

Weisfeiler-Lehman Graph Kernel (JMLR 2011) and Neighborhood Hash Graph Kernel (ICDM 2009)

Presenter: Jose Lugo
Pedja's Lab Meeting

October 12, 2011



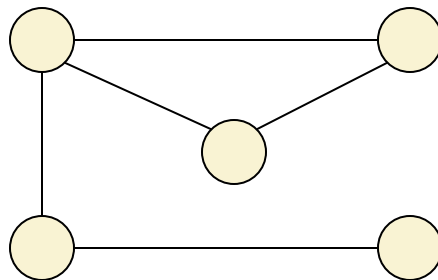


Learning on Graphs

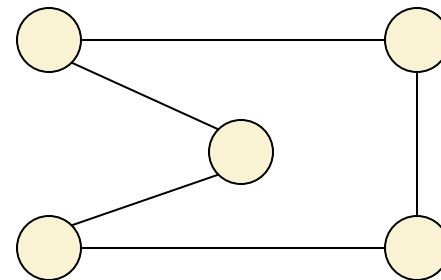
- **Application domains**
 - Bioinformatics, Cheminformatics, WWW link, Social networks
- **Motivation:** Study relationships between structured objects (graphs)

Graph Comparison Problem

G



G'





Graph Kernels

- Define kernels on pair of graphs

$$k(G, G') = \langle \Phi(G), \Phi(G') \rangle$$

$k(G, G')$ – measure of similarity between G and G'

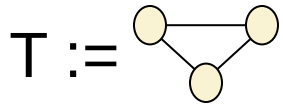
- Kernel Matrix K , where $K_{ij} = k(G_i, G_j)$ for $1 \leq i, j \leq n$

– Properties of K

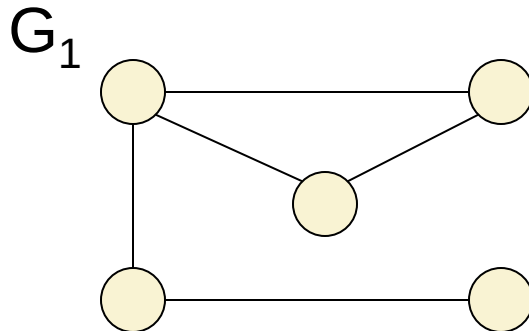
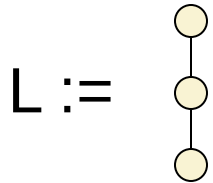
- I. Symmetric
- II. Positive semi-definite



