日期: /

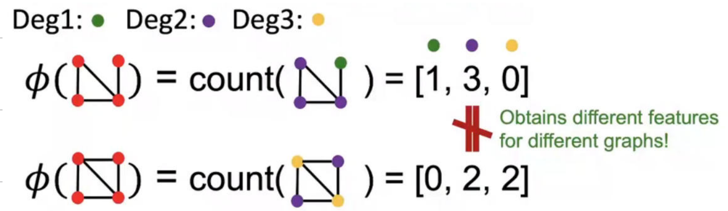
# Graph-Level Features and Graph Kernels

## [Kernel methods]

- 对于图，我们以设计核方法代替设计特征。
- 核函数 $K(G,G')$ 衡量 $G,G'$ 的相似性
- $K=(K(G,G'))_{G,G'}$ 必须半正定
- 必须存在一个特征表示 $\phi(\cdot)$ 使 $K(G,G')=\phi(G)^T\phi(G')$

核函数的目标是设计一个特征向量 $\phi$，而 $\phi$ 是图的一种词袋类型(Bag-of-Words/BoW)的表示。

· Degree Kernel




以不同度的节点数作为特征

· Graphlet Kernel

无需连通，无需根植

以不同图元在图中的出现次数为特征。

日期:　　/

Given graph $G$, and a graphlet list $\mathcal{G}_k = (g_1, g_2, \cdots, g_{n_k})$
define the graphlet count vector $f_G \in \mathbb{R}^{n_k}$ as

$$(f_G)_i = \#(g_i \subseteq G) \text{ for } i = 1, 2, \cdots, n_k$$

对于两个图 $G$, $G'$, 定义图元核为：

$$K(G, G') = f_G^T f_{G'}$$

但 $G$, $G'$ 大小可能相差极大, 所以常进行归一化:

$$h_G = \frac{f_G}{\mathrm{sum}(f_G)}, \quad K(G, G') = h_G^T h_{G'}$$

计算图元核的时间成本极大, 在大小为 $n$ 的图中计算大小为 $k$ 的图元, 枚举时间为 $O(n^k)$

- Weisfeiler-Lehman Kernel

  使用邻域结构来迭代丰富节点词汇

  - 颜色细化算法 (Color Refinement)

  – Given a graph $G$ with a set of nodes $V$.

  – Assign an initial color $c^{(0)}(v)$ to each node.

  – Iteratively refine node colors by

将不同输入映射为不同颜色

$$c^{(k+1)}(v) = \mathrm{HASH}\left(\left\{c^{(k)}(v), \left\{c^{(k)}(u)\right\}_{u \in N(v)}\right\}\right)$$

日期:　　　/

- After k times color refinement, WL kernel counts number of nodes with a given color.
- WL kernel value is computed by the inner product of the color count vectors.

WL核有效且计算成本低