Proof that Random walk kernel is psd kernel

http://www.cse.iitb.ac.in/~cs717/notes/classNotes/kernels/Graphs.pdf

$\Rightarrow$ Slide 10 has statement of psd

$$K(G_1, G_2) = \frac{1}{|G_1||G_2|} \sum_{k=1}^{\infty} \lambda^k p_\otimes^T A_\otimes^k q_\otimes$$

$$= \frac{1}{|G_1||G_2|} \sum_{k=1}^{\infty} \lambda^k (p_1 \otimes p_2)^T (A_1 \otimes A_2)^k (q_1 \otimes q_2)$$

$$= \frac{1}{|G_1||G_2|} \sum_{k=1}^{\infty} \lambda^k (p_1 A_1^k q_1) \otimes (p_2 A_2^k q_2)$$

Scalars

$$= \frac{1}{|G_1||G_2|} \sum_{k=1}^{\infty} \lambda^k (p_1 A_1^k q_1)(p_2 A_2^k q_2)$$

(Option 1) $= \frac{1}{|G_1||G_2|} \langle \phi(G_1), \phi(G_2) \rangle$

Where

$\phi(G_1) \in$ Hilbert

$$\phi(G_1) = \begin{bmatrix} p_1 A_1 q_1 \\ p_1 A_1^2 q_1 \\ \vdots \\ p_1 A_1^k q_1 \\ \vdots \end{bmatrix}$$

$$\langle \phi_1, \phi_2 \rangle = \sum_{k=1}^{\infty} \lambda^k \phi_1[k] \phi_2[k]$$

Can prove in our choice of ϕ that $\langle \ \rangle$ is a valid inner prod

Option 2 ↓

Consider

$$K \quad \begin{bmatrix} k(G_1, G_1) & \cdots & k(G_1, G_n) \\ \vdots & & \vdots \\ k(G_n, G_1) & \cdots & k(G_n, G_n) \end{bmatrix}$$

$v^T K v \geq 0 \quad \forall v$ (H/W)

$1 + x + x^2 + \ldots = \dfrac{1}{1-x} \quad x \in [0, 1)$

Replace x by $\lambda A_{\otimes}$ subject to $\lambda < 1$ & $A_{\otimes}$ is stochastic & positive

(you can have weaker conditions)

Similarly:

$$1 + \frac{x}{\lfloor 1} + \frac{x^2}{\lfloor 2} \ldots = e^x$$

http://www.cs.ucsb.edu/~xyan/tutorial/KDD08_graph_partII.pdf ⇒ We discuss other problems & kernels from here

Tottering & Halting

$$K(G_1, G_2) = \sum_{k=1}^{\infty} \lambda^k q_{\times}^T A_{\times}^k P_{\times}$$

Cycles leading to too many hops to same node

Cycles are however less of a problem than the fact that algo might tend to hop back to source node at an alarmingly fast rate

Solve by setting $\lambda \to 0$

Too small a value of λ means larger walks ignored

Set $\lambda \to 1$ or change to $\lambda^{\log k}$

$\lambda^{\log k}$ will not give convergence

H/W:

Consider shortest path kernel

$$k(G_1, G_2) = \sum_{v_1, v_2 \in G_1} \sum_{v_1', v_2' \in G_2} k(v_1', v_2', v_1, v_2)$$

Where

$$k(v_1 v_2, v_1', v_2') = k_{length}(d(v_1, v_2), d(v_1', v_2'))$$

- When k_{length} is linear, k is product of weiner indices

H/W:

$$\boxed{\text{Can you efficiently compute } k \text{ when } k_{length} = 1 \text{ if } d(v_1, v_2) = d(v_1', v_2'),\ = 0 \text{ o/w}}$$

(Sort merge $= O(n^3)$)

Klog:

http://www.cse.iitb.ac.in/~cs717/notes/classNotes/kernels/paoloKLogSlides.pdf