Graphlet Kernels

Karsten Borgwardt and Nino Shervashidze

joint work with SVN Vishwanathan, Tobias Petri,
and Kurt Mehlhorn

Machine Learning and
Computational Biology Research Group,
Max Planck Institute for Biological Cybernetics and
Max Planck Institute for Developmental Biology, Tübingen

appeared in AISTATS 2009

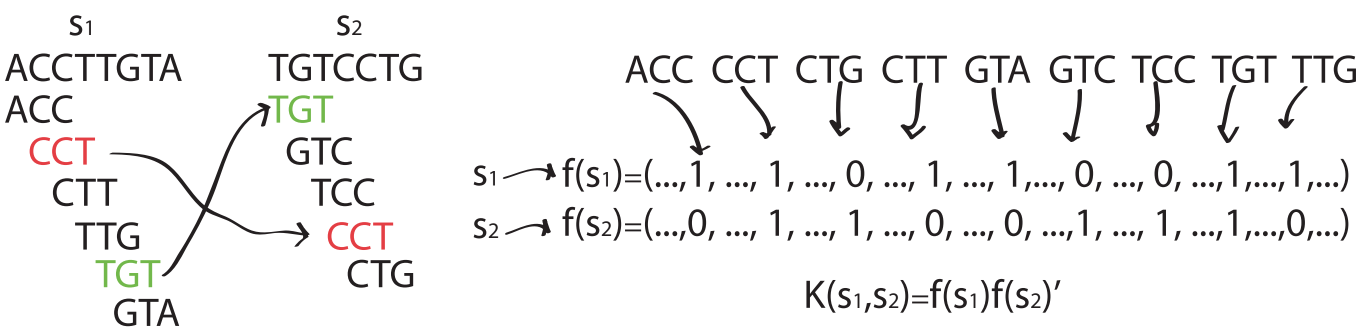
String kernels

Recall the *k*-mer kernel on strings

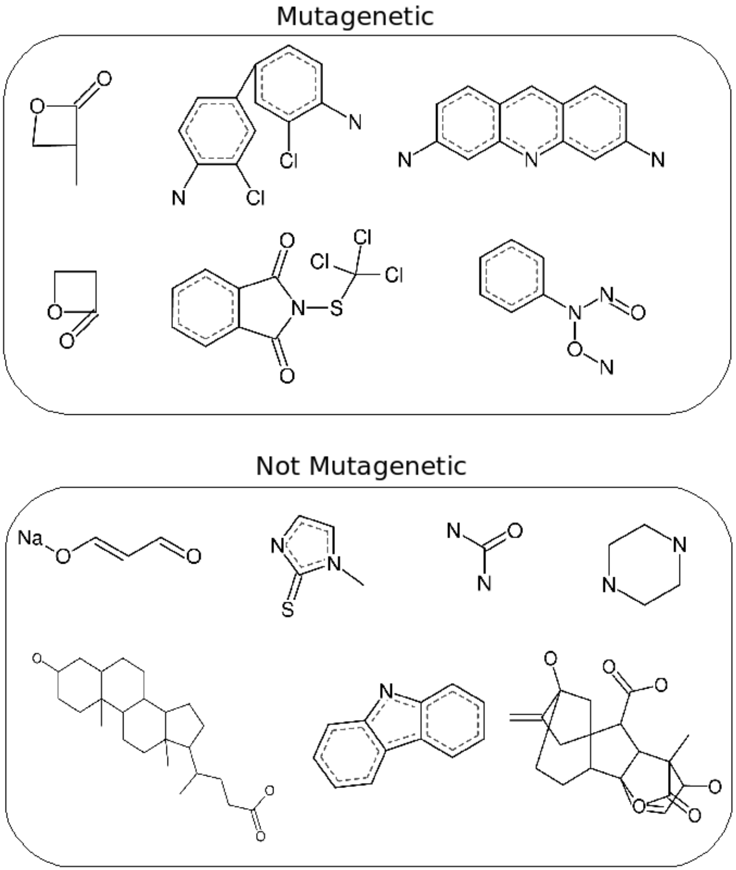
- Basic idea: count the number of common contiguous substrings of length k

This is equivalent to:

- count the number of occurrences of all k-mers in strings s_1 and s_2 separately,
- compute the inner product between these counts.