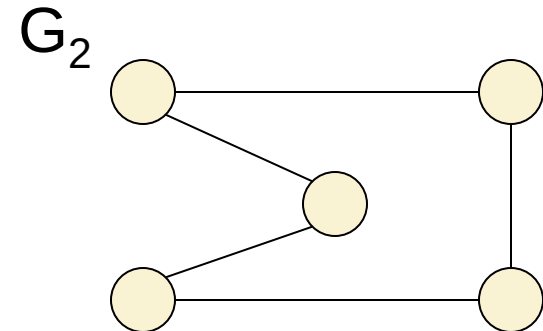
Graph Kernel Example



$$\Phi(G) = (\#(T), \#(L))$$



$$\Phi(G_1) = (1, 3)$$

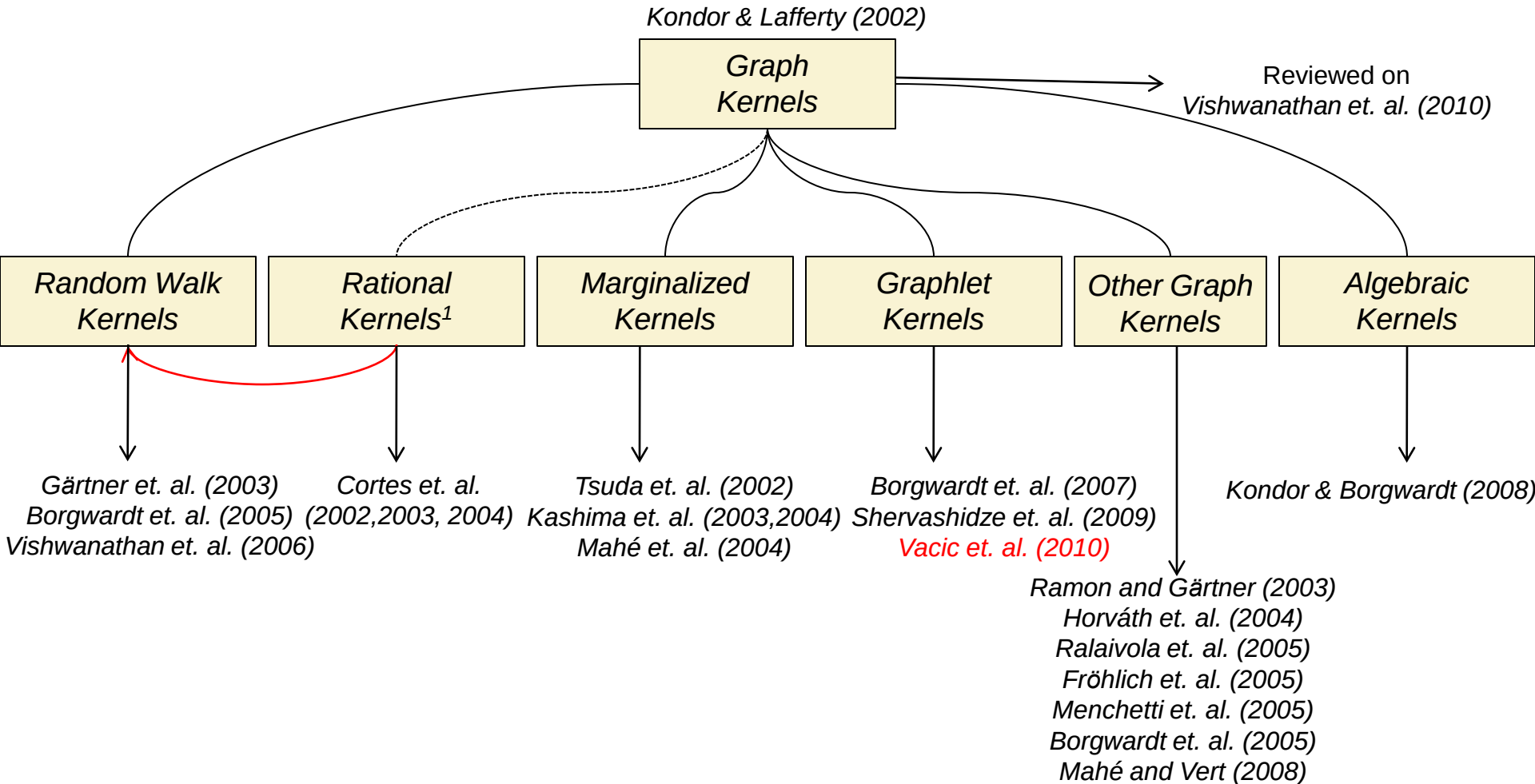


$$\Phi(G_2) = (0, 5)$$

$$k(G_1, G_2) = \langle \Phi(G_1), \Phi(G_2) \rangle = 15$$



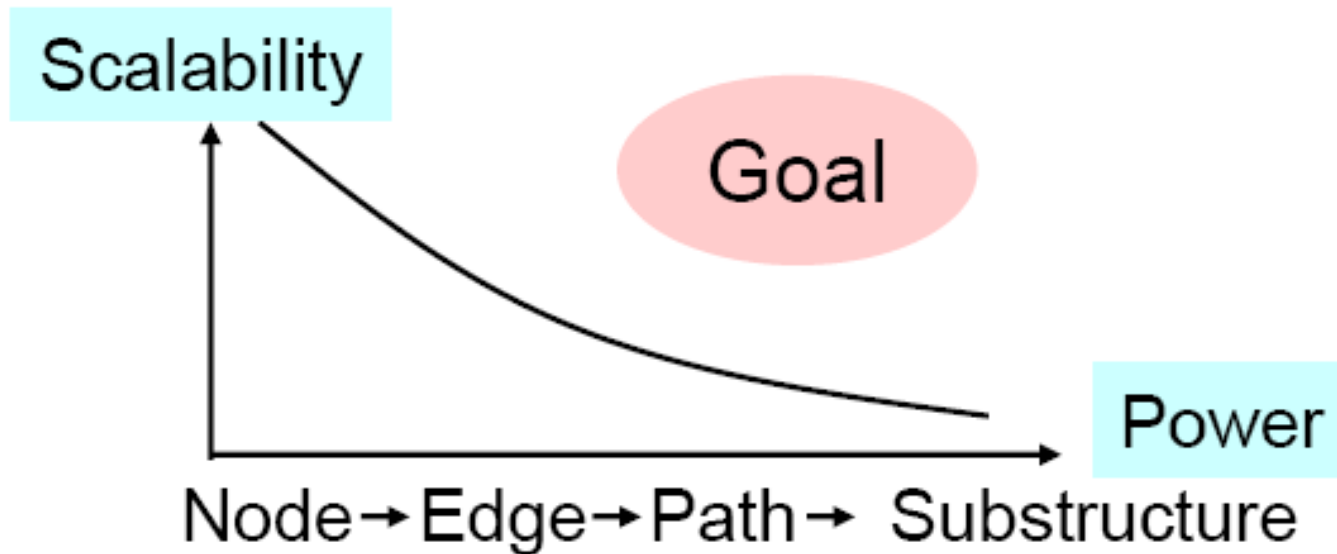
Graph Kernels Research Efforts



1. Certain Rational Kernels when specialized to graphs reduced to Random Walk Graph Kernel (*Vishwanathan et. al. 2010*)



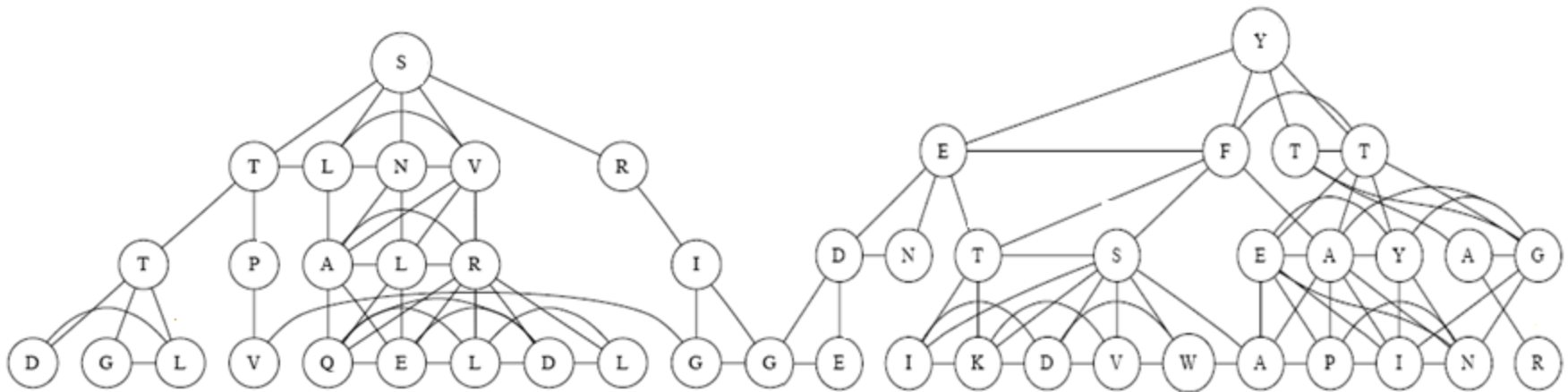
Question: How to scale up graph kernels to large, labeled graphs?





Graph Kernels on (Large) Labeled Graphs

$$G = (V, E, f), |V| = n, |E| = m$$



$$f: V \rightarrow \Sigma = \{A, R, N, D, C, E, Q, G, H, I, L, K, M, F, P, S, T, W, T, Y, V\}$$

or

$$f: V \rightarrow \Sigma = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 19, 20\}$$



Weisfeiler-Lehman Graphlet Kernel

Shervashidze et. al. (2010)

- Weisfeiler-Lehman test of isomorphism (1968)

Algorithm: Perform the following three steps h times:

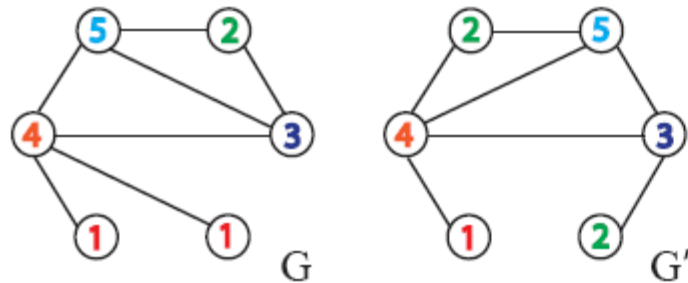
1. **Sorting:** Represent each node v as a sorted list L_v of its neighbours ($O(m)$)
2. **Compression:** Compress this list into a **hash value** $h(L_v)$ ($O(m)$)
3. **Relabeling:** Relabel v with $h(L_v)$ as its new node label ($O(n)$)

- Define *Weisfeiler-Lehman* graph kernels:
 - $k_{\text{WL}}(G, G')$, $k_{\text{WLsubtree}}(G, G')$ and $k_{\text{WLshortestpath}}(G, G')$



INDIANA UNIVERSITY

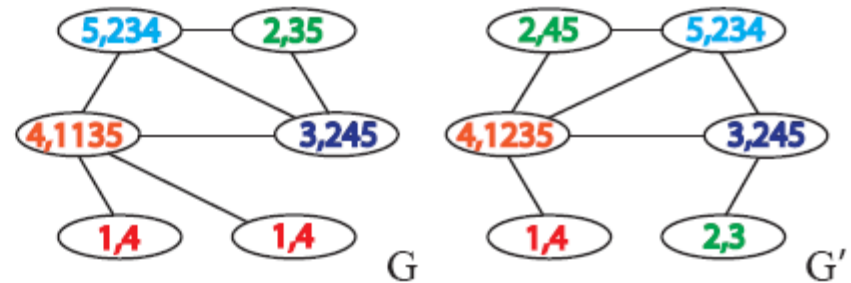
Given labeled graphs G and G'



