A Survey on Graph Kernels

HAORAN LI 2023/02/24

Kernel Method(1)

- 现实任务中，原始样本空间内也许并不存在一个能正确划分两类样本的超平面，如图6.3所示；
- 对于这样的问题，可以将样本从原始空间映射到一个更高维度的新空间，使得这个样本在新空间内线性可分；
- 映射前：$f(\vec{x}) = \vec{w} \cdot \vec{x} + b$
- 映射后：$f(\vec{x}) = \vec{w} \cdot \phi(\vec{x}) + b$，通常求解其对偶问题

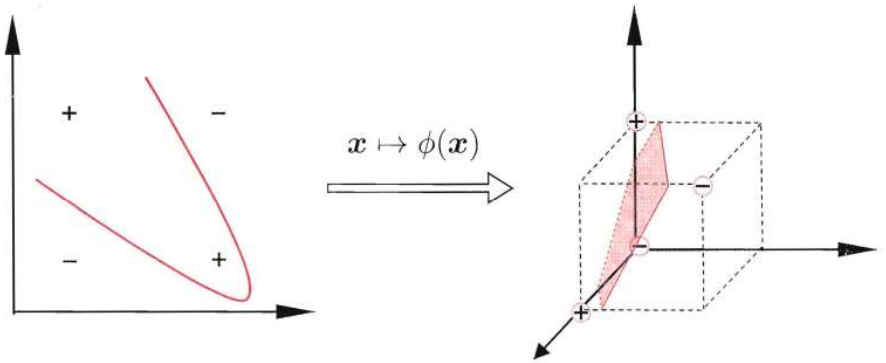


图 6.3 异或问题与非线性映射

$$\max_{\vec{\alpha}} \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j y^{(i)} y^{(j)} \phi(\boldsymbol{x}^{(i)}) \phi(\boldsymbol{x}^{(j)}), \vec{w} = \sum_{i=1}^{m} \alpha_i y^{(i)} \phi(\boldsymbol{x}^{(i)}), \sum_{i=1}^{m} \alpha_i y^{(i)} = 0$$

Kernel Method(2)

$$\max_{\vec{\alpha}} \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j y^{(i)} y^{(j)} \phi(\boldsymbol{x}^{(i)}) \phi(\boldsymbol{x}^{(j)}), \vec{w} = \sum_{i=1}^{m} \alpha_i y^{(i)} \phi(\boldsymbol{x}^{(i)}), \sum_{i=1}^{m} \alpha_i y^{(i)} = 0 (28)$$

- 求解(28)需要涉及到计算$\phi(\boldsymbol{x}^{(i)})\phi(\boldsymbol{x}^{(j)})$，这通常是困难的，但是如果存在函数$\kappa$：

$$\kappa(\boldsymbol{x}^{(i)}, \boldsymbol{x}^{(j)}) = \phi(\boldsymbol{x}^{(i)}) \cdot \phi(\boldsymbol{x}^{(j)})$$

- 计算$\phi(\boldsymbol{x}^{(i)})\phi(\boldsymbol{x}^{(j)})$可以通过计算$\kappa(\boldsymbol{x}^{(i)}, \boldsymbol{x}^{(j)})$间接获得，代入(28)求解$\vec{\alpha}$；最终得到

$$f(\vec{x}) = \vec{w} \cdot \phi(\vec{x}) + b = \sum_{i=1}^{m} \alpha_i y^{(i)} \phi(\boldsymbol{x}^{(i)}) \phi(\vec{x}) + b = \vec{w'} \phi(\vec{x}) + b$$

- 其中$\vec{w'} = (w'_1, w'_2, \ldots, w'_m) = (\alpha_i y^{(1)} \phi(\boldsymbol{x}^{(1)}), \alpha_i y^{(2)} \phi(\boldsymbol{x}^{(2)}), \ldots, \alpha_i y^{(m)} \phi(\boldsymbol{x}^{(m)}))$

Kernel Method(3)

- 简单来说就是

为了样本线性可分，需要将样本从原始空间映射到一个更高维度的空间；通常这个映射很难直接求得；数学知识告诉我们，只需要求解样本空间中任意两点在映射后的空间中的内积，就可以求解映射后的线性目标函数

