Graph Kernels

Viveka Kulharia
Arnab Ghosh

Kind of Graphs

1. Directed or undirected
2. Fully connected
3. Edges have weights
4. Modeled from Molecules and Proteins
5. If $E = O(V)$, then the graph is sparse

Graph Kernel

$$k(G, G') = \sum_{k=0}^{\infty} \mu(k) q_x^T W_x^k p_x$$

where,

$$W_x = \phi(X) \otimes \phi(X')$$

$$q_x = q \otimes q'$$

$$p_x = p \otimes p'$$

Lemma 12

If $A \in X^{n \times m}$, $B \in \mathbb{R}^{m \times p}$, and $C \in X^{p \times q}$, then

$$vec(\phi(A)B\phi(C)) = (\phi(C)^T \otimes \phi(A))vec(B) \in \mathbb{R}^{nq \times 1}$$

LHS requires $O(n^3 d)$ computations (Working in feature space)

RHS requires $O(n^4)$ kernel computation

So, LHS is faster.

Proof of p.s.d.

It’s a p.s.d. kernel if coefficients $\mu(k)$ are such that it converges.

The proof heavily uses lemma 12.

We can write it as summation over some coeff times the terms $p_k(G)^T p_k(G')$

Each of these terms are p.s.d. kernels and non-negative linear combination over p.s.d. kernels which has pointwise limit is also p.s.d. kernel.

It’s a generalized kernel

1. When $\mu(k) = \lambda^k$ then

$$\sum_{k=0}^{\infty} \lambda^k q_x^T W_x^k p_x = q_x^T (I - \lambda W_x)^{-1} p_x$$

2. When $\mu(k) = \lambda_k$ and $p_i = q_i = 1/n$ with linear feature map,

$$k(G, G') = \frac{1}{n^2 n'^2} \sum_{i=1}^{n} \sum_{j=1}^{n'} \sum_{k=0}^{\infty} \lambda_k [\tilde{A}_x^k]_{ij}$$

It’s a generalized kernel

3. In the previous Kernel by Gartner et al. (2003), we can replace $\lambda_k = \frac{\lambda^k}{k!}$

   to get exponential kernel defined as:

$$k(G, G') = \frac{1}{n^2 n'^2} \sum_{i=1}^{n} \sum_{j=1}^{n'} [e^{\lambda \tilde{A}_x}]_{ij} = \mathbf{e}^T e^{\lambda \tilde{A}_x} \mathbf{e}$$

Efficient computation

1. Sylvester Equation Methods:

$$M = \sum_{i=1}^{d} \lambda({}^{i}A')M({}^{i}A^{T}) + M_0$$

$$(I - \lambda \sum_{i=1}^{d} ({}^{i}A) \otimes ({}^{i}A'))vec(M) = vec(M_0) = p_x$$

$$q_x^{T} vec(M) = q_x^{T} (I - \lambda W_x)^{-1} p_x$$

Efficient computation

2. Conjugate Gradient Methods

$$(I - \lambda W_x)x = p_x$$

$$W_x y = (\phi(X) \otimes \phi(X'))vec(Y) = vec(\phi(X')Y\phi(X)^T)$$

Efficient computation

3. Fixed-Point Iterations

$$x_{t+1} = p_x + \lambda W_x x_t$$

$$k = \bigcirc(\frac{ln\epsilon}{ln\lambda + ln|\xi_1|})$$

Efficient computation

4. Spectral Decomposition Method

$$k(G,G') = \sum_{k=0}^{\infty} \mu(k) q_x^T (P_x D_x P_x^{-1})^k p_x = q_x^T P_x \left(\sum_{k=0}^{\infty} \mu(k) D_x^k\right) P_x^{-1} p_x$$

$$k(G,G') = (q_i^T P_i \otimes q_j^T P_j)\left(\sum_{k=0}^{\infty} \mu(k) (D_i \otimes D_j)^k\right)(P_i^{-1} p_i \otimes P_j^{-1} p_j)$$

Efficient computation

5. Nearest Kronecker Product Approximation

$$W_x \approx S \otimes T$$

$$\min_{S,T} \ \mathrm{norm}||W_x - S \otimes T||_F$$

$$k' = \bigcirc(\frac{\ln\ n}{ln|\xi_1| - ln|\xi_2|})$$

Shortest Path Kernels

- Floyd Transformation of the graph into a shortest path graph
- There exists an Edge in the Floyd Transformed graph if there exists a walk between the 2 vertices and the label of the graph is the shortest path between the 2 vertices
- The shortest path between all pairs of vertices can be found using Floyd Warshall Algorithm

Shortest Path Kernel

$$k_{shortestpaths}(S_1, S_2) = \sum_{e_1 \in E_1} \sum_{e_2 \in E_2} k^{(1)}_{walk}(e_1, e_2)$$

where $k^{(1)}_{walk}(e_1, e_2)$ is any positive definite kernel function on the edge walks of length 1.

