RetGK: Graph Kernels Based on Return Probabilities of Random Walks

Zhen Zhang, Mianzhi Wang, Yijian Xiang, Yan Huang, and Arye Nehorai
Preston M. Green Department of Electrical and Systems Engineering, Washington University in St. Louis, Missouri

Abstract

Structured data modeled as graphs arise in many application domains, such as computer vision, bioinformatics, and social network mining. One interesting problem for graph-type data is quantifying their similarities based on the connectivity structure and attribute information. Graph kernels, which are positive definite functions on graphs, are powerful similarity measures, in the sense that they make various kernel-based learning algorithms, for example, clustering, classification, and regression, applicable to structured data.

A real-world example:

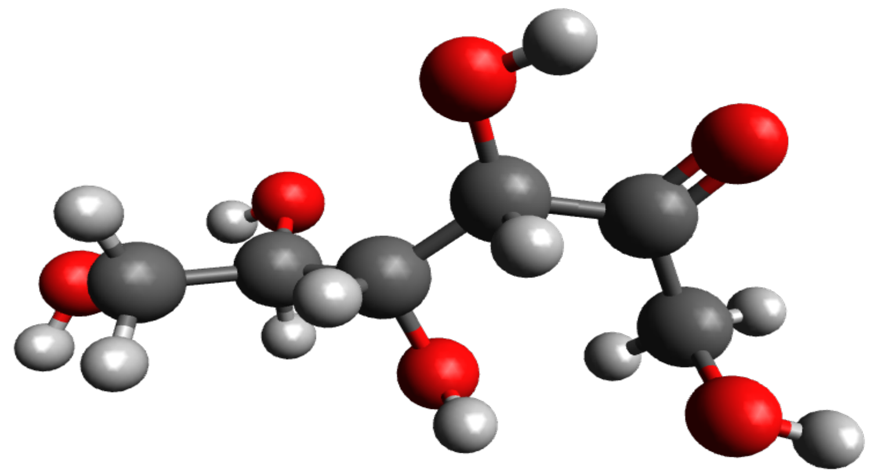

Figure 1: A 3-D model of a Linear D-fructose.

Challenges for designing graph kernels

(i) The graph kernels should satisfy the isomorphism-invariant property, while being informative on the topological structure difference.
(ii) The graph kernels should integrate both graph structure and node attribute information.
(iii) The graph kernels should be scalable to large graphs.

Our solution:

We introduce a new node structural role descriptor, the return probability feature (RPF) of random walks. With the RPF, we can embed attributed graphs into a Hilbert space. After that, we naturally obtain our return probability-based graph kernels ("RetGK"). Employing the approximate feature maps technique, we represent each graph with a multi-dimensional tensor and design a family of computationally efficient graphs kernels.

Return Probability Feature

Given a graph G, let $\boldsymbol{P}_G$ be the transition probability matrix. We assign each node v_i an S-dimensional feature [1], which describes the "structural role" of v_i,

$$\vec{\boldsymbol{p}}_i = [\boldsymbol{P}_G^1(i,i), \boldsymbol{P}_G^2(i,i), ..., \boldsymbol{P}_G^S(i,i)]^T, \tag{1}$$

where $\boldsymbol{P}_G^s(i,i)$, $s = 1, 2, ..., S$, is the return probability of a s-step random walk starting from v_i. Now each graph is represented by a set of feature vectors in $\mathbb{R}^S$: $\mathrm{RPF}_G^S = \{\vec{\boldsymbol{p}}_1, \vec{\boldsymbol{p}}_2, ..., \vec{\boldsymbol{p}}_n\}$.

Properties of Return Probability Feature

(Isomorphism-invariant). Let G and H be two isomorphic graphs. Then, $\forall S = 1, 2, ..., \infty$, $\mathrm{RPF}_G^S = \mathrm{RPF}_H^S$.
(Multi-resolution). $\boldsymbol{P}_G^s(i,i)$ reflects the interaction between node v_i and the subgraph involving v_i.If s increases, the subgraph becomes larger.
(Informativeness). Given two graphs G and$H$ of n nodes, let τ be the permuatation map. Let $\{(\lambda_k, \vec{\boldsymbol{\psi}}_k)\}_{k=1}^n$ and $\{(\mu_k, \vec{\boldsymbol{\varphi}}_k)\}_{k=1}^n$ be eigenpairs of $\boldsymbol{P}_G$ and $\boldsymbol{P}_H$, respectively. Let $\tau : \{1, 2, ..., n\} \to \{1, 2, ..., n\}$ be a permutation map. If $\mathrm{RPF}_G^n = \mathrm{RPF}_H^n$, then,