  - 这个求解任意两点在映射后的空间中的内积的函数称为核函数

Radial Basis Function (RBF) kernel

- 径向基函数 (RBF) 核是一种流行的核函数，用于机器学习和数据分析中聚类、分类和回归等任务

- RBF核也用于基于图的方法中，以衡量图节点之间的相似性或相异性，或者将图数据转换为新的特征空间(feature space)以供后续分析

- 在图的上下文中，RBF 核可以定义如下：给定两个图节点（或节点集），RBF 核测量它们在图中距离的函数作为它们之间的相似性，$K(\boldsymbol{x}, \boldsymbol{y}) = e^{-\frac{\|\boldsymbol{x}-\boldsymbol{y}\|^2}{2\sigma^2}}$，其中x和y分别表示图节点的特征向量(feature vectors)，$\| \boldsymbol{x} - \boldsymbol{y} \|$是特征向量之间的距离

Graph Kernel

- A popular approach to learning with graph-structured data is to make use of graph kernels—functions which measure the similarity between graphs—plugged into a kernel machine, such as a support vector machine

- Graph kernels是度量图的相似性(graph similarity)的函数，常用于学习图结构数据的方法

- Graph kernels用于支持向量机(support vector machine)，可以实现图分类(classification)

R-convolution Kernel

- 给定两个图$G_1(V_1, E_1)$和$G_2(V_2, E_2)$，一种图的分解方式F
- 分解后结构$F(G_1) = \{S_{1,1}, S_{1,2}, \ldots, S_{1,N_1}\}$，$F(G_2) = \{S_{2,1}, S_{2,2}, \ldots, S_{2,N_2}\}$

$$k_{R-conv}(G_1, G_2) = \sum_{n_1=1}^{N_1} \sum_{n_2=1}^{N_2} \delta(S_{1,n_1}, S_{2,n_2})$$

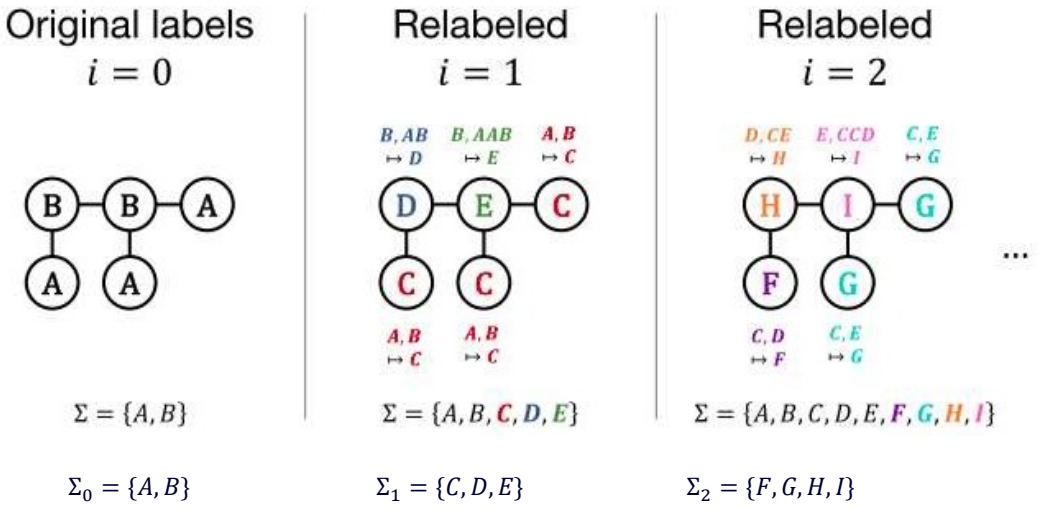
- 其中$\delta(S_{1,n_1}, S_{1,n_2})$在$S_{1,n_1}$和$S_{1,n_2}$同构时为$1$(近似结果)，否则为$0$

