Instructor: Sael Lee

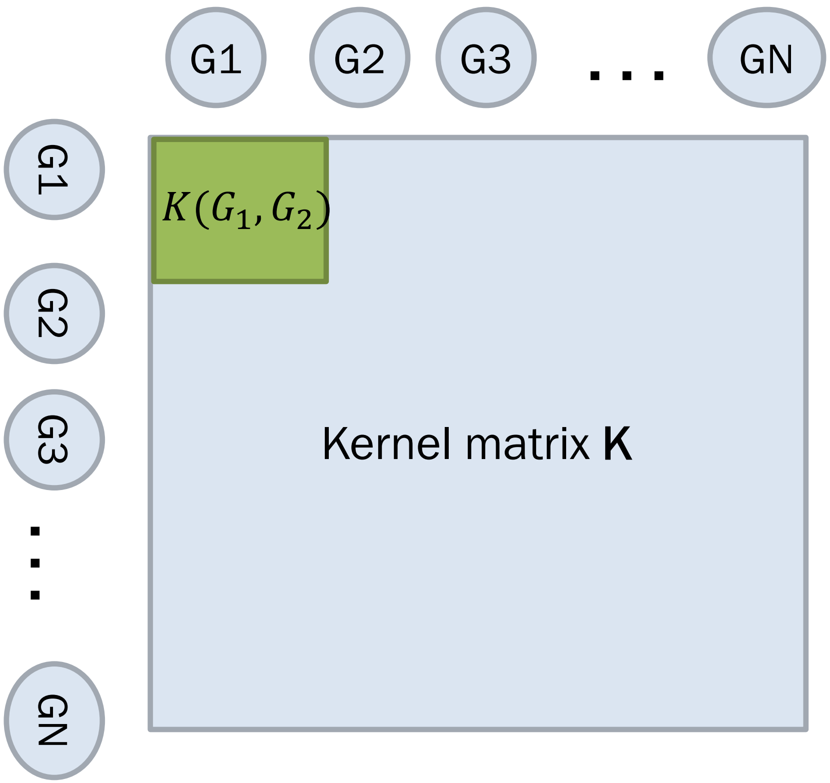
CS549 Spring – Computational Biology

LECTURE 21: GRAPH KERNELS

Resources:

- Gärtner, T., Flach, P., & Wrobel, S. (2003). On graph kernels: Hardness results and efficient alternatives. *Learning Theory and Kernel Machines* (pp. 129–143).
- Kashima, H., Tsuda, K., & Inokuchi, A. (2003). Marginalized kernels between labeled graphs. *ICML2003*.
- Mahé, P., Ueda, N., & Akutsu, T. (2004). Extensions of marginalized graph kernels. *ICML2004*.
    - **Also their slides presented in ICML2004**

GRAPH KERNELS



How to define a **valid kernel** function $K(G_j, G_j)$, between two graphs G_j and G_j.

- $K(G_j, G_j)$ should provide relationship (similarity / dissimilarity / correlation etc.) measure for between two graphs.
- $K(G_j, G_j)$ should be able to be applied in kernel based machine learning methods such that it provide optimal classification / clustering performance.

We will look at graph kernels that states similarity between kernels.

MARGINALIZED KERNELS BETWEEN LABELED GRAPHS

(Kashima et al., ICML 2003)

Marginalized Kernels

- Assume **hidden variables h** (ex> walk of a graph) and make use of the probability distribution of **visible variables x, x’** (structured data ex> Graph) and hidden variables

Marginalized Kernels: Expectation of the joint kernel over all possible values of *h* and *h’*

posterior probability

$$K(\boldsymbol{x},\boldsymbol{x}') = \sum_{h}\sum_{h'} K_z(\boldsymbol{z},\boldsymbol{z}')\boldsymbol{p}(\boldsymbol{h}|\boldsymbol{x})p(\boldsymbol{h}|\boldsymbol{x}')$$

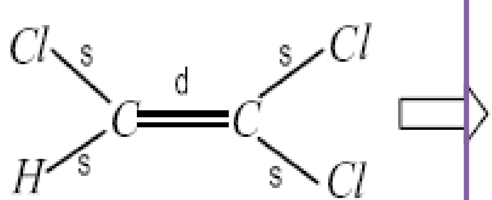
joint kernel & $z = [x;h]$

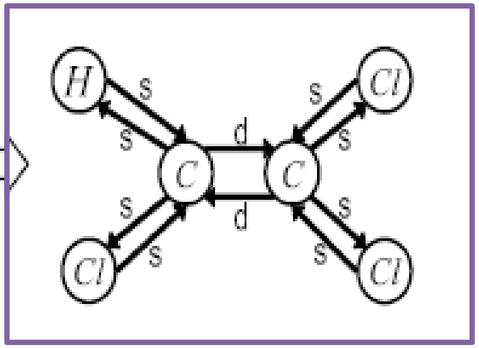
posterior probability $p(h|x)$ can be interpreted as a **feature extractor** that extracts informative features for classification from x

GRAPH KERNELS TERMINOLOGY

- A **graph** G =(V, E, l),
  - V is the set of vertices,
  - $E \subset (V \times V)$ is the set of undirected edges (Changed to directed for random walk), and
  - $l : V, E \rightarrow \Sigma$ is a function that assigns labels from an alphabet Σ to nodes in the graph.