1. $\mathrm{RPF}_G^S = \mathrm{RPF}_H^S$, $\forall S = n+1, n+2, ..., \infty$;
2. $\{\lambda_1, \lambda_2, ..., \lambda_n\} = \{\mu_1, \mu_2, ..., \mu_n\}$;
3. If the eigenvalues sorted by their magnitudes satisfy: $|\lambda_1| > |\lambda_2| > ... > |\lambda_m| > 0$, $|\lambda_{m+1}| = ... = |\lambda_n| = 0$, then we have that $|\vec{\boldsymbol{\psi}}_k(i)| = |\vec{\boldsymbol{\varphi}}_k(\tau(i))|$, $\forall v_i \in V_G$, $\forall k = 1, 2, ..., m$.

An illustrative example:

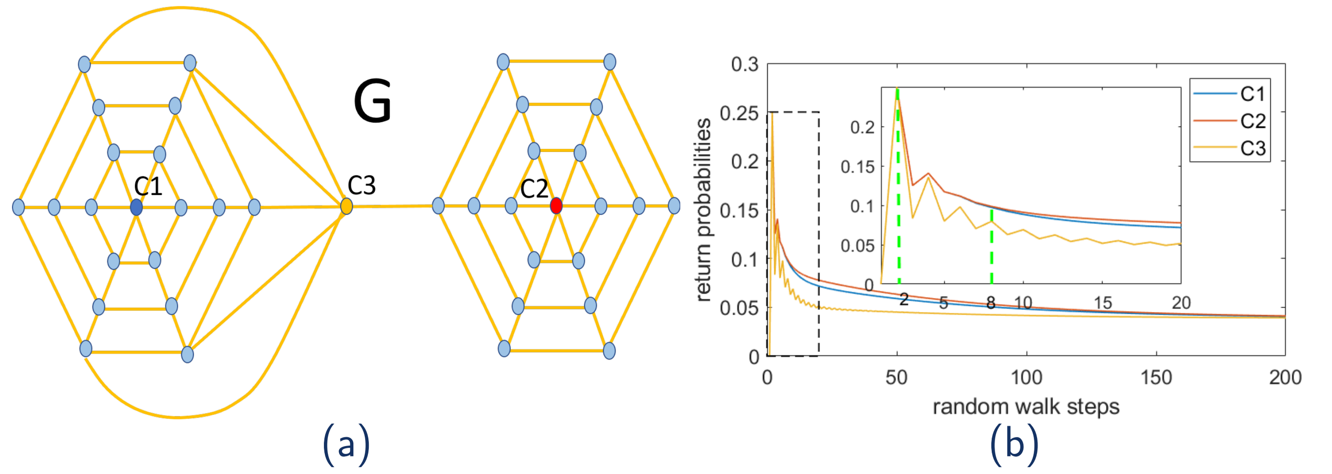

Figure 2: (a) Toy Graph G; (b) The s-step return probability of the nodes C_1, C_2 and C_3 in the toy graph, $s = 1, 2, ..., 200$. The nested figure is a close-up view of the rectangular region.

Observations: (i). Since C_1 and C_2 share the similar neighbourhoods at larger scales, their return probability values are close until the eighth step. Because C_3 plays a very different structural role from C_1 and C_2, its return probabilities values deviate from those of C_1 and C_2 in early steps.
(ii). When the random walk step s approaches infinity, the return probability $\boldsymbol{P}_G^s(i,i)$ will not change much and will converge to the stationary probability.

Hilbert Space Embeddings of Graphs.

Graphs may have many types of attributes, which can be obtained by physical measurements. Let $\mathcal{A}_1, \mathcal{A}_2, ..., \mathcal{A}_L$ be their attribute domains. When combined with RPF, an attributed graph can be represented by the set $\{(\vec{\boldsymbol{p}}_i, a_i^1, ..., a_i^L)\}_{i=1}^n \subseteq \mathcal{A}_0 \times \mathcal{A}_1 \times ... \times \mathcal{A}_L$, where $\mathcal{A}_0 := \mathbb{R}^S$ is the RPF domain. The set representation forms an empirical distribution $\mu = \frac{1}{n}\sum_{i=1}^n \delta_{(\vec{\boldsymbol{p}}_i, a_i^1, ..., a_i^L)}$ on $\mathcal{A} = \times_{l=0}^L \mathcal{A}_l$, which can be embedded into a reproducing kernel Hilbert space (RKHS) by kernel mean embedding [2].
Let k_l, $l = 0, 1, ..., L$ be a kernel on $\mathcal{A}_l$. Then we can define a kernel on $\mathcal{A}$ through the tensor product of kernels, i.e., $k = \otimes_{l=0}^L k_l$, $k\big[(\vec{\boldsymbol{p}}, a^1, a^2, ..., a^L), (\vec{\boldsymbol{q}}, b^1, b^2, ..., b^L)\big] = k_0(\vec{\boldsymbol{p}}, \vec{\boldsymbol{q}}) \prod_{l=1}^L k_l(a^l, b^l)$. Therefore, given a graph G, we can embed it into a Hilbert space reproduced by k,