Graph comparison



Graph kernels

Graph kernels have traditionally been based on different ideas

- Random walk kernel ($O(n^3)$)
- Shortest path kernel ($O(n^4)$)
- Subtree kernel (NP-hard)
- Cycle kernel (NP-hard)
- All possible subgraphs kernel (NP-hard)

Graphlet kernel

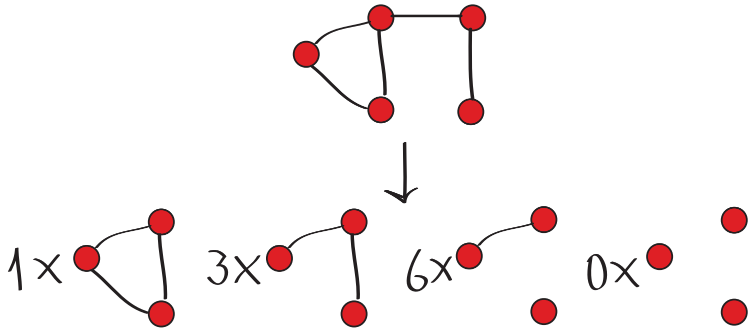
We call graphlets subgraphs of size $\{3, 4, 5\}$.

Let $\mathcal{G} = \{graphlet(1), \ldots, graphlet(N_k)\}$ be the set of size-k graphlets and G be a graph of size n.

Define a vector f_G of length N_k such that

$$f_{G_i} = \#(graphlet(i) \sqsubseteq G).$$

We call f_G the k-spectrum of G.



In this figure $n = 5$, $k = 3$, $f_G = (1, 3, 6, 0)$.

Graphlet kernel

Given two graphs G and G' of size $n \geq k$, the graphlet kernel k_g is defined as

$$k_g(G, G') := f_G^\top f_{G'}.$$

Problem: if G and G' have different sizes, this will greatly skew the counts f_G

Solution: normalize the counts to frequency vectors:

$$D_G = \frac{1}{\#all\ graphlets\ in\ G} f_G$$

and work with the normalized variant of k_g

$$k_g(G, G') = D_G^\top D_{G'}.$$

Link to graph reconstruction

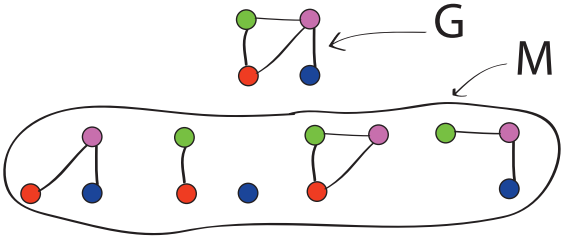
Isomorphism of graphs $\rightarrow$ equality of their k-spectra.

Equality of their k-spectra $\rightarrow$ isomorphism?

Yes, when $n = k + 1$ and $n \leq 11$...

Graph reconstruction conjecture

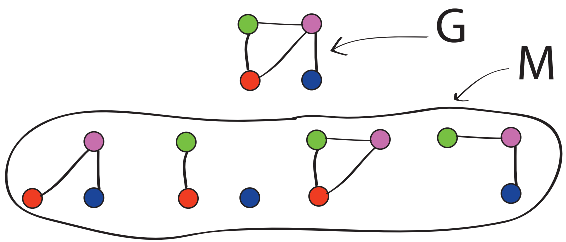
- Let G_v denote a subgraph of G, obtained by deleting node v and all the edges incident to it.
- Let G and G' be graphs of size greater than 2 and $g : V \rightarrow V'$ be an isomorphism function such that G_v is isomorphic to $G'_{g(v)}$ for all $v \in V$. Then G is isomorphic to G'.



Link to graph reconstruction

Recursive definition of the graphlet kernel

Given two graphs G and G' of size $n \geq k$, let $\mathcal{M}$ and $\mathcal{M}'$ denote the set of size-n-1 subgraphs of G and G' respectively.