Changing undirected graph to directed graph





G =(V, E, l),

- ’s’ and ’d’ denote single and double bonds, respectively.
- Kernel assumes a directed graph, undirected edges are replaced by directed edges

FIRST ORDER MARKOV RANDOM WALKS ON GRAPHS

Hidden variable: Random Walks on Graphs

$$K(\boldsymbol{x},\boldsymbol{x}') = \boxed{\sum_{\boldsymbol{h}}\sum_{\boldsymbol{h}'}} K_z(\boldsymbol{z},\boldsymbol{z}')p(\boldsymbol{h}|\boldsymbol{x})p(\boldsymbol{h}'|\boldsymbol{x}').$$

- Hidden variable $\boldsymbol{h} = (h_1, \ldots, h_l)$ associated with graph G is a sequence of natural numbers from 1 to |G|.

|G| : number of vertices

- *h* is generated by a random walk

uniform distribution can be used for uninformative prior

1-st step) h_1 is sampled from the **prior probability distribution** $p_s(\boldsymbol{h})$.

i-th step) h_i sampled subject to the **transition probability** $p_t(h_i|h_{i-1})$ and with **walk termination probability** $p_q(h_{i-1})$:

$$\sum_{j=1}^{|G|} p_t(j|i) + p_q(i) = 1.$$

- Posterior probability for the walk ***h* : *p(h|G)***

$$p(\boldsymbol{h}|G) = p_s(h_1)\prod_{i=2}^{\ell} p_t(h_i|h_{i-1})p_q(h_\ell),$$ *where* l is the length of ***h***

- traversed labels are listed: $v_{h_1} e_{h_1 h_2} v_{h_2} e_{h_2 h_3} v_{h_3} \cdots$

DEFINE JOINT KERNEL

$$K(\boldsymbol{x}, \boldsymbol{x}') = \sum_{\boldsymbol{h}} \sum_{\boldsymbol{h}'} \boxed{K_z(\boldsymbol{z}, \boldsymbol{z}')} p(\boldsymbol{h}|\boldsymbol{x}) p(\boldsymbol{h}'|\boldsymbol{x}').$$

Define vertex kernel & edge kernel

Assume that two kernel functions are readily defined:

- *K(v; v')* : *Kernel* between vertex labels
- *K(e; e')*: *Kernel* between edge labels,

Constrain both kernels to be nonnegative

$$K(v;\ v') \geq 0;\ K(e;\ e') \geq 0$$

Example of the vertex label kernels

Dirac kernel: For Discrete labels

$$K(v, v') = \delta(v = v'),$$

Gaussian kernel: For Real value labels

$$K(v, v') = \exp(- \parallel v - v' \parallel^2 / 2\sigma^2)$$

Joint Kernel

$$K_z(z, z') = \begin{cases} 0 & (\ell \neq \ell') \\ K(v_{h_1}, v'_{h'_1}) \prod_{i=2}^{\ell} K(e_{h_{i-1}h_i}, e'_{h'_{i-1}h'_i}) \times \\ \qquad K(v_{h_i}, v'_{h'_i}) & (\ell = \ell') \end{cases}$$

where $z = (G, h)$.

COMPUTING JOINT KERNEL

$$K(G,G') = \sum_{\ell=1}^{\infty}\sum_{\mathbf{h}}\sum_{\mathbf{h}'} p_s(h_1)\prod_{i=2}^{\ell} p_t(h_i|h_{i-1})p_q(h_l) \times p'_s(h'_1)\prod_{j=2}^{\ell} p'_t(h'_j|h'_{j-1})p'_q(h'_\ell) \times K(v_{h_1},v'_{h'_1})\prod_{k=2}^{\ell} K(e_{h_{k-1}h_k},e'_{h'_{k-1},h'_k})K(v_{h_k},v'_{h'_k}),$$

Where $\sum_{\mathbf{h}} := \sum_{h_1=1}^{|G|}\cdots\sum_{h_\ell=1}^{|G|}$

The straightforward enumeration is **impossible**, because l spans from 1 to infinity.

$$K(G,G') = \sum_{h_1,h'_1} s(h_1,h'_1)\lim_{L\to\infty}\sum_{\ell=1}^{L} r_\ell(h_1,h'_1) = \sum_{h_1,h'_1} s(h_1,h'_1)\lim_{L\to\infty} R_L(h_1,h'_1),$$

$$r_\ell(h_1,h'_1) := \left(\sum_{h_2,h'_2} t(h_2,h'_2,h_1,h'_1)\left(\sum_{h_3,h'_3} t(h_3,h'_3,h_2,h'_2)\times \left(\cdots\left(\sum_{h_\ell,h'_\ell} t(h_\ell,h'_\ell,h_{\ell-1},h'_{\ell-1})q(h_\ell,h'_\ell)\right)\right)\cdots\right)\right),$$

$\ell \geq 2$

$$s(h_1,h'_1) := p_s(h_1)p'_s(h'_1)K(v_{h_1},v'_{h'_1})$$

$$t(h_i,h'_i,h_{i-1},h'_{i-1}) := p_t(h_i|h_{i-1})p'_t(h'_i|h'_{i-1}) \times K(v_{h_i},v'_{h'_i})K(e_{h_{i-1}h_i},e_{h'_{i-1}h'_i})$$

$$q(h_\ell,h'_\ell) := p_q(h_\ell)p'_q(h'_\ell)$$

$$r_1(h_1,h'_1) := q(h_1,h'_1).$$

$$R_L(h_1,h'_1) := \sum_{\ell=1}^{L} r_\ell(h_1,h'_1).$$

COMPUTING JOINT KERNEL CONT.