a

1st iteration

Result of steps 1 and 2: multiset-label determination and sorting



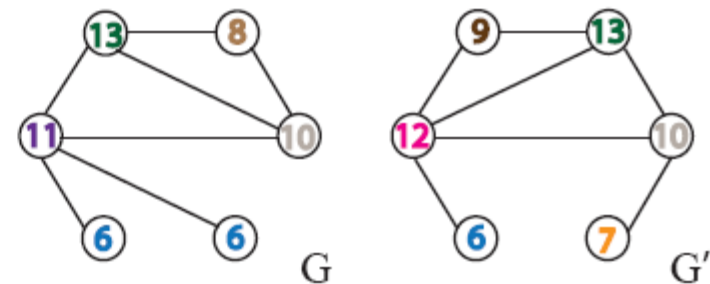
b

1st iteration
Result of step 3: label compression

1,4	→	6	3,245	→	10
2,3	→	7	4,1135	→	11
2,35	→	8	4,1235	→	12
2,45	→	9	5,234	→	13

c

1st iteration
Result of step 4: relabeling

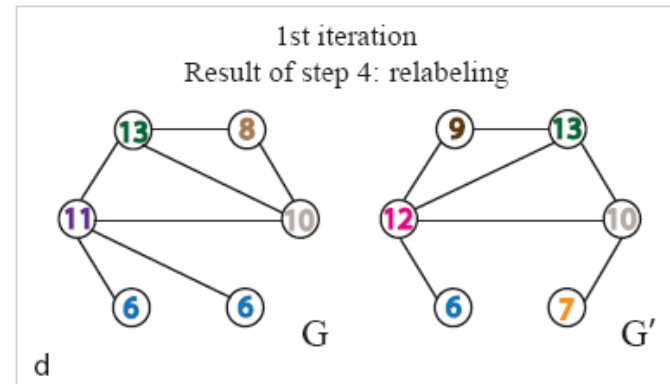
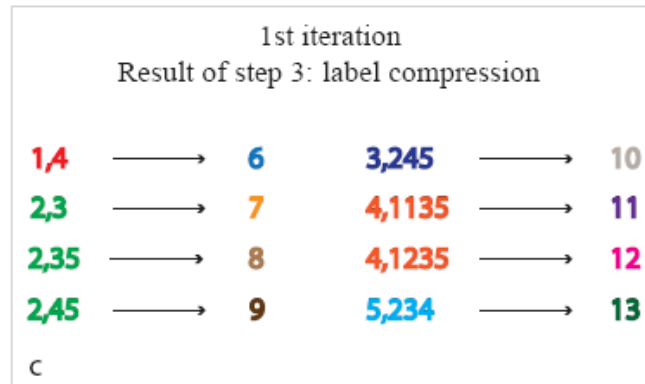
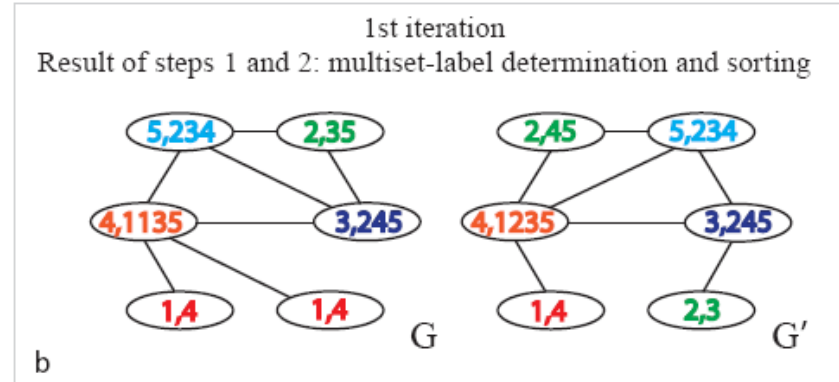
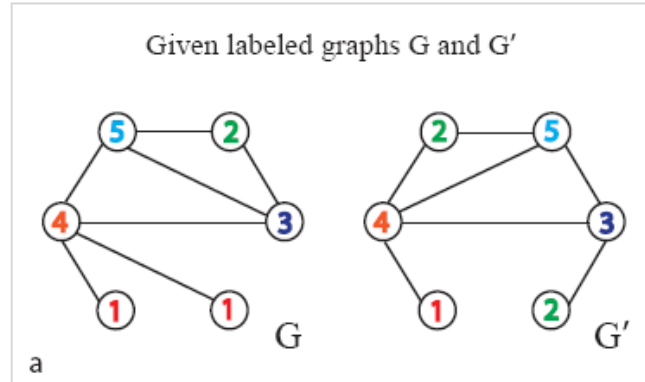


d

Observed that compressed labels $l_i(v)$ correspond to subtree patterns of height i rooted at v

Example

$$k_{WL}^{(h)}(G, G') = k(G_0, G'_0) + k(G_1, G'_1) + \dots + k(G_h, G'_h),$$



$$\Phi(G_0) = (2, 1, 1, 1, 1)$$

$$\Phi(G'_0) = (1, 2, 1, 1, 1)$$

$$k(G_0, G'_0) = \langle \Phi(G_0), \Phi(G'_0) \rangle = 7$$

$$\Phi(G_1) = (2, 0, 1, 0, 1, 1, 0, 1)$$

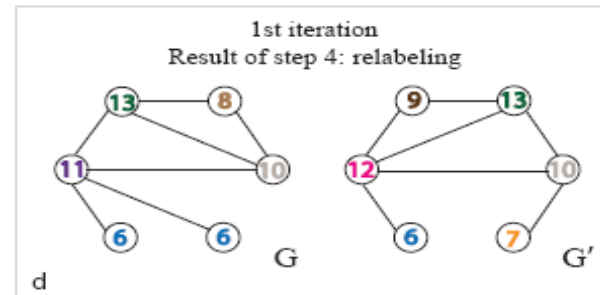
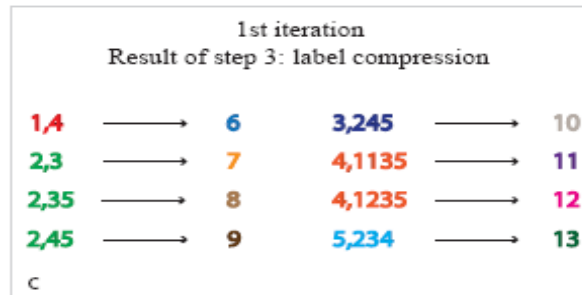
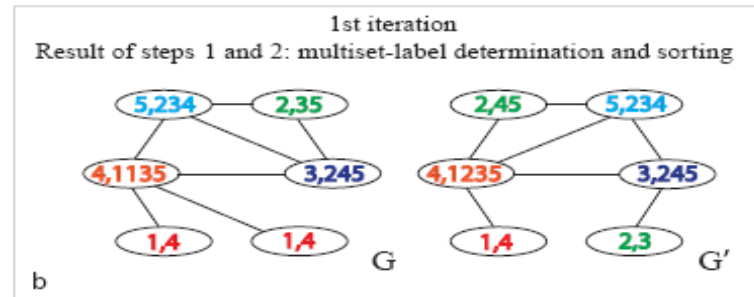
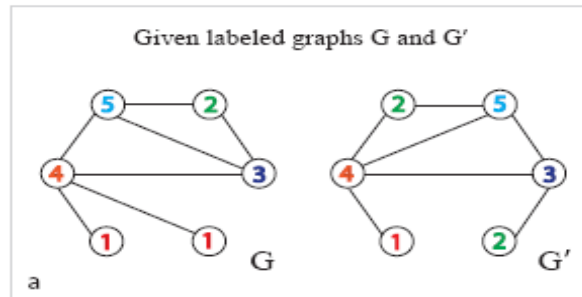
$$\Phi(G'_1) = (1, 1, 0, 1, 1, 0, 1, 1)$$