The recursive graph kernel based on these subgraphs is defined as

$$k_n(G, G') = \begin{cases} \frac{1}{(n-k)^2} \sum_{S \in \mathcal{M}, S' \in \mathcal{M}'} k_{n-1}(S, S') & \text{if } n > k, \\ \delta(G \cong G') & \text{if } n = k \end{cases}$$

where $\delta(G \cong G')$ is 1 if G and G' are isomorphic, 0 otherwise.

The graphlet kernel is defined as $k_g(G, G') := k_n(G, G')$.

How to reduce runtime?

The kernel is defined, but how to compute graphlet distributions?

Counting size-k graphlets by exhaustive enumeration takes $O(n^k)$.

This is too expensive.

We propose 2 schemes to speed up the computation. We show that

- sampling a fixed number of graphlets suffices to bound the l_1 deviation of the empirical estimates of the graphlet distribution from the true distribution.
- for graphs of degree bounded by d, the exact number of all graphlets of size k can be determined in time $O(nd^{k-1})$. Large real world graphs are often sparse with $d \ll n$.

Sampling from graphs

Given a multiset $X := \{X_j\}_{j=1}^m$ of independent identically distributed (iid) random variables $X_j \sim D$, the empirical estimate of D is defined as

$$\hat{D}^m(i) = \frac{1}{m} \sum_{j=1}^{m} \delta(X_j = i),$$

where $i \in \mathcal{A}$, and δ is an indicator function.
Let D be a probability distribution on the finite set $\mathcal{A} = \{1, \ldots, a\}$. Let $X := \{X_j\}_{j=1}^m$, with $X_j \sim D$. For a given $\epsilon > 0$ and $\delta > 0$,

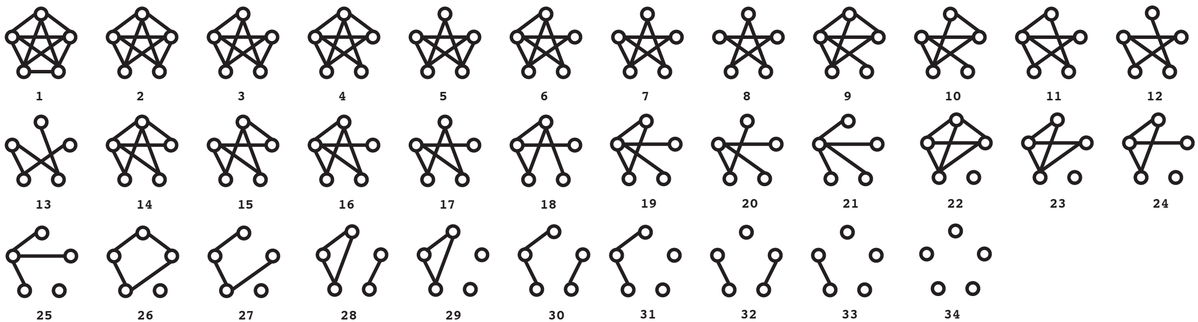
$$m = \left\lceil \frac{2\left(\log 2 \cdot a + \log\left(\frac{1}{\delta}\right)\right)}{\epsilon^2} \right\rceil$$

samples suffice to ensure that $P\left\{||D - \hat{D}^m||_1 \geq \epsilon\right\} \leq \delta$.

Sampling from graphs

Example

- Consider size-5 graphlets with $\epsilon = 0.05$, $\delta = 0.05$
- $a = 34$, as there are 34 pairwise non-isomorphic graphlets of size 5



- We obtain $m = 21251$ *independent* from the size of graphs we want to compare
- $21251 \ll n^5$, $\forall n > 9$.

Bounded degree graphs

There is a large fraction of graphs on which complete counting of graphlets can be performed efficiently: graphs of bounded degree d.

We present 2 algorithms which exploit the low degree:

- one for enumerating *all connected graphlets*,
- one for counting *all graphlets*.

