{V}^s|, \left|\mathcal{V}^t\right|\right\}$
- $\mathbf{K}\left(\mathbf{Z}^s(\boldsymbol{\theta}^s), \mathbf{Z}^t\left(\boldsymbol{\theta}^t\right)\right)$: $\mathcal{O}\left(V^2 d\right)$, where d is the dimension of embeddings
- Total cost: $\mathcal{O}\left(V^2 d + N V^3\right)$.

Updating Embeddings

Let $\ell(\cdot,\cdot)$ be an element-wise loss function. Then $R\left(\mathbf{Z}^s(\boldsymbol{\theta}^s),\mathbf{Z}^t\left(\boldsymbol{\theta}^t\right)\right)$ for homogeneous graphs is given by

$$\sum_{z=s,t}\ell\left(\mathbf{K}\left(\mathbf{Z}^z(\boldsymbol{\theta}^z),\mathbf{Z}^z(\boldsymbol{\theta}^z)\right),\mathbf{G}^k\right)+\underbrace{\ell\left(\mathbf{K}\left(\mathbf{Z}^s(\boldsymbol{\theta}^s),\mathbf{Z}^t\left(\boldsymbol{\theta}^t\right)\right),\mathbf{G}^{st}\right)}_{\text{optional}}.$$

$\mathbf{G}^{st}\in\mathbb{R}^{|\mathcal{V}^s|\times|\mathcal{V}^t|}$: ground truth correspondences across $\mathcal{G}^s$ and $\mathcal{G}^t$.
Given the learned transport plan $\hat{\mathbf{T}}_m$, we update the embeddings as follows

$$\begin{gathered}\min_{\boldsymbol{\theta}^s,\boldsymbol{\theta}^t}\left\langle\mathbf{L}\left(\mathbf{C}^s(\boldsymbol{\theta}^s),\mathbf{C}^t\left(\boldsymbol{\theta}^t\right),\hat{\mathbf{T}}_m\right),\hat{\mathbf{T}}_m\right\rangle\\+\alpha\left\langle\mathbf{K}\left(\mathbf{Z}^s(\boldsymbol{\theta}^s),\mathbf{Z}^t\left(\boldsymbol{\theta}^t\right)\right),\hat{\mathbf{T}}_m\right\rangle+\alpha_1 R\left(\mathbf{Z}^s(\boldsymbol{\theta}^s),\mathbf{Z}^t\left(\boldsymbol{\theta}^t\right)\right)\end{gathered}$$

which can be solved effectively by SGD.

PPGM

Algorithm 1 Partial optimal transport-based Partial Graph Matching (PPGM)

Input: α_1, total rounds M, graphs $\mathcal{G}^s$ and $\mathcal{G}^t$, total transport mass b.
Output: Correspondence set $\mathcal{C}$.
Initialize $\boldsymbol{\theta}^s$ and $\boldsymbol{\theta}^t$ randomly, $\hat{\mathbf{T}}_0 = \boldsymbol{\mu}^s \boldsymbol{\mu}^{t\top}$.
for $m = 0, \ldots, M-1$ **do**
  $\alpha = \frac{m}{M}$
  Calculate $\hat{\mathbf{T}}_{m+1}$ by solving (8).
  Update embedding via solving (14) or (15)
  Store the embedding
end for
$\hat{\mathbf{T}} = \hat{\mathbf{T}}_M$
$\tilde{\mathbf{T}} = [\hat{\mathbf{T}}, \mathbf{1} - \hat{\mathbf{T}}\mathbf{1}]$
Initialize correspondence set $\mathcal{C} = \varnothing$
for $u_i \in \mathcal{V}^s$ **do**
  $j = \arg\max_j \tilde{T}_{ij}$.
  if $j \neq (|\mathcal{V}^t| + 1)$ **then**
    $\mathcal{C} = \mathcal{C} \cup \{(u_i, v_j)\}$
  end if
end for

Heterogeneous Network Embedding

3 parts to obtain the embeddings $\{f(u; \boldsymbol{\theta}^z) \mid u \in \mathcal{V}^z, z = s, t\}$:

1. sampling heterogeneous neighbours: Random Walk with Restart(RWR) strategy;
2. encoding contents: multilayer perceptron MLP-r;
3. aggregating heterogeneous neighbours: Bi-LSTM and attention mechanism.

Increasing Inter-type Separability

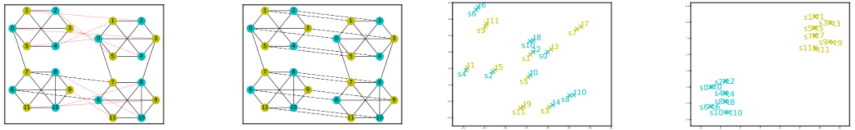
$$
\begin{aligned}
R\left(\mathbf{Z}^{s}\left(\boldsymbol{\theta}^{s}\right), \mathbf{Z}^{t}\left(\boldsymbol{\theta}^{t}\right)\right) = \sum_{z=s,t} & \left(-\log \mathbb{P}\left(\mathcal{N}_{u}^{z} \mid f(u; \boldsymbol{\theta}^{z})\right)\right. \\
& \left. +\zeta \sum_{u \in \mathcal{V}^{z}} \left(r_{u} - h^{z}\left(\mathbf{z}\left(\boldsymbol{\theta}^{z}\right)\right)\right)^{2}\right)
\end{aligned} \tag{16}
$$

