Rethinking Kernel Methods for Node Representation Learning on Graphs

Yu Tian[1*] Long Zhao[1*] Xi Peng[2] Dimitris Metaxas[1]
[1] Rutgers University [2] University of Delaware
* Equal Contribution

NEURAL INFORMATION PROCESSING SYSTEMS

Highlights

- We study the problem of learning node representations on graphs.
- We propose a learnable kernel-based framework for node classification.
- We demonstrate the validity of our learnable kernel function and show that our formulation is powerful enough to express any p.s.d. kernels.
- A novel feature aggregation mechanism for learning node representation is derived.

Kernel Concepts

- A kernel $K: \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}$ is a function of two arguments: $K(x, y)$ for $x, y \in \mathcal{X}$.
- The kernel function K is symmetric: $K(x,y) = K(y,x)$.
- K is a p.s.d. kernel $\Leftrightarrow$ $\mathrm{K} = \begin{bmatrix} K(x_1,x_1) & \cdots & K(x_1,x_N) \\ \vdots & \ddots & \vdots \\ K(x_N,x_1) & \cdots & K(x_N,x_N) \end{bmatrix}$ is positive semidefinite for any $\{x_i\}_{i=1}^N$.

Graph Kernels

- Given two graphs $G_i = (V_i, E_i)$ and $G_j = (V_j, E_j)$, the graph kernel $K_G(G_i, G_j) \coloneqq \sum_{v_i \in V_i} \sum_{v_j \in V_j} k_{base}\left(f(v_i), f(v_j)\right)$.
- K_G should be p.s.d and symmetric.
- Drawback: hand-crafted kernel, little learnable params.

Representation Learning on Graphs

- Representation learning as an encoder-decoder framework: $\mathcal{L} = \sum_{(v_i,v_j)\in\mathcal{D}} \ell\left(ENC_{DEC(v_i,v_j)}, s_G(v_i, v_j)\right)$.
- ENC_DEC: encoder-decoder function.
- s_G: measuring the similarity between nodes in G.
- ℓ: loss function.

Learning Kernels for Node Representation

- Replace ENC_DEC with a kernel function K_θ: $\mathcal{L} = \sum_{(v_i,v_j)\in\mathcal{D}} \ell\left(K_\theta(v_i,v_j), s_G(v_i,v_j)\right)$.
- Decouple K_θ into two components: $K_\theta(v_i,v_j) = k_{base}(g_\theta(v_i), g_\theta(v_j))$

base p.s.d. kernel k_{base} | learnable feature mapping function g_θ

- Theorem 1 (validity): k_{base} is p.s.d. $\Longrightarrow$ K_θ is p.s.d.
- Theorem 2 (powerful): For some k_{base}, $k_{base}(g_\theta(v_i), g_\theta(v_j))$ can express any p.s.d. kernel.

Feature Mapping Function:

- $g_\theta(V) \coloneqq \left(\sum_h \omega_h \cdot (\bar{A}^h \odot M^{(h)})\right) \cdot MLP^{(l)}(X_V)$