Restate this problem in **recursive form**

$$r_\ell(h_1, h_1') := \left(\sum_{h_2, h_2'} t(h_2, h_2', h_1, h_1') \left(\sum_{h_3, h_3'} t(h_3, h_3', h_2, h_2') \times \left(\cdots \left(\sum_{h_\ell, h_\ell'} t(h_\ell, h_\ell', h_{\ell-1}, h_{\ell-1}') q(h_\ell, h_\ell') \right) \right) \cdots \right) \right.$$

$$r_1(h_1, h_1') := q(h_1, h_1')$$

$$R_L(h_1, h_1') := \sum_{\ell=1}^{L} r_\ell(h_1, h_1').$$

$$r_k(h_1, h_1') = \sum_{i,j} t(i, j, h_1, h_1') r_{k-1}(i, j).$$

$$R_L(h_1, h_1') = r_1(h_1, h_1') + \sum_{k=2}^{T} r_k(h_1, h_1')$$

$$= r_1(h_1, h_1') + \sum_{k=2}^{T} \sum_{i,j} t(i, j, h_1, h_1') r_{k-1}(i, j)$$

$$= r_1(h_1, h_1') + \sum_{i,j} t(i, j, h_1, h_1') R_{L-1}(i, j). \; ($$

Equilibrium equation:

$$R_\infty(h_1, h_1') = r_1(h_1, h_1') + \sum_{i,j} t(i, j, h_1, h_1') R_\infty(i, j)$$

COMPUTING JOINT KERNEL CONT.

computation of the marginalized kernel finally comes down to iteratively solving for

$$R_L(h_1, h_1') = r_1(h_1, h_1') + \sum_{k=2}^{T} r_k(h_1, h_1')$$

$$r_k(h_1, h_1') = \sum_{i,j} t(i, j, h_1, h_1') r_{k-1}(i, j).$$

$$= r_1(h_1, h_1') + \sum_{k=2}^{T} \sum_{i,j} t(i, j, h_1, h_1') r_{k-1}(i, j)$$

$$= r_1(h_1, h_1') + \sum_{i,j} t(i, j, h_1, h_1') R_{L-1}(i, j). \; ($$

until convergence starting from

$$R_1(h_1, h_1') = r_1(h_1, h_1') := q(h_1, h_1')$$

$$q(h_\ell, h_\ell') := p_q(h_\ell) p_q'(h_\ell')$$

Proof of convergence in Section 3.4 of Kashima et al., 2003

and substituting the solutions into

$$K(G, G') = \sum_{h_1, h_1'} s(h_1, h_1') \lim_{L \to \infty} \sum_{\ell=1}^{L} r_\ell(h_1, h_1')$$

$$s(h_1, h_1') := p_s(h_1) p_s'(h_1') K(v_{h_1}, v'_{h_1'})$$

$$= \sum_{h_1, h_1'} s(h_1, h_1') \lim_{L \to \infty} R_L(h_1, h_1'),$$

EXTENSION TO MARGINALIZED GRAPH KERNEL

(Mahé et al. ICML 2004)

Model: Marginalized Graph Kernel with **Dirac** joint kernel

Approaches:

1 Size of product graph affects runtime of kernel computation

- The **more node labels, the smaller the product graph**
- Trick: Introduce new artificial node labels

Iterative Label Enrichment:
Morgan Index (1965)

2 Focusing on non-tottering walks is a way to get closer to the path kernel

Reduce Tottering effect by
Using **2nd Order Markov Random Walk** instead of 1st order

SIMPLIFIED MARGINALIZED GRAPH KERNEL

K: Marginalized graph kernel

$$K(\boldsymbol{x},\boldsymbol{x}') = \sum_{\boldsymbol{h}}\sum_{\boldsymbol{h}'} K_z(\boldsymbol{z},\boldsymbol{z}')p(\boldsymbol{h}|\boldsymbol{x})p(\boldsymbol{h}'|\boldsymbol{x}').$$

$$K_z(\boldsymbol{z},\boldsymbol{z}') = \begin{cases} 0 & (\ell \neq \ell') \\ K(v_{h_1}, v'_{h'_1}) \prod_{i=2}^{\ell} K(e_{h_{i-1}h_i}, e'_{h'_{i-1}h'_i}) \times K(v_{h_i}, v'_{h'_i}) & (\ell = \ell') \end{cases}$$

where $\boldsymbol{z} = (G, \boldsymbol{h})$.

Simplified by
1) not using edge kernel defined
2) Using Dirac vertex kernel

$$K(G,G') = \sum_{(\boldsymbol{h},\boldsymbol{h}') \in V^* \times V'^*} p(\boldsymbol{h}|G)p'(\boldsymbol{h}'|G')K_L(l(\boldsymbol{h}), l(\boldsymbol{h}'))$$

K_L: Dirac kernel between labeled sequence

$$K_L(l,l') = \begin{cases} 1 & \text{if } l = l' \\ 0 & \text{otherwise} \end{cases}$$

$$p(v_1 \ldots v_n) = p_s(v_1)\prod_{i=2}^{n} p_t(v_i|v_{i-1}).$$

$$\begin{cases} p_s(v) = p_0(v)p_q(v), \\ p_t(u|v) = \frac{1-p_q(v)}{p_q(v)} p_a(u|v)p_q(u). \end{cases}$$

SIMPLIFIED MARGINALIZED GRAPH KERNEL IN MATRIX FORMAT

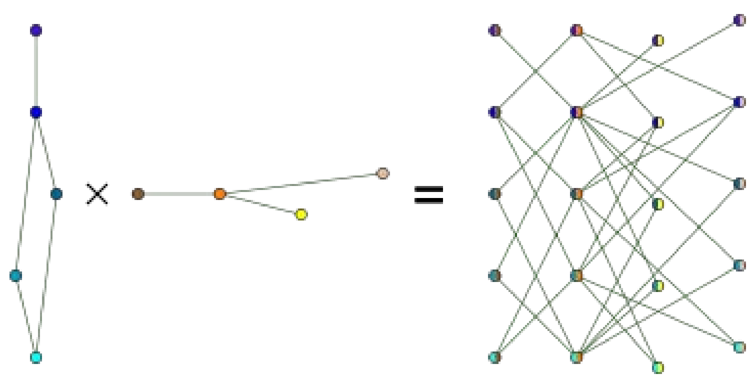
two labeled graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$