Graph Kernel Approaches

1-d Weisfeiler-Lehman subtree kernel

- 如果两个图由具有相似邻域的顶点组成，即它们具有相似的局部结构，则它们被认为是相似的。

$$l^i(v) = relabel(l^{i-1}(v), sort(\{l^{i-1}(u)|u \in N(v)\}))$$



for graph G, feature vector of all h iterati

$$\phi_{WL}(G) = (\phi^0_{\sigma^0_1}(G), \ldots, \phi^0_{\sigma^0_{|\Sigma_0|}}(G), \ldots, \phi^0_{\sigma^h_1}(G), \ldots, \phi^0_{\sigma^h_{|\Sigma}}$$

Weisfeiler-Lehman subtree kernel for h ite

$$k_{WL}(G, H) = \phi_{WL}(G) \cdot \phi_{WL}(H)$$

Optimal assignment kernel

图X, Y具有结构：$X = \{x_1, ..., x_n\}, Y = \{y_1, ..., y_n\}, x_i, y_j \in \mathcal{R}$

定义base kernel，$k: \mathcal{R} \times \mathcal{R} \to \mathbb{R}, \mathbb{R}$*is real number set*

Optimal assign kernel：$K_A(X, Y) = \max_{\pi \in \Pi_n} \sum_{i=1}^{n} k(x_i, y_{\pi(i)})$

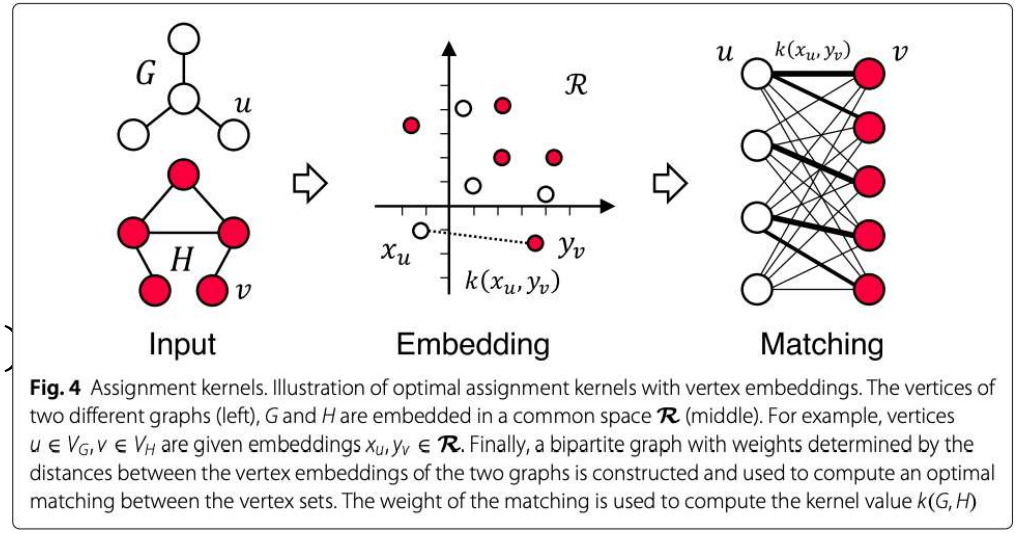


Fig. 4 Assignment kernels. Illustration of optimal assignment kernels with vertex embeddings. The vertices of two different graphs (left), *G* and *H* are embedded in a common space $\mathcal{R}$ (middle). For example, vertices $u \in V_G, v \in V_H$ are given embeddings $x_u, y_v \in \mathcal{R}$. Finally, a bipartite graph with weights determined by the distances between the vertex embeddings of the two graphs is constructed and used to compute an optimal matching between the vertex sets. The weight of the matching is used to compute the kernel value $k(G, H)$

FRÖHLICH H, WEGNER J K, SIEKER F, ET AL. OPTIMAL ASSIGNMENT KERNELS FOR ATTRIBUTED MOLECULAR GRAPHS[C]//PROCEEDINGS OF THE 22ND INTERNATIONAL CONFERENCE ON MACHINE LEARNING. 2005: 225-232.