$$k(G_1, G'_1) = \langle \Phi(G_1), \Phi(G'_1) \rangle = 4$$



Example

$$k_{WLsubtree}^{(h)}(G, G') = \langle \phi_{WLsubtree}^{(h)}(G), \phi_{WLsubtree}^{(h)}(G') \rangle$$



End of the 1st iteration
Feature vector representations of G and G'

$$\phi_{WLsubtree}^{(1)}(G) = (\textcolor{red}{2}, \textcolor{green}{1}, \textcolor{blue}{1}, \textcolor{red}{1}, \textcolor{blue}{1}, \textcolor{blue}{2}, \textcolor{orange}{0}, \textcolor{red}{1}, \textcolor{orange}{0}, \textcolor{blue}{1}, \textcolor{blue}{1}, \textcolor{orange}{0}, \textcolor{blue}{1})$$

$$\phi_{WLsubtree}^{(1)}(G') = (\textcolor{red}{1}, \textcolor{green}{2}, \textcolor{blue}{1}, \textcolor{red}{1}, \textcolor{blue}{1}, \textcolor{blue}{1}, \textcolor{orange}{1}, \textcolor{orange}{0}, \textcolor{red}{1}, \textcolor{blue}{1}, \textcolor{orange}{0}, \textcolor{blue}{1}, \textcolor{blue}{1})$$

Counts of
original
node labels

Counts of
compressed
node labels

$$k_{WLsubtree}^{(1)}(G, G') = \langle \phi_{WLsubtree}^{(1)}(G), \phi_{WLsubtree}^{(1)}(G') \rangle = 11.$$

e



Neighborhood Hash Graphlet Kernel

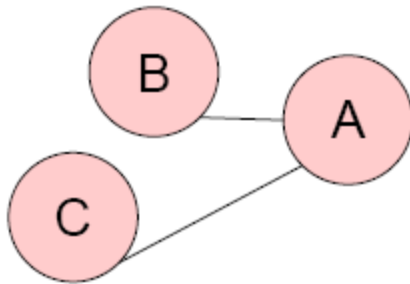
Hido and Kashima (2009)

- Bit-represented node label
- Logical operations
- Neighborhood hash over nodes
- Define *neighborhood hash* graph kernel, $k_{\text{NH}}(G, G')$
- Linear time complexity with # of edges

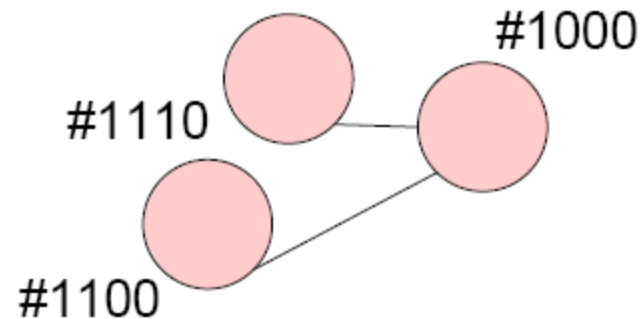


Bit-represented Node Label

Original graph



Replaced with 4-bit labels





Logical Operations on Bit Labels

- XOR (s_i, s_j)
 - Exclusive OR
 - Order-independent
- ROT_k
 - k-bit rotation
 - move left most k-bits to the right