Tensor product graph id defined as labeled graph $G_p = (V_p, \mathrm{E}_p)$ with $V_p \subset V_1 \times V_2$ are pairs of vertices with identical labels

$$(v_1, v_2) \in V_p \ iff \ l(v_1) = l(v_2)$$

and edges connecting the vertices

$$(u_1, u_2)\mathrm{and}(v_1, v_2) \ iff \ (u_i, v_i) \in E_p, for \ \ i = 1,2, \dots l$$



G H G × H

$$\mathbf{A} \otimes \mathbf{B} = \begin{bmatrix}
a_{11}b_{11} & a_{11}b_{12} & \cdots & a_{11}b_{1q} & \cdots & \cdots & a_{1n}b_{11} & a_{1n}b_{12} & \cdots & a_{1n}b_{1q} \\
a_{11}b_{21} & a_{11}b_{22} & \cdots & a_{11}b_{2q} & \cdots & \cdots & a_{1n}b_{21} & a_{1n}b_{22} & \cdots & a_{1n}b_{2q} \\
\vdots & \vdots & \ddots & \vdots & & & \vdots & \vdots & \ddots & \vdots \\
a_{11}b_{p1} & a_{11}b_{p2} & \cdots & a_{11}b_{pq} & \cdots & \cdots & a_{1n}b_{p1} & a_{1n}b_{p2} & \cdots & a_{1n}b_{pq} \\
\vdots & \vdots & & \vdots & \ddots & & \vdots & \vdots & & \vdots \\
\vdots & \vdots & & \vdots & & \ddots & \vdots & \vdots & & \vdots \\
a_{m1}b_{11} & a_{m1}b_{12} & \cdots & a_{m1}b_{1q} & \cdots & \cdots & a_{mn}b_{11} & a_{mn}b_{12} & \cdots & a_{mn}b_{1q} \\
a_{m1}b_{21} & a_{m1}b_{22} & \cdots & a_{m1}b_{2q} & \cdots & \cdots & a_{mn}b_{21} & a_{mn}b_{22} & \cdots & a_{mn}b_{2q} \\
\vdots & \vdots & \ddots & \vdots & & & \vdots & \vdots & \ddots & \vdots \\
a_{m1}b_{p1} & a_{m1}b_{p2} & \cdots & a_{m1}b_{pq} & \cdots & \cdots & a_{mn}b_{p1} & a_{mn}b_{p2} & \cdots & a_{mn}b_{pq}
\end{bmatrix}.$$

tensor product of agency matrix G × H

Fig: http://en.wikipedia.org/wiki/Tensor_product_of_graphs

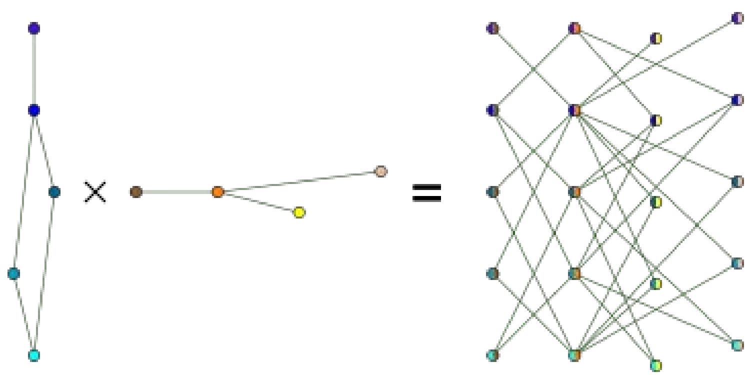
SIMPLIFIED MARGINALIZED GRAPH KERNEL IN MATRIX FORMAT

A function π on the set of walks(paths) $H(G_p)$

$$\pi\left((u_1,v_1)(u_2,v_2)\ldots(u_n,v_n)\right) = \pi_s\left(u_1,v_1\right)\prod_{i=2}^{n}\pi_t\left((u_i,v_i)|(u_{i-1},v_{i-1})\right),$$

with

$$\begin{cases}\pi_s(u_1,u_2) = p_s^{(1)}(u_1)p_s^{(2)}(u_2),\\ \pi_t((v_1,v_2)|(u_1,u_2)) = p_t^{(1)}(v_1|u_1)p_t^{(2)}(v_2|u_2),\end{cases}$$



G H G × H

π_t

$$\mathbf{A}\otimes\mathbf{B} = \begin{bmatrix} a_{11}b_{11} & a_{11}b_{12} & \cdots & a_{11}b_{1q} & \cdots & \cdots & a_{1n}b_{11} & a_{1n}b_{12} & \cdots & a_{1n}b_{1q} \\ a_{11}b_{21} & a_{11}b_{22} & \cdots & a_{11}b_{2q} & \cdots & \cdots & a_{1n}b_{21} & a_{1n}b_{22} & \cdots & a_{1n}b_{2q} \\ \vdots & \vdots & \ddots & \vdots & & & \vdots & \vdots & \ddots & \vdots \\ a_{11}b_{p1} & a_{11}b_{p2} & \cdots & a_{11}b_{pq} & \cdots & \cdots & a_{1n}b_{p1} & a_{1n}b_{p2} & \cdots & a_{1n}b_{pq} \\ \vdots & \vdots & & \vdots & \ddots & & \vdots & \vdots & & \vdots \\ \vdots & \vdots & & \vdots & & \ddots & \vdots & \vdots & & \vdots \\ a_{m1}b_{11} & a_{m1}b_{12} & \cdots & a_{m1}b_{1q} & \cdots & \cdots & a_{mn}b_{11} & a_{mn}b_{12} & \cdots & a_{mn}b_{1q} \\ a_{m1}b_{21} & a_{m1}b_{22} & \cdots & a_{m1}b_{2q} & \cdots & \cdots & a_{mn}b_{21} & a_{mn}b_{22} & \cdots & a_{mn}b_{2q} \\ \vdots & \vdots & \ddots & \vdots & & & \vdots & \vdots & \ddots & \vdots \\ a_{m1}b_{p1} & a_{m1}b_{p2} & \cdots & a_{m1}b_{pq} & \cdots & \cdots & a_{mn}b_{p1} & a_{mn}b_{p2} & \cdots & a_{mn}b_{pq} \end{bmatrix}.$$