Given the optimal transport $\hat{\mathbf{T}}_m$, we update the embeddings by solving

$$
\begin{aligned}
& \min_{\boldsymbol{\theta}^{s}, \boldsymbol{\theta}^{t}, \boldsymbol{\beta}^{s}, \boldsymbol{\beta}^{t}} \left\langle \mathbf{L}\left(\mathbf{C}^{s}\left(\boldsymbol{\theta}^{s}\right), \mathbf{C}^{t}\left(\boldsymbol{\theta}^{t}\right), \hat{\mathbf{T}}_{m}\right), \hat{\mathbf{T}}_{m}\right\rangle + \alpha \left\langle \mathbf{K}\left(\mathbf{Z}^{s}\left(\boldsymbol{\theta}^{s}\right), \mathbf{Z}^{t}\left(\boldsymbol{\theta}^{t}\right)\right), \hat{\mathbf{T}}_{m}\right\rangle \\
& +\alpha_{1} \sum_{z=s,t} \left(-\log \mathbb{P}\left(\mathcal{N}_{u}^{z} \mid f(u; \boldsymbol{\theta}^{z})\right)\right. \\
& \qquad \left. + \zeta \sum_{u \in \mathcal{V}^{z}} \left(r_{u} - h^{z}\left(\mathbf{z}\left(\boldsymbol{\theta}^{z}\right)\right)\right)^{2} \right)
\end{aligned}
$$

where r_u is the type of u.

Comparison between PPGW(b=1) and GWL



(a) Alignment of GWL (b) Alignment of ours (c) Embeddings of GWL (d) Embeddings of ours

- PPGW successfully aligns all nodes while GWL only aligns two pairs of nodes correctly.
- The distances between embedding vectors directly indicate the node correspondence. In GWL, embedding vectors of different types are in the same region and cannot be distinguished, which worsens the alignment.

Setting

- $\ell(\cdot,\cdot)$: square loss
- $g\left(w_{ij}\right) = \frac{1}{1+w_{ij}}$
- $k\left(\mathbf{z}_i, \mathbf{z}_j\right) = 1 - \exp\left(-\delta\left(1 - \frac{\mathbf{z}_{\mathbf{z}}^{\top}\mathbf{z}_j}{\|\mathbf{z}_i\|\|\mathbf{z}_j\|}\right)\right)$

$$\begin{aligned} \text{recall} &= \frac{\#\{\text{ correct matching }\}}{\#\{\text{ ground truth matching }\}} \\ \text{precision} &= \frac{\#\{\text{ correct matching }\}}{\#\{\text{ total predicted matching }\}}, \end{aligned}$$

$$\text{F1} = 2\frac{\text{recall } \cdot \text{ precision}}{\text{recall } + \text{ precision}}.$$

Homogeneous Synthetic Datasets

ρ	0.7			0.5			0.3		
	recall	precision	F1	recall	precision	F1	recall	precision	F1
SuperGlue	.40 ± .10	.36 ± .12	.38 ± .11	.40 ± .10	.33 ± .08	.36 ± .09	.23 ± .01	.19 ± .01	.21 ± .01
ZAC	.69 ± .03	.62 ± .01	.65 ± .02	.35 ± .07	.30 ± .08	.32 ± .07	.20 ± .02	.17 ± .02	.19 ± .02
BB-GM	.25 ± .03	.20 ± .03	.23 ± .03	.22 ± .04	.16 ± .03	.18 ± .03	.19 ± .03	.12 ± .01	.14 ± .01
PPGM	**.83 ± .02**	**.73 ± .01**	**.78 ± .02**	**.46 ± .02**	**.42 ± .01**	**.44 ± .01**	**.38 ± .02**	**.35 ± .03**	**.36 ± .02**

Table 1: The performance of PPGM and state-of-the-art methods on K-NN graph datasets with varying overlap ratio ρ.

ρ	0.7			0.5			0.3		
	recall	precision	F1	recall	precision	F1	recall	precision	F1
SuperGlue	.34 ± .06	.31 ± .05	.32 ± .05	.29 ± .03	.25 ± .04	.27 ± .04	.26 ± .02	.19 ± .01	.22 ± .01
ZAC	.37 ± .03	.34 ± .04	.35 ± .04	.34 ± .01	.30 ± .01	.32 ± .01	.24 ± .06	.21 ± .06	.23 ± .06
BB-GM	.31 ± .07	.26 ± .06	.28 ± .06	.25 ± .03	.19 ± .02	.21 ± .03	.18 ± .01	.12 ± .01	.14 ± .01
PPGM	**.64 ± .09**	**.54 ± .08**	**.58 ± .08**	**.55 ± .06**	**.43 ± .06**	**.48 ± .06**	**.48 ± .06**	**.35 ± .05**	**.40 ± .05**

Table 2: The performance of PPGM and state-of-the-art methods on BA graph datasets with varying overlap ratio ρ.

Heterogeneous Real-world Graphs

Data	Node	Edge
Source Graph	# author: 286 # paper: 286 # venue: 18	# author-paper: 618 # paper-paper: 133 # paper-venue: 286
Target Graph	# author: 286 # paper: 286 # venue: 18	# author-paper: 618 # paper-paper: 123 # paper-venue: 286

Table 3: Datasets used in the heterogeneous graph matching experiment.

method	recall	precision	F1
SuperGlue	$.019 \pm .005$	$.026 \pm .008$	$.022 \pm .006$
ZAC	$.006 \pm .001$	$.005 \pm .001$	$.005 \pm .001$
BB-GM	$.001 \pm .001$	$.005 \pm .005$	$.002 \pm .002$
PPGM	$\mathbf{.030 \pm .007}$	$\mathbf{.058 \pm .011}$	$\mathbf{.039 \pm .006}$

Table 4: The performance of PPGM and state-of-the-art methods on academic social networks.