Table I
PROPERTY OF BITWISE XOR

(a)	$s \oplus s = s_{zero}$
(b)	$s \oplus s_{zero} = s$
(c)	$s \oplus s' = s' \oplus s$
(d)	$(s \oplus s') \oplus s'' = s \oplus (s' \oplus s'')$
(e)	$s' \oplus s' \oplus s = s$

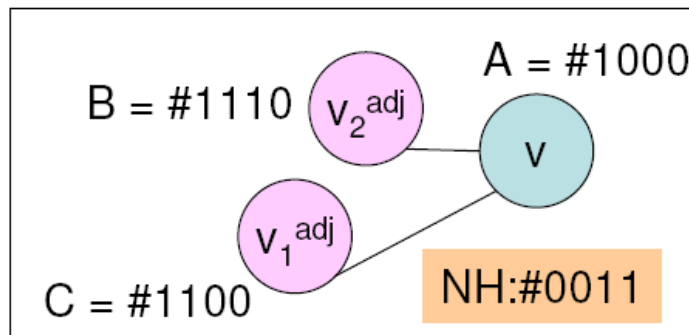
Table II
EXAMPLES OF XOR AND ROT

(a)	$\#0101 \oplus \#1000 = \#1101$
(b)	$\#1101 \oplus \#1100 = \#0001$
(c)	$\text{ROT}_1(\#0001) = \#0010$
(d)	$\text{ROT}_2(\#0110) = \#1001$

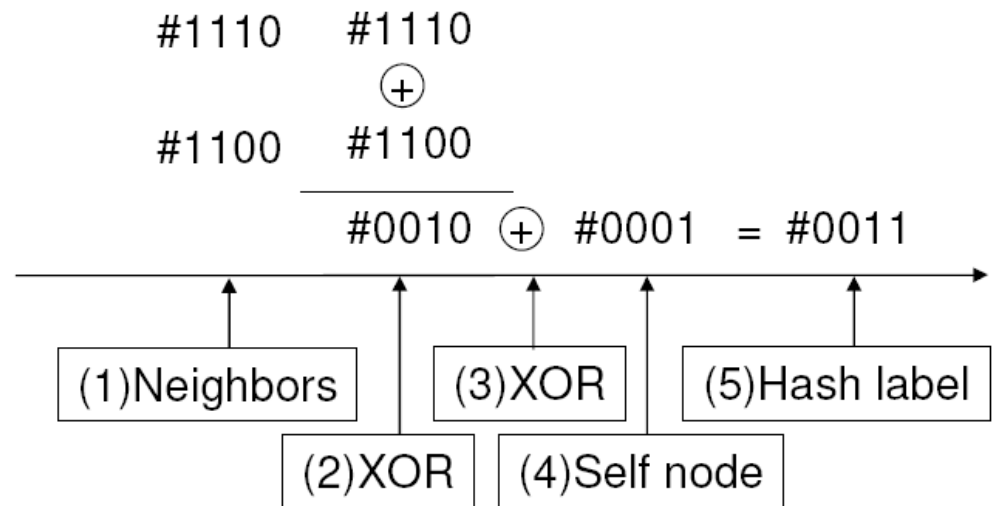


Neighborhood Hash over a Node, $NH(v)$

$$NH(v) = ROT_1(\ell(v)) \oplus (\ell(v_1^{adj}) \oplus \dots \oplus \ell(v_d^{adj})).$$



(a)