SIMPLIFIED MARGINALIZED GRAPH KERNEL IN MATRIX FORMAT CONT.

$$K(G_1, G_2) = \sum_{(\boldsymbol{h_1}, h_2) \in V_1{}^* \times V_2{}^*} p_1(\boldsymbol{h_1} | G_1) p_2(\boldsymbol{h_2} | G_1) K_L(l(\boldsymbol{h_1}), l(\boldsymbol{h_2}))$$

$$K(G_1, G_2) = \sum_{h \in H(\mathcal{G})} \pi(h).$$

$$\sum_{h \in H(\mathcal{G}), |h| = n} \pi(h) = \pi_s^\top \Pi_t^n \mathbf{1},$$

$$K(G_1, G_2) = \sum_{n=1}^{\infty} \left(\sum_{h \in H(\mathcal{G}), |h| = n} \pi(h) \right) = \pi_s^\top (I - \Pi_t)^{-1} \mathbf{1}.$$

LABEL ENRICHMENT WITH MORGAN INDEX (1965)

Problems:

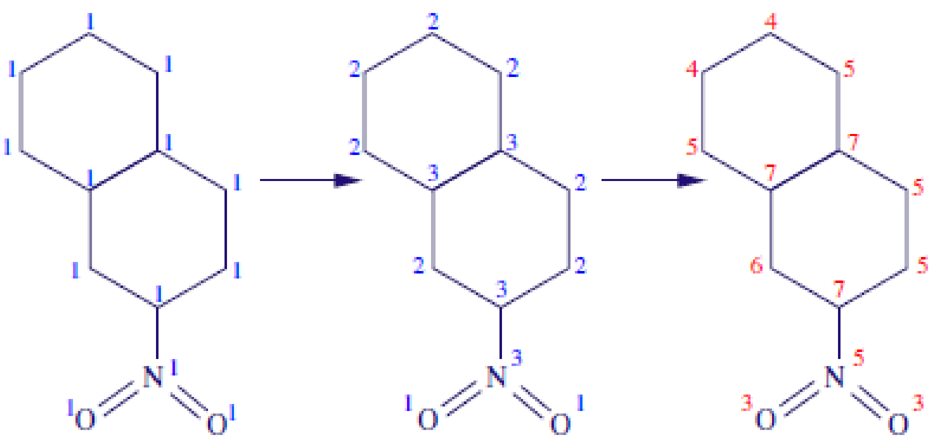
- The computation of graph kernels is time-consuming.
- Need to increase the relevance of the features used to compare graphs.

Expected outcome:

- The computation of graph kernels is time-consuming.
- Need to increase the relevance of the features used to compare graphs.

LABEL ENRICHMENT WITH MORGAN INDEX CONT.

Enrichment with <u>vertex connectivity properties</u>
→ **extended connectivity descriptor :**

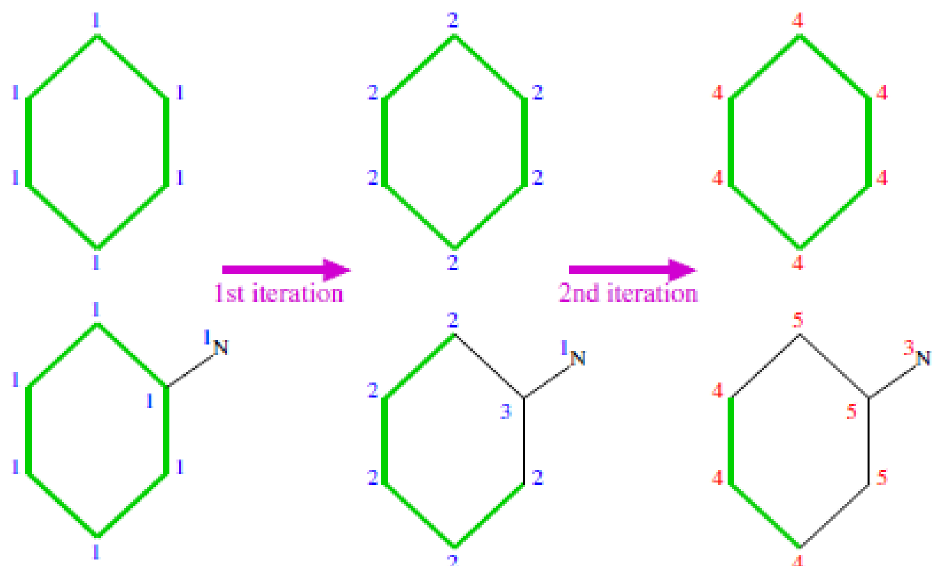


Original Graph — 1ST ITERATION — 2ND ITERATION

<u>Algorithm</u> :

- $M_0(v) = 1, \ \forall\, v$
- $M_t(v) = \sum_{\text{neig}(v)} M_{t-1}(u)$
- $\Rightarrow$ New label :
  $l_t(v) = l(v) \circ M_t(v)$

M_n: vector of labels in graph
Given adjacency matrix A and setting $M_0 = \mathbf{1}$
$M_{n+1} = (A + I)M_n$



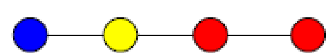
$\Rightarrow$ Family of kernels K_n : $\begin{cases} \text{decreased kernel complexity } (\mathcal{O}(|G_1 \times G_2|^3)) \\ \text{(potential) increased kernel expressivity} \end{cases}$

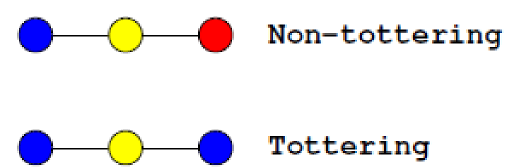
PREVENTING TOTTERING

Tottering

A **tottering walk** is a walk $w = v_1 \ldots v_n$ with $v_i = v_i + 2$ for some i.