$$G \to \mu_G \to m_G, \text{ and } m_G = \int_{\mathcal{A}} \phi \, \mathrm{d}\mu_G = \frac{1}{n}\sum_{i=1}^n \phi(\boldsymbol{p}_i, a_i^1, ..., a_i^L). \tag{2}$$

Graph Kernels(I). Given two graphs G and H, let m_G and m_H be the corresponding embeddings. Then, the following functions are positive definite graph kernels.

$$K_1(G,H) = (c + \langle m_G, m_H \rangle_{\mathcal{H}})^d, c \geq 0, d \in \mathbb{N}, \tag{3a}$$

$$K_2(G,H) = \exp(-\gamma \|m_G - m_H\|_{\mathcal{H}}^p), \gamma > 0, 0 < p \leq 2, \tag{3b}$$

where $\langle m_G, m_H \rangle_{\mathcal{H}} = \frac{1}{n_G n_H} \vec{\mathbf{1}}_{n_G}^T \boldsymbol{K}_{GH} \vec{\mathbf{1}}_{n_H}$, $\mathrm{MMD}(\mu_G, \mu_H) = (\frac{1}{n_G^2} \vec{\mathbf{1}}_{n_G}^T \boldsymbol{K}_{GG} \vec{\mathbf{1}}_{n_G} + \frac{1}{n_H^2} \vec{\mathbf{1}}_{n_H}^T \boldsymbol{K}_{HH} \vec{\mathbf{1}}_{n_H} - \frac{2}{n_G n_H} \vec{\mathbf{1}}_{n_G}^T \boldsymbol{K}_{GH} \vec{\mathbf{1}}_{n_H})^{\frac{1}{2}}$ (maximum mean discrepancy), and $\boldsymbol{K}_{GG}$, $\boldsymbol{K}_{GH}$, and $\boldsymbol{K}_{HH}$ are kernel matrices induced by k.

Approximate Hilbert Space Embeddings of Graphs.

To accelerate the speed of computing graph kernel, we employ the approximate explicit feature maps [3]. For a kernel k_l on the attribute domain $\mathcal{A}_l$, $l = 0, 1, ..., L$, we find an explicit map $\hat{\phi} : \mathcal{A}_l \to \mathbb{R}^{D_l}$, so that

$$\forall a, b \in \mathcal{A}_l, \langle \hat{\phi}(a), \hat{\phi}(b) \rangle = \hat{k}_l(a,b), \text{ and } \hat{k}_l(a,b) \to k_l(a,b) \text{ as } D_l \to \infty. \tag{4}$$

Tensor Representation of Attributed Graphs.

Let G be an attributed graph, and let $\{(\vec{\boldsymbol{p}}_i, a_i^1, a_i^2, ..., a_i^L)\}_{i=1}^{n_G}$ be its set representation. Then the approximate explicit graph embeddings, $\hat{m}_G$ is a tensor in $\mathbb{R}^{D_0 \times D_1 \times ... \times D_L}$, and can be written as

$$\hat{m}_G = \frac{1}{n_G}\sum_{i=1}^{n_G} \hat{\phi}_0(\vec{\boldsymbol{p}}_i) \circ \hat{\phi}_1(a_i^1) \circ ... \circ \hat{\phi}_L(a_i^L). \tag{5}$$

Graph Kernels(II). The following functions are positive definite graph kernels defined on $\mathcal{G} \times \mathcal{G}$.

$$\hat{K}_1(G,H) = \big[c + \mathrm{vec}(\hat{m}_{\mathcal{G}})^T \mathrm{vec}(\hat{m}_{\mathcal{H}})\big]^d, c \geq 0, d \in \mathbb{N}, \tag{6a}$$

$$\hat{K}_2(G,H) = \exp(-\gamma \|\mathrm{vec}(\hat{m}_G) - \mathrm{vec}(\hat{m}_H)\|_2^p), \gamma > 0, 0 < p \leq 2.. \tag{6b}$$