$M^{(h)}(i,j) = 1$ if v_j is a h-hop neighbor of v_i

ω_h learnable

- GCN as feature mapping: g_{GCN}
- GAT as feature mapping: g_{GAT}

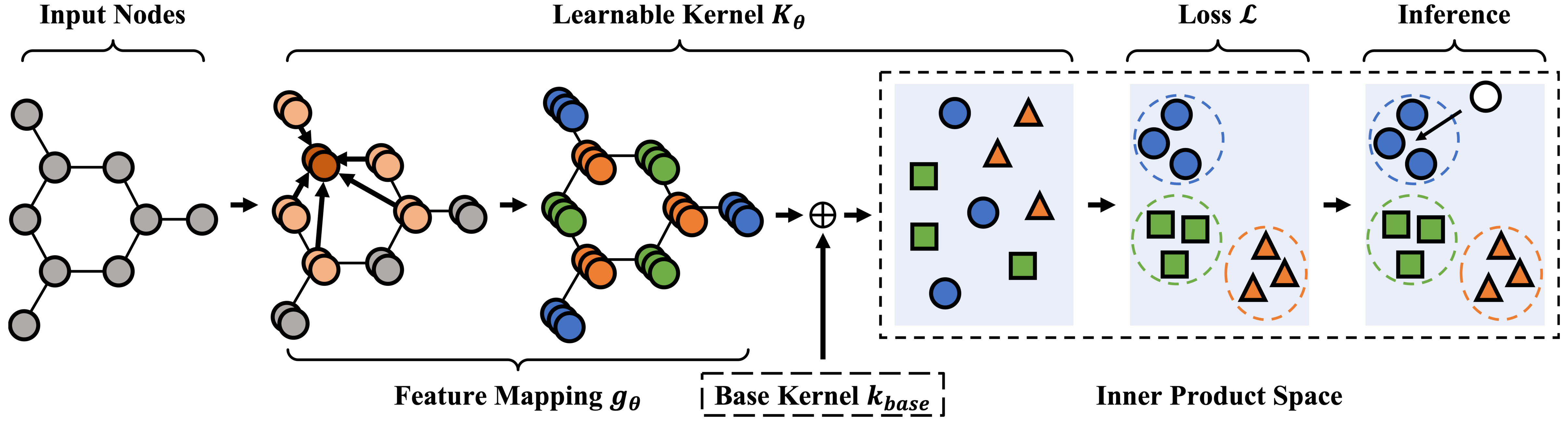
Base Kernel k_{base}:

- Dot product $k_{\langle\cdot,\cdot\rangle}$:

$$k_{base}(x,y) = \langle x, y\rangle$$

- RBF kernel k_{RBF}:

$$k_{base}(x,y) = \exp\left(-\frac{\|x-y\|^2}{2\sigma^2}\right)$$



Similarity Metric s_G and Criteria ℓ:

- We require that the similarity of node pairs with the same label (v_i, v_j) is greater than those with distinct labels (v_i, v_k) by a margin:

$$\mathcal{L}_K = \sum_{(v_i,v_j,v_k)\in\mathcal{T}} \ell\left(K_\theta(v_i,v_j), K_\theta(v_i,v_k)\right)$$

$$\ell\left(K_\theta(v_i,v_j), K_\theta(v_i,v_k)\right) = \left[K_\theta(v_i,v_k) - K_\theta(v_i,v_j) + \alpha\right]_+$$

Inference for Node Classification:

- Nearest Centroid Classifier $\mathcal{C}_K$:

$$y^* = \operatorname*{argmax}_{y\in Y} \mu_y, \mu_y = \frac{1}{|V_y|}\sum_{v_i\in V_y} K_\theta(\mu, v_i)$$

- Softmax Classifier $\mathcal{C}_Y$:

$$\mathcal{L}_Y = -\sum_{v_i\in V} q(y_i)\log\left(\sigma(g_\theta(v_i))\right)$$