Both have $O(nd^{k-1})$ runtime complexity, but the first one is faster in practice

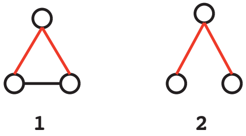
Bounded degree graphs

Count connected graphlets of size k, $k \in \{3, 4, 5\}$

Notice that most connected graphlets contain size-k simple paths

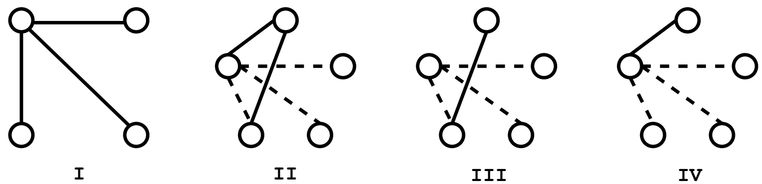
Provided this, the idea is simple:

- enumerate simple paths of k nodes ($O(nd^{k-1})$)
- for each path, look up adjacencies among these k nodes to decide which graphlet we obtain ($O(1)$ provided that we have a data structure allowing for this)
- each graphlet will be counted as many times, as the number of k-node paths it contains → divide counts by these numbers



Bounded degree graphs

Problem: while for size-3 graphlets all connected graphlets contain simple paths of k nodes, this is no more the case for size-4 and 5 graphlets.



Solution:

- To count I, we look up the $\binom{d_i}{3}$ neighbor triplets of each v_i, and check if they induce the graphlet we are interested in ($O(nd^3)$)
- II, III and IV contain I. So we first enumerate all occurrences of I, and then check the neighbors of each node in I to see if they induce the graphlets in question ($O(nd^4)$)

Bounded degree graphs

Count all graphlets of size k, $k \in \{3, 4, 5\}$

The basic idea:

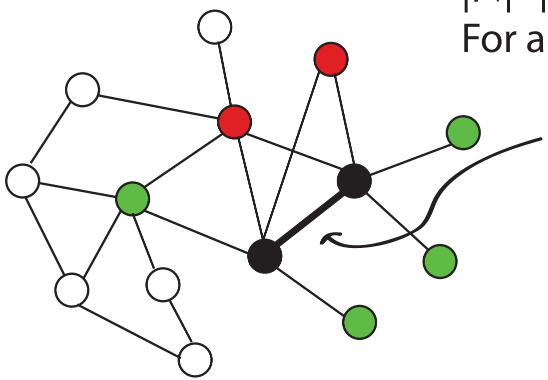
- enumerate all connected graphlets
- obtain counts of disconnected graphlets by subtracting previously obtained quantities from precomputed quantities

Bounded degree graphs

Count all graphlets of size k, $k \in \{3, 4, 5\}$ (continued)

Example: 3-node graphlets

There are 4 types of 3-node graphlets: denote them F_i, $i \in \{0, 1, 2, 3\}$, F_i contains i edges



First count graphlets containing at least one edge

$|F_1|=|F_2|=|F_3|=0$

For all edges do (0(nd))

Current edge

$|F_3|=|F_3|+\#$(red nodes)

$|F_2|=|F_2|+\#$(green nodes)

$|F_1|=|F_1|+(n-2-\#$(red and green nodes))

} (0(d))

$|F_3|=|F_3|/6$, $|F_2|=|F_2|/4$, $|F_1|=|F_1|/2$

$|F_0|=\binom{n}{3} - (|F_1|+|F_2|+|F_3|)$

Experiments

Statistics on datasets

dataset	size	classes	# nodes	# edges	d
MUTAG	188	2 (125 vs. 63)	17.7	38.9	4
PTC	344	2 (192 vs. 152)	26.7	50.7	4
Enzyme	600	6 (100 each)	32.6	124.3	9
D & D	1178	2 (691 vs. 587)	284.4	1921.6	52