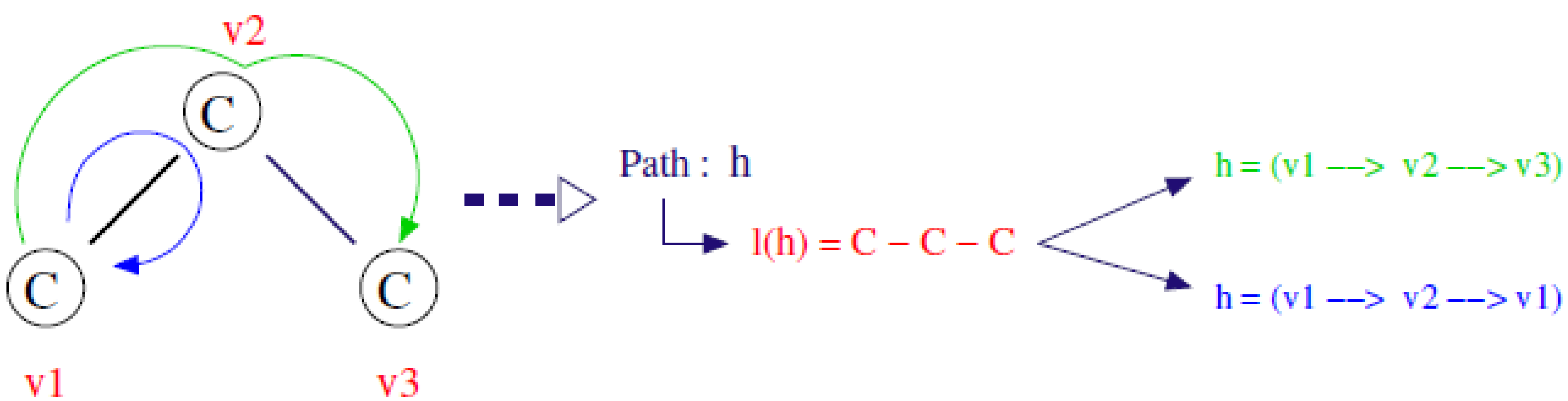
- A walk can visit the same cycle of nodes all over again
- Kernel measures similarity in terms of common walks
- Hence a small structural similarity can cause a huge kernel value
- Focusing on non-tottering walks is a way to get closer to the path kernel (e.g., equivalent on trees).





PREVENTING TOTTERING CONT.

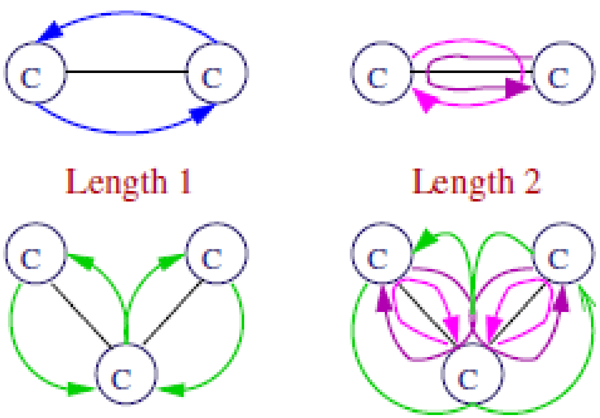
- Tottering path : $h = (v_1, \dots, v_n)$, $\exists\, i : v_{i+2} = v_i$.



$\Rightarrow$ preventing totters $\Leftrightarrow$ filtering blue path.

PREVENTING TOTTERING CONT.

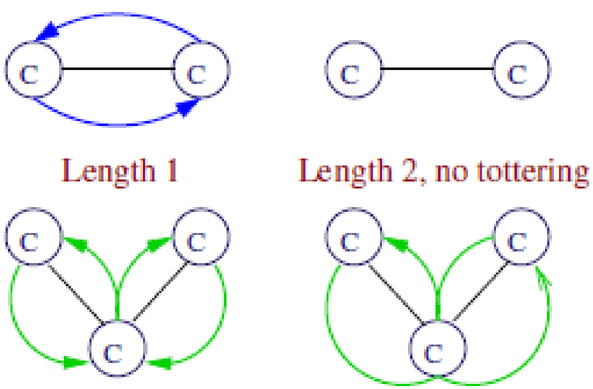
- Motivation:



- ⋆ Every path of G_2 can be matched to a tottering path of G_1
- ⋆ $\Rightarrow$ Compounds are considered as identical

PREVENTING TOTTERING CONT.

- Motivation:



★ Only "real" chemical path are matched

★ $\Rightarrow$ Compounds are now seen as different

- Solution : increase the order of the random walk model :

$\Rightarrow p_G(h) = p_s(v_1)p_t(v_2|v_1)\prod_{i=3}^{n} p_t(v_i|v_{i-2}, v_{i-1})$

2ND ORDER MARKOV RANDOM WALK

$$p_G(h) = p_s(v_1)p_t(v_2|v_1)\prod_{i=3}^{n} p_t(v_i|v_{i-2}, v_{i-1})$$

$$\begin{cases} p_s(v) = p_0(v)p_q^{(0)}(v), \\ p_t(u|v) = \frac{1-p_q^{(0)}(v)}{p_q^{(0)}(v)} p_a(u|v)p_q(u), \\ p_t(u|w,v) = \frac{1-p_q(v)}{p_q(v)} p_a(u|w,v)p_q(u). \end{cases}$$

The function is still a valid kernel but the implementation described for the first order Markov random walk cannot be directly used anymore.