- In our implementation we have used k(v1,v2)=1 and k(e1,e2)=1 which was also used by the authors in their experiments

Shortest Path Kernel is psd

- Choose a psd Kernel on nodes and a psd kernel on edges
- Define a kernel on pairs of walks of length 1 as the product of kernels on nodes and edges encountered along the walk
- Since Kernels are closed under product then it is also closed under Tensor Product
- As a tensor product of node and edge kernels the kernel is positive semidefinite
- Aliter : The positive definiteness of the shortest path kernel follows directly from its definition as a convolution kernel proven to be positive semidefinite by D Haussler in Convolutional Kernels on Discrete Structures

Experiments

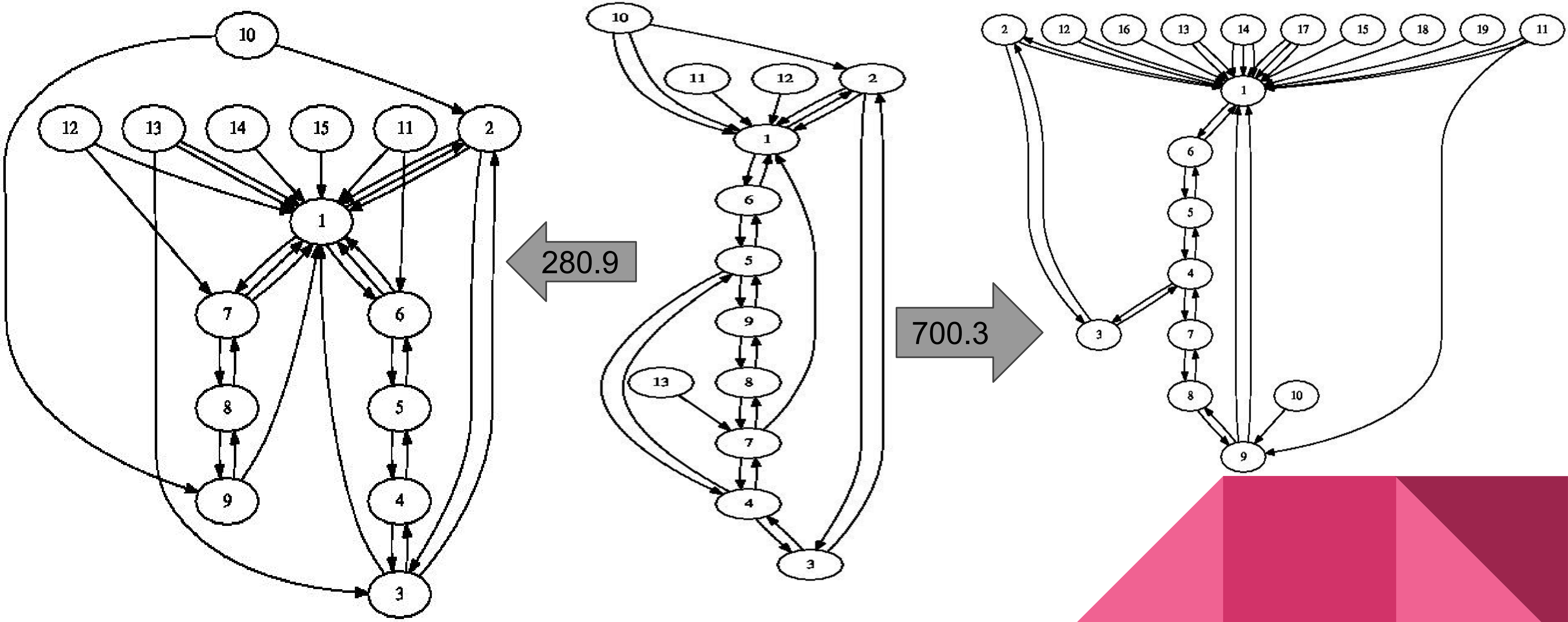
Data sets:

Dataset	Avg. no. of Nodes/Graph	Avg. no. of Edges/Graph
MUTAG	17.72	38.76
PTC	26.70	52.06
E (Enzyme)	32.63	124.27
PROTEIN	38.57	143.75

Time Statistics

Dataset	Sylvester Equations	Conjugate Gradient	Shortest Path	Product Graph
MUTAG	7.7279s	5.6300 s	5.7280 s	17.3661 s
PTC	15.2994 s	9.2300 s	22.5813 s	28.9442 s
E	19.0723 s	10.8200 s	27.1933 s	100.49 s
PROTEINS	67.0005	36.1900 s	166.949 s	498.35 s

Graphic View of the Random Walk Kernel



280.9

700.3

Classification Results from Borgwardt 2005

Kernel Type	Accuracy
2 shortest paths	94.44
shortest paths	93.33
Walks upto Length 4	89.63
Walks upto Length 5	88.89
Walks upto Length 6	88.15
walks up to length 7	87.96