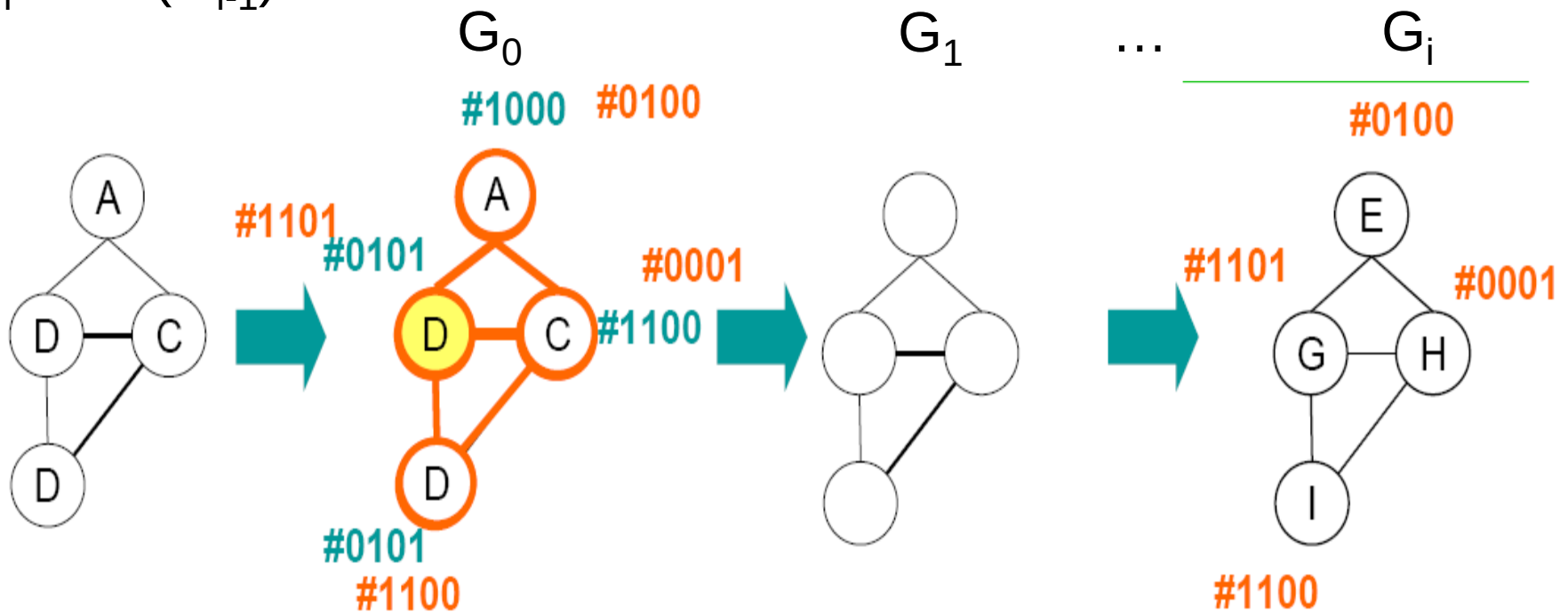
(b)

$NH(v)$ uniquely represents the distribution of the node labels around v

Neighborhood Hash over a Graph, NH(G)

$$G_i = \text{NH}(G_{i-1})$$

1st-Hash graph ith-Hash graph



G_{r+1} contains high-order relationships between the nodes with order r



Neighborhood Hash Graph Kernel

$$K^{(i)}_{NH}(G_i, G_i')$$

- Matching coefficient
 - 1.Sort hash values
 - 2.Count common labels
 - 3.Compute coefficient

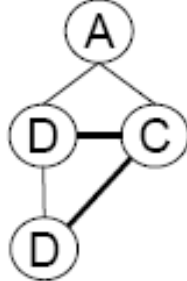
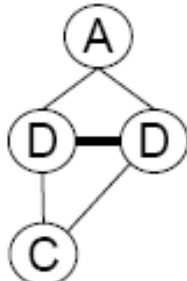
c

$$n_a + n_b - c$$

$(n_a, n_b : \text{\#node of each graphs})$



Example

Original graph		
Bit labels	A D C D #1000 #0101 #1100 #0101	A D D C #1000 #0101 #0101 #1100
1-Hash graph (one time hash)	E G H I #0100 #1101 #1100 #0001	F G G J #1001 #1101 #1101 #1011



INDIANA UNIVERSITY

Thank You!

QUESTIONS?



INDIANA UNIVERSITY

TABLE 2. FORECASTS FROM TABLE 1
SORTED BY DECREASING ELVIS-ED (SEE
APPENDIX) FORECAST PROBABILITY

Year	Event (1)/ non-event (0)	Probability (%)
1994	1	98.4
1995	1	95.2
1984	1	94.4
1981	0	92.8
1985	1	83.2
1986	1	81.6
1988	1	58.4
1982	0	57.6
1991	0	28.0
1987	0	13.6
1989	1	3.2
1992	0	2.4
1990	0	1.6
1983	0	0.8
1993	0	0.0

