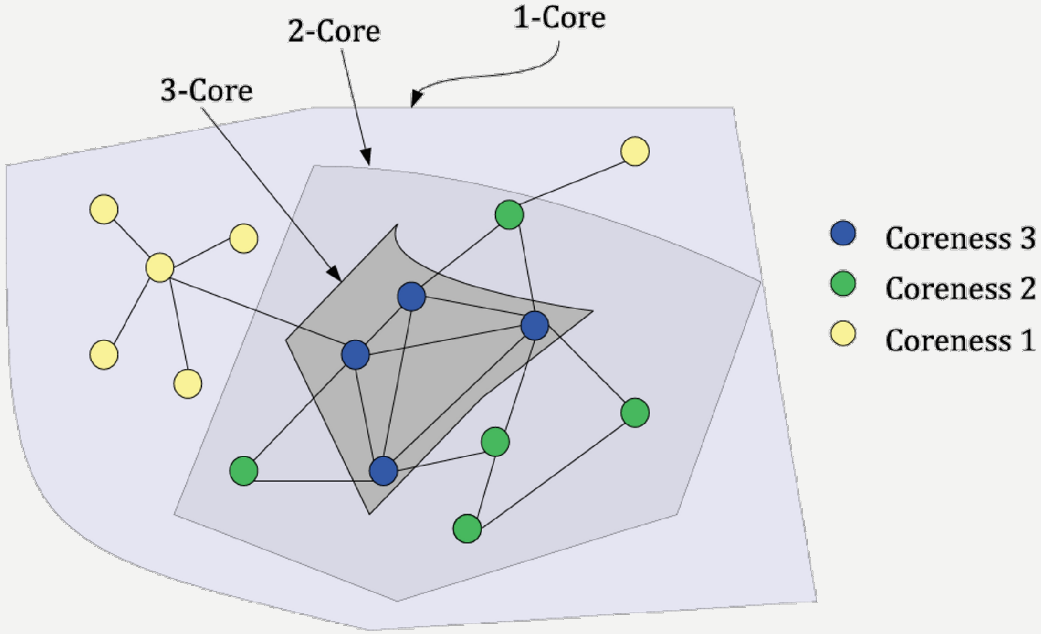
A Degeneracy Framework for Graph Similarity

图退化(Degeneracy)



- k-shell：属于k-core但不属于(k+1)-core的节点构成集合称为k-shell
- coreness k：节点的coreness为 k 等价于此节点在网络的k-shell集中

k-core 基础算法

```
for i = 1 2… (k-1) do:
    while there're nodes with degree less than or equal to i in the graph do:
        remove all nodes with degree less than or equal to i in the graph
    end while
end for
```

Algorithm 1 k-core Decomposition

Input: A graph $G = (V, E)$
Output: A set of k-cores $\mathcal{C}$
$\mathcal{C} = \{V\}$
$k = \min_{v \in V} d(v)$
for $i = 1$ **to** n **do**
    Let v be the vertex with the smallest degree in G
    if $d(v) > k$ **then**
        add V to $\mathcal{C}$
        $k = d(v)$
    end if
    $V = V \setminus \{v\}$
end for

- $C = \{0 - core\ set,\ 1 - core\ set,\ \ldots,\ k - core\ set\}$

k-core & degeneracy

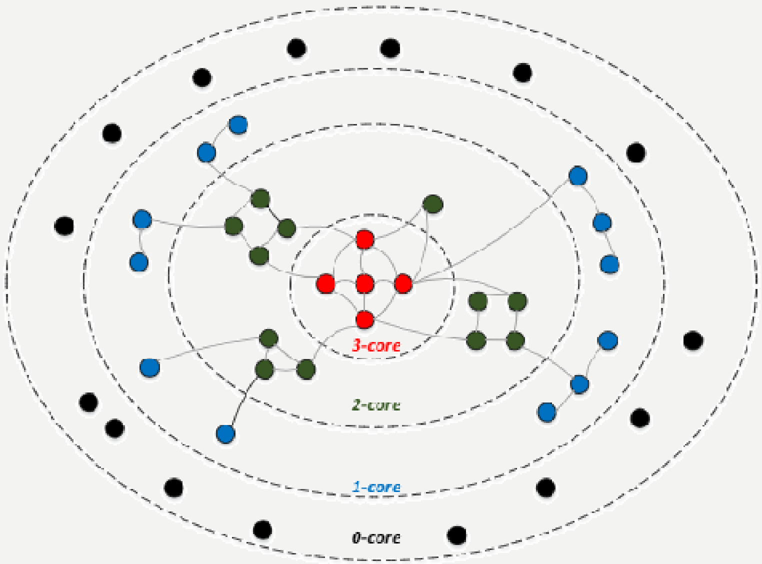


Figure 1: Example of core decomposition of graph.