Results

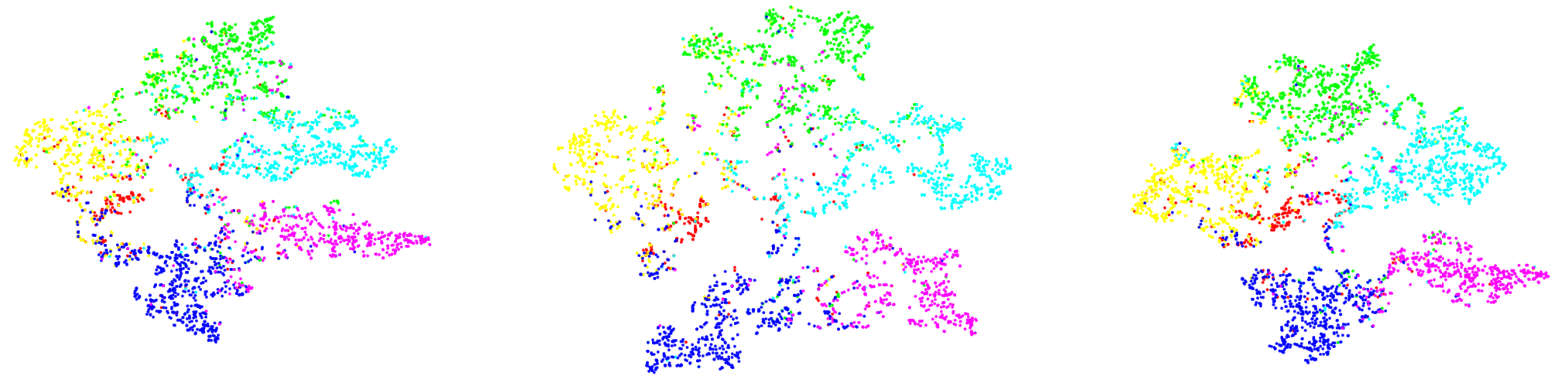
- **Results of Node Classification**

Method	Cora [24]	Citeseer [11]	Pubmed [32]
KLED [9]	82.3	-	82.3
GCN [21]	86.0	77.2	86.5
GAT [40]	85.6	76.9	86.2
FastGCN [5]	85.0	77.6	88.0
$\mathcal{K}_1 = \{k_{\langle\cdot,\cdot\rangle}, g_\theta, \mathcal{L}_K, \mathcal{C}_K\}$	86.68 ± 0.17	77.92 ± 0.25	89.22 ± 0.17
$\mathcal{K}_2 = \{k_{\mathrm{RBF}}, g_\theta, \mathcal{L}_K, \mathcal{C}_K\}$	86.12 ± 0.05	78.68 ± 0.38	89.36 ± 0.21
$\mathcal{K}_3 = \{k_{\langle\cdot,\cdot\rangle}, g_\theta, \mathcal{L}_{K+Y}, \mathcal{C}_Y\}$	**88.40 ± 0.24**	**80.28 ± 0.03**	**89.42 ± 0.01**
$\mathcal{N}_1 = \{g_\theta, \mathcal{L}_Y, \mathcal{C}_Y\}$	87.56 ± 0.14	79.80 ± 0.03	89.24 ± 0.14
$\mathcal{K}_1^* = \{k_{\langle\cdot,\cdot\rangle}, g_{\mathrm{GCN}}, \mathcal{L}_K, \mathcal{C}_K\}$	87.04 ± 0.09	77.12 ± 0.23	87.84 ± 0.12
$\mathcal{K}_2^* = \{k_{\langle\cdot,\cdot\rangle}, g_{\mathrm{GAT}}, \mathcal{L}_K, \mathcal{C}_K\}$	86.10 ± 0.33	77.92 ± 0.19	-

- **Ablation Study on Node Feature Aggregation Schema**

Variants of $\mathcal{K}_3$	Cora [24]	Citeseer [11]	Pubmed [32]
Default	**88.40 ± 0.24**	**80.28 ± 0.03**	89.42 ± 0.01
1-hop	85.56 ± 0.02	77.73 ± 0.02	88.98 ± 0.01
3-hop	88.25 ± 0.01	80.13 ± 0.01	**89.53 ± 0.01**
1-layer	82.60 ± 0.01	77.63 ± 0.01	85.80 ± 0.01
3-layer	86.33 ± 0.04	78.53 ± 0.20	89.46 ± 0.05
$c = 0.25$	69.33 ± 0.09	74.48 ± 0.03	84.68 ± 0.02
$c = 0.50$	76.98 ± 0.10	77.47 ± 0.04	86.45 ± 0.01
$c = 0.75$	84.25 ± 0.01	77.99 ± 0.01	87.45 ± 0.01
$c = 1.00$	87.31 ± 0.01	78.57 ± 0.01	88.68 ± 0.01

- **t-SNE Visualization of Node Embeddings on Citeseer Dataset**



(a) GCN (b) GAT (c) Ours