Experimental Results

Table 1: Classification results (in %) for non-attributed graph datasets

Datasets	WL	GK	DGK	PSCN	RetGK$_{\mathrm{I}}$	RetGK$_{\mathrm{II}}$
COLLAB	74.8(0.2)	72.8(0.3)	73.1(0.3)	72.6(2.2)	**81.0(0.3)**	80.6(0.3)
IMDB-BINARY	70.8(0.5)	65.9(1.0)	67.0(0.6)	71.0(2.3)	71.9(1.0)	**72.3(0.6)**
IMDB-MULTI	**49.8(0.5)**	43.9(0.4)	44.6(0.5)	45.2(2.8)	47.7(0.3)	48.7(0.6)
REDDIT-BINARY	68.2(0.2)	77.3(0.2)	78.0(0.4)	86.3(1.6)	**92.6(0.3)**	91.6(0.2)
REDDIT-MULTI(5K)	51.2(0.3)	41.0(0.2)	41.3(0.2)	49.1(0.7)	**56.1(0.5)**	55.3(0.3)
REDDIT-MULTI(12K)	32.6(0.3)	31.8(0.1)	32.2(0.1)	41.3(0.4)	**48.7(0.2)**	47.1(0.3)
Total time	2h3m	–	–	–	48h14m	17m14s

Table 2: Classification results (in %) for graph datasets with discrete attributes

Datasets	WL	CSM	DGCNN	DGK	PSCN	RetGK$_{\mathrm{I}}$	RetGK$_{\mathrm{II}}$
ENZYMES	53.4(0.9)	**60.4(1.6)**	–	53.4(0.9)	–	**60.4(0.8)**	59.1(1.1)
PROTEINS	71.2(0.8)	–	75.5(0.9)	75.7(0.5)	75.0(2.5)	**75.8(0.6)**	75.2(0.3)
MUTAG	84.4(1.5)	85.4(1.2)	85.8(1.7)	87.4(2.7)	89.0(4.4)	**90.3(1.1)**	90.1(1.0)
DD	78.6(0.4)	–	79.4(0.9)	–	76.2(2.6)	**81.6(0.3)**	81.0(0.5)
NCI1	**85.4(0.3)**	–	74.4(0.5)	80.3(0.5)	76.3(1.7)	84.5(0.2)	83.5(0.2)
PTC-FM	55.2(2.3)	63.8(1.0)	–	–	–	62.3(1.0)	**63.9(1.3)**
PTC-FR	63.9(1.4)	65.5(1.4)	–	–	–	66.7(1.4)	**67.8(1.1)**
PTC-MM	60.6(1.1)	63.3(1.7)	–	–	–	65.6(1.1)	**67.9(1.4)**
PTC-MR	55.4(1.5)	58.1(1.6)	58.6(2.5)	60.1(2.6)	62.3(5.7)	**62.5(1.6)**	62.1(1.5)
Total time	2m27s	–	–	–	–	38m4s	**49.9s**

Table 3: Classification results (in %) for graph datasets with both discrete and continuous attributes

Datasets	GIK	CSM	RetGK$_{\mathrm{I}}$	RetGK$_{\mathrm{II}}$
ENZYMES	71.7(0.8)	69.8(0.7)	**72.2(0.8)**	70.6(0.7)
PROTEINS	76.1(0.3)	–	**78.0(0.3)**	77.3(0.5)
BZR	–	79.4(1.2)	86.4(1.2)	**87.1(0.7)**
COX2	–	74.4(1.7)	80.1(0.9)	**81.4(0.6)**
DHFR	–	79.9(1.1)	81.5(0.9)	**82.5(0.8)**
Total time	–	–	4m17s	**2m51s**

Observation: In most cases, our graph kernel RetGK$_{\mathrm{II}}$ outperforms the state-of-the-art methods in both classification accuracy and computational efficiency.

References

[1] Zhang, Zhen, Mianzhi Wang, Yijian Xiang, Yan Huang, and Arye Nehorai, "RetGK: Graph Kernels based on return probabilities of random walks," to appear in *NIPS 2018*.
[2] Arthur Gretton, Karsten M. Borgwardt, Malte J. Rasch, Bernhard Schölkopf, Alexander Smola, "A kernel two sample test," *JMLR 2012*.
[3] Ali Rahimi and Ben Recht, "Random features for large-scale kernel machines," *NIPS 2007*.