图G的退化性(degenracy)$\delta^*(G)$定义为图G包含非空k-core子图的最大k， 如上图$\delta^*(G) = 3$

Core-based Kernel

Algorithm 2 Core-based Kernel

Input: A pair of graphs G and G'
Output: Result of the kernel function val
$val = 0$
$\delta^*_{min} = \min\left(\delta^*(G), \delta^*(G')\right)$
Let C_i, C'_i be the i-cores of G, G', for $i = 0, \ldots, \delta^*_{min}$
for $i = \delta^*_{min}$ **to** 0 **do**
  $val = val + kernel(C_i, C'_i)$
end for

Core-based Kernel Calculation

Definition 1. *Let $G = (V, E)$ and $G' = (V', E')$ be two graphs. Let also k be any kernel for graphs. Then, the core variant of the base kernel k is defined as*

$$k_c(G, G') = k(C_0, C'_0) + k(C_1, C'_1) + \ldots + k(C_{\delta^*_{min}}, C'_{\delta^*_{min}}) \quad (1)$$

where δ^*_{min} is the minimum of the degeneracies of the two graphs, and $C_0, C_1, \ldots, C_{\delta^*_{min}}$ and $C'_0, C'_1, \ldots, C'_{\delta^*_{min}}$ are the 0-core, 1-core,..., δ^*_{min}-core subgraphs of G and G' respectively.

实验

- Core-based Kernel本身指定base kernel，作者采用了graphlet、shortest path、WL-subtree和pyramid match四种base kernel进行实验；
- Kernel本身是用来计算图的相似性的，因此也可以用来做图分类；在支持向量机上应用图核监督学习可以实现图分类；
- 另外作者也提到k-core分解的作用可以用于对图降维；例如直接计算复杂图的图核需要很长时间，而计算k-core子图能够在提供一定准确率的条件下大幅降低计算时间。

Method \ Dataset	MUTAG	ENZYMES	NCI1	PTC-MR	D&D
GR	69.97 (± 2.22)	33.08 (± 0.93)	65.47 (± 0.14)	56.63 (± 1.61)	77.77 (± 0.47)
CoRE GR	**82.34** (± 1.29)	33.66 (± 0.65)	**66.85** (± 0.20)	57.68 (± 1.26)	78.05 (± 0.56)
SP	84.03 (± 1.49)	40.75 (± 0.81)	72.85 (± 0.24)	60.14 (± 1.80)	77.14 (± 0.77)
CoRE SP	**88.29** (± 1.55)	41.20 (± 1.21)	**73.46** (± 0.32)	59.06 (± 0.93)	77.30 (± 0.80)
WL	83.63 (± 1.57)	51.56 (± 2.75)	84.42 (± 0.25)	61.93 (± 2.35)	79.19 (± 0.39)
CoRE WL	**87.47** (± 1.08)	47.82 (± 4.62)	**85.01** (± 0.19)	59.43 (± 1.20)	79.24 (± 0.34)
PM	80.66 (± 0.90)	42.17 (± 2.02)	72.27 (± 0.59)	56.41 (± 1.45)	77.34 (± 0.97)
CoRE PM	**87.19** (± 1.47)	42.42 (± 1.06)	**74.90** (± 0.45)	**61.13** (± 1.44)	77.72 (± 0.71)

Method \ Dataset	IMDB BINARY	IMDB MULTI	REDDIT BINARY	REDDIT MULTI-5K	REDDIT MULTI-12K
GR	59.85 (± 0.41)	35.28 (± 0.14)	76.82 (± 0.15)	35.32 (± 0.09)	22.68 (± 0.18)
CoRE GR	**69.91** (± 0.19)	**47.34** (± 0.84)	**80.67** (± 0.16)	**46.77** (± 0.09)	**32.41** (± 0.08)
SP	60.65 (± 0.34)	40.10 (± 0.71)	83.10 (± 0.22)	49.48 (± 0.14)	35.79 (± 0.09)
CoRE SP	**72.62** (± 0.59)	**49.43** (± 0.42)	**90.84** (± 0.14)	**54.35** (± 0.11)	**43.30** (± 0.04)
WL	72.44 (± 0.77)	51.19 (± 0.43)	74.99 (± 0.57)	49.69 (± 0.27)	33.44 (± 0.08)
CoRE WL	**74.02** (± 0.42)	51.35 (± 0.48)	**78.02** (± 0.23)	50.14 (± 0.21)	**35.23** (± 0.17)
PM	68.53 (± 0.61)	45.75 (± 0.66)	82.70 (± 0.68)	42.91 (± 0.42)	38.16 (± 0.19)
CoRE PM	**71.04** (± 0.64)	**48.30** (± 1.01)	**87.39** (± 0.55)	**50.63** (± 0.50)	**42.89** (± 0.14)

加速比

	MUTAG	ENZYMES	NCI1	PTC-MR	D&D	IMDB BINARY	IMDB MULTI	REDDIT BINARY	REDDIT MULTI-5K	REDDIT MULTI-12K
SP	1.69x	2.52x	1.62x	1.65x	3.00x	12.42x	17.34x	1.04x	1.05x	1.18x
GR	1.85x	2.94x	1.75x	1.50x	3.44x	7.95x	8.20x	2.24x	2.37x	2.80x
WL	1.76x	2.77x	1.68x	1.62x	3.34x	7.13x	6.84x	1.52x	1.58x	1.54x
PM	1.87x	2.79x	1.68x	1.50x	3.67x	6.92x	6.33x	1.90x	1.98x	1.96x
δ^*_{ave}	2.00	2.98	1.98	1.73	3.96	9.15	8.15	2.33	2.27	2.24

Table 2: Comparison of running times of base kernels vs their core variants. The values indicate the relative increase in running time when compared to the corresponding base kernel.