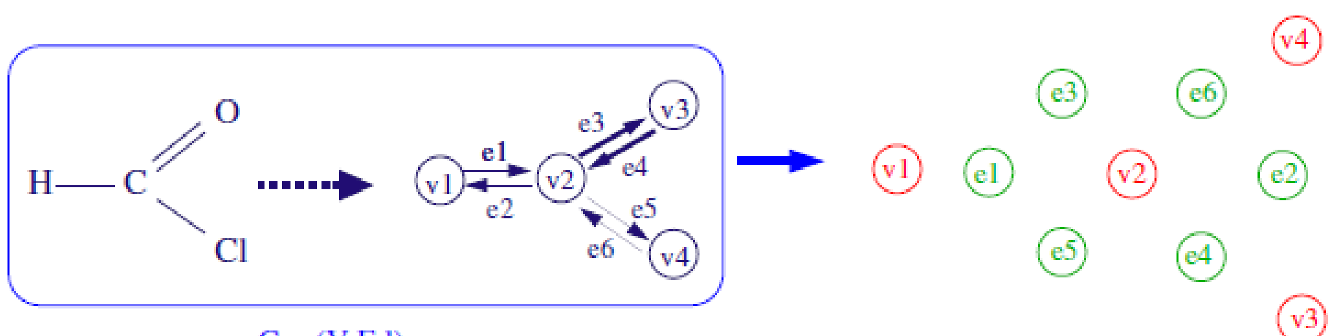
=> Instead of explicitly working with 2nd Order Markov Random walk, transform the original graph G to G' such that G' contains the look ahead information.

GRAPH TRANSFORMATION CONT.

* Don’t confuse G’ used in the last notation for compared Graph

Transformation : $G = (V, E, l) \Rightarrow G' = (V', E', l')$ where :

- $V' = V \cup E$
- $E' = \{(v, (v, t)) \mid v \in V, (v, t) \in E\}$
  $\cup \{((u, v), (v, t)) \mid (u, v), (v, t) \in E, u \neq t\}$

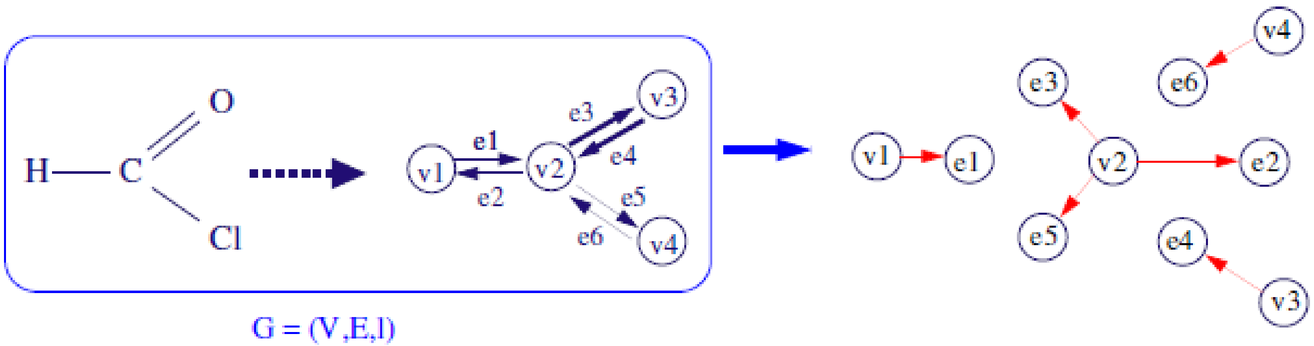


G = (V,E,l)

GRAPH TRANSFORMATION CONT.

Transformation : $G = (V, E, l) \Rightarrow G' = (V', E', l')$ where :

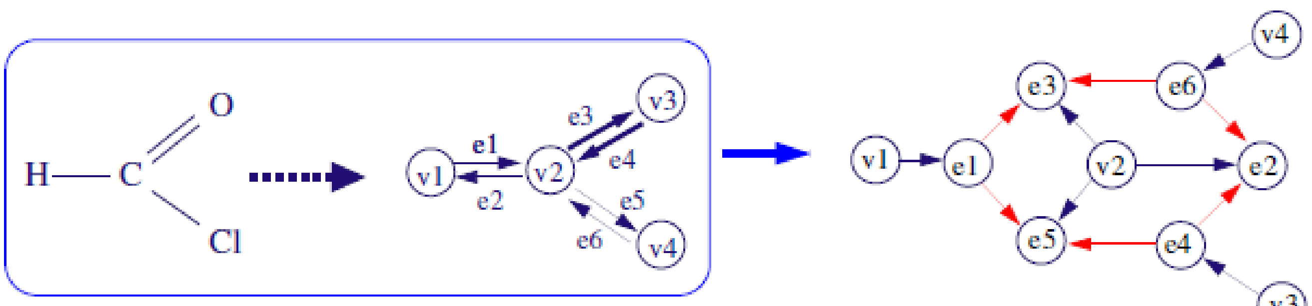
- $V' = V \cup E$
- $E' = \{(v, (v,t)) \mid v \in V, (v,t) \in E\}$
  $\cup \{((u,v), (v,t)) \mid (u,v), (v,t) \in E, u \neq t\}$



GRAPH TRANSFORMATION CONT.