Vertex/Edge label kernel

$$K_{VL}(G,H) = \sum_{u \in V(G)} \sum_{v \in V(H)} k(l(u), l(v))$$

- Base kernel k is the quality indicator function
- 缺点是忽略了结构和标签之间的相互作用，并且对于未标记的图几乎完全没有作用
- Instead of viewing graphs as bags of vertices or edges, we may view them as bags of subgraph patterns. To this end, Shervashidze et al. introduced a kernel based on counting occurrences of subgraph patterns of a fixed size

Graphlet kernel

- 给定k个顶点，列举出所有的同构类型σ_i，$1 \le i \le N$，即不同类型的图都是异构的
- 计算图G中同构类型σ_i的出现次数$\phi(G)_{\sigma_i}$，feature map $\phi_{GR}(G) = (\phi(G)_{\sigma_1}, \ldots, \phi(G)_{\sigma_N})$
- Graphlet kernel：$k_{GR}(G,H) = \phi_{GR}(G) \cdot \phi_{GR}(G)$

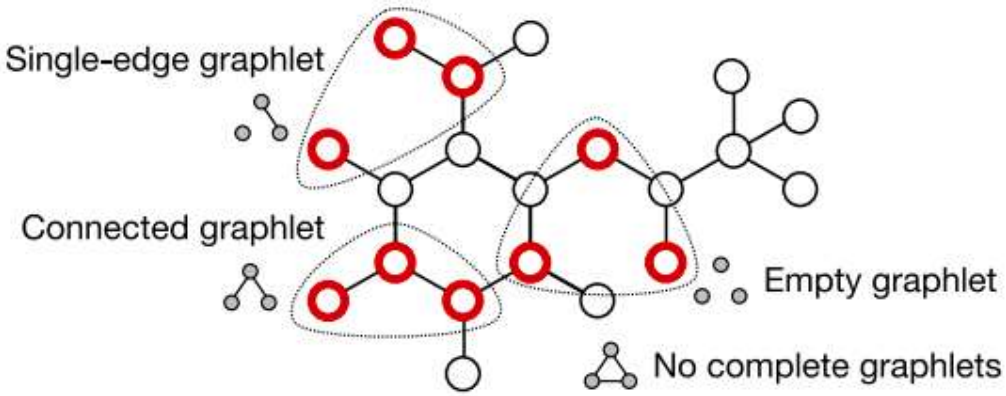


Fig. 5 Graphlets. Illustration of graphlets on 3 vertices in a graph G. Each circle represents a vertex and each line connecting two circles an edge. A 3-graphlet is an instance of an edge pattern on the induced subgraph of 3 vertices. We highlight examples of empty (right), single-edge (top-left), and double-edge (bottom-left) 3-graphlets. No complete graphlets are present in the graph. The graphlet kernel is computed by comparing the number of instances of each pattern in two graphs

SHERVASHIDZE N, VISHWANATHAN S V N, PETRI T, ET AL. EFFICIENT GRAPHLET KERNELS FOR LARGE GRAPH COMPARISON[C]//ARTIFICIAL INTELLIGENCE AND STATISTICS. PMLR, 2009: 488-495.

Shortest-path Kernel

$$k_{SP}(G,H) = \sum_{(u,v)\in V(G)^2, u\neq v} \sum_{(w,z)\in V(H)^2, w\neq z} k((u,v),(w,z))$$

$$k((u,v),(w,z)) = k_L(l(u),l(w)) \cdot k_L(l(v),l(z)) \cdot k_D(d(u,v),d(w,z))$$

- k_L是用于比较顶点标签(vertex label)的核
- k_D是用于比较两个顶点最短路径距离的核

BORGWARDT K M, KRIEGEL H P. SHORTEST-PATH KERNELS ON GRAPHS[C]//FIFTH IEEE INTERNATIONAL CONFERENCE ON DATA MINING (ICDM'05). IEEE, 2005: 8 PP.

Direct Product of Graphs

$G \times H = (\mathcal{V}, \mathcal{E})$ 笠图 $G = (V, E), H = (V', E')$,

图的直积

$$\mathcal{V} = \{(v, v') \in V \times V' \mid l(v) = l(v')\}$$

$$\mathcal{E} = \{((u, u'), (v, v')) \in \mathcal{V} \mid (u, v) \in E \wedge (u', v') \in E' \wedge l((u, v)) = l((u', v'))\}$$