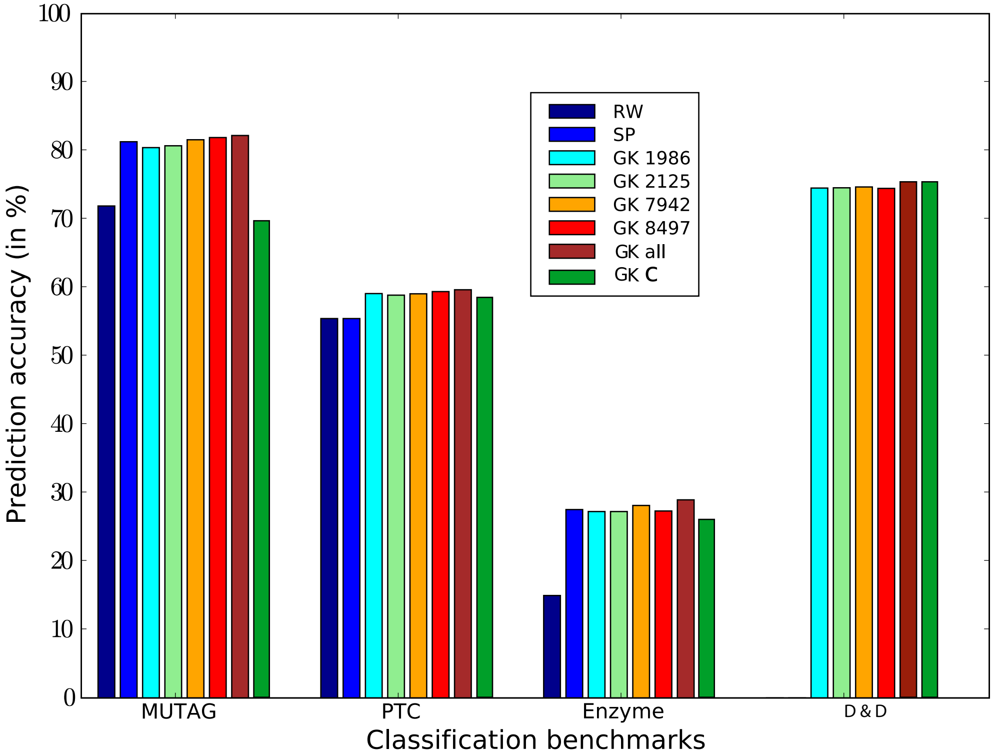
MUTAG, PTC - chemicals

Enzyme, D & D - biological datasets

We did not consider node labels

Experiments

Classification accuracy for $k = 4$



Experiments

Runtime

Kernel	MUTAG	PTC	Enzymes	D & D
RW	42.3”	2’ 39”	10’ 45”	> 1 day
SP	23.2”	2’ 35”	5’ 1”	> 1 day
GK A3 1016	21.5”	29.7”	39”	2’ 9”
GK A3 1154	23.1”	42.6”	48.7”	2’ 19”
GK A3 4061	1’ 18”	2’ 39”	1’ 51”	6’ 35”
GK A3 4615	1’ 38”	3’ 1”	2’ 51”	5’ 58”
GK A3 all	0.35”	0.9”	3.34”	2’ 34”
GK C3	0.14”	0.36”	1.3”	2’ 14”
GK A4 1986	1’ 39”	3’ 2”	4’ 20”	11’ 35”
GK A4 2125	1’ 46”	3’ 16”	4’ 36”	12’ 21”
GK A4 7942	6’ 33”	12’ 3”	16’ 35”	42’ 45”
GK A4 8497	6’ 57”	12’ 49”	17’ 38”	45’ 36”
GK A4 all	4.38”	10.8”	49.3”	2h 44’ 59”
GK C4	0.26”	0.9”	4.1”	35’ 22”
GK A5 5174	3’ 14”	8’ 1”	16’ 57”	1h 29’ 54”
GK A5 5313	3’ 18”	8’ 6”	17’ 3”	1h 1’ 54”
GK A5 20696	8’ 56”	18’ 28”	42’ 2”	1h 30’ 18”
GK A5 21251	9’ 5”	18’ 4”	27’	2h 6’ 45”
GK A5 all	7’ 17”	16h 2’ 16”	20h 26’ 8”	> 1 day
GK C5	0.79”	2.1”	40.7”	> 1 day

Conclusion

- We have proposed efficient graph kernels based on counting or sampling limited size subgraphs in a graph
- Our methods for efficient counting of graph features are not limited to being used in graph kernels
- Future research: take node labels into account