Transformation : $G = (V, E, l) \Rightarrow G' = (V', E', l')$ where :

- $V' = V \cup E$
- $E' = \{(v, (v, t)) \,|\, v \in V, (v, t) \in E\}$
  $\cup \; \{((u, v), (v, t)) \,|\, (u, v), (v, t) \in E, u \neq t\}$

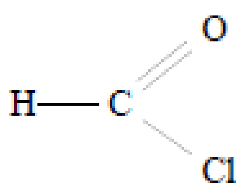


G = (V,E,l)

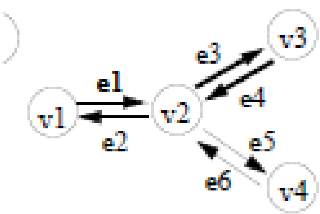
Original Graph

Corresponding directed graph G = (V,E,I)

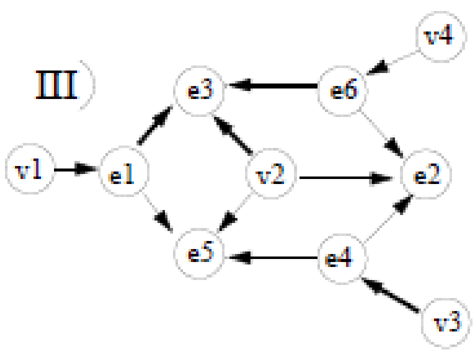
I)



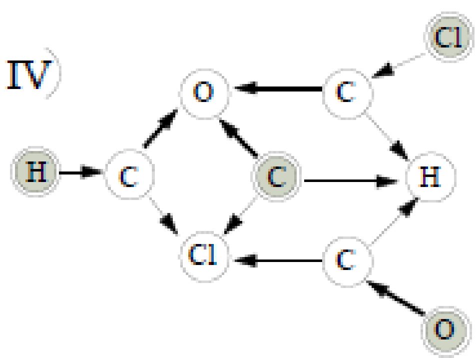
II)



III)



IV)



Transformed Graph

Labels in the transformed graph

MODIFIED KERNEL COMPUTATION CONT.

- Consider : $\begin{cases} H_0(G) = \{\text{Non tottering paths of G}\} \\ H_1(G') = \{\text{Paths of } G' \text{ starting from a node } v \in V \ \} \end{cases}$

- Theorem: p' factorizes as

$$p'(h') = p'_s(v'_1) \prod_{i=2}^{n} p'_t(v'_i | v'_{i-1})$$

  - ⋆ $p'_s(v') = p_s(v')$
  - ⋆ $p'_t(u'|v') = \begin{cases} p_t(u|v') \text{ if } v' \in V \text{ and } u' = (v', u) \in E \\ p_t(u|v, w) \text{ if } v' = (v, w) \text{ and } u' = (w, u) \in E \end{cases}$

- Corollary :

$$\left.\begin{array}{l}\text{- graph transformation} \\ \text{- original graph kernel}\end{array}\right\} \Rightarrow \text{tottering paths removed}$$

MODIFIED KERNEL COMPUTATION

- Consider : $\begin{cases} H_0(G) = \{\text{Non tottering paths of G}\} \\ H_1(G') = \{\text{Paths of } G' \text{ starting from a node } v \in V\ \} \end{cases}$

- The mapping $f : H_0(G) \to H_1(G')$ defined by

$$h = (v_1, ..., v_n) \mapsto h' = (v'_1, ..., v'_n) \text{ such that } \begin{cases} v'_1 = v_1 \\ v'_i = (v_{i-1}, v_i) \end{cases}$$

establishes a bijection between $H_0(G)$ and $H_1(G')$

one-to-one correspondence

- Let p' be the image of p_G by f:

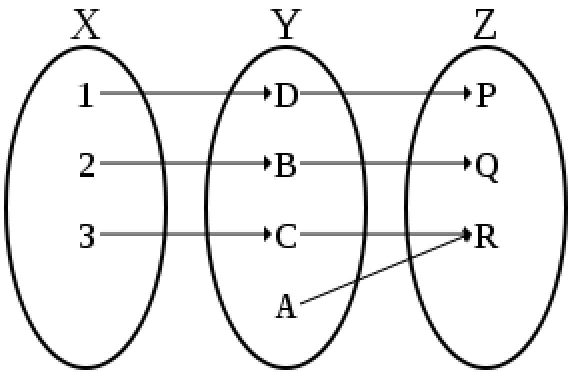
$$\forall h' \in H_1(G'), \quad p'(h') := p_G\left(f^{-1}(h')\right)$$

REVIEW BIJECTION

http://en.wikipedia.org/wiki/Bijection

- **Bijection** (or bijective function or one-to-one correspondence) is a function giving an exact pairing of the elements of two sets.

- **Bijective function** *f*: *X* → *Y* is a one to one and onto mapping of a set *X* to a set *Y*.



A bijection composed of an injection (left) and a surjection (right).

Theorem 1. f is a **Bijective function** between $H_0(G)$ and $H_1(G')$, and for any path $\mathbf{h} \in H_0(G)$ we have

$$f: H_0(G) \rightarrow H_1(G')$$

$$\begin{cases} l(\boldsymbol{h}|G) = l'(f(\boldsymbol{h})|G') \\ p(\boldsymbol{h}|G) = p'(f(\boldsymbol{h})|G') \end{cases}$$

Corollary 1. For any two graphs G_1 and G_2, the marginalized graph kernel can be expressed in terms of the transfromed graphs G'_1 and G'_2 by:

$$K(G_1, G_2) = \sum_{(h'_1, h'_2) \in (\Sigma'_1)^* \times (\Sigma'_2)^*} p'_1(h'_1) p'_2(h'_2)\, K_L(l'_1(h'_1) l'_2(h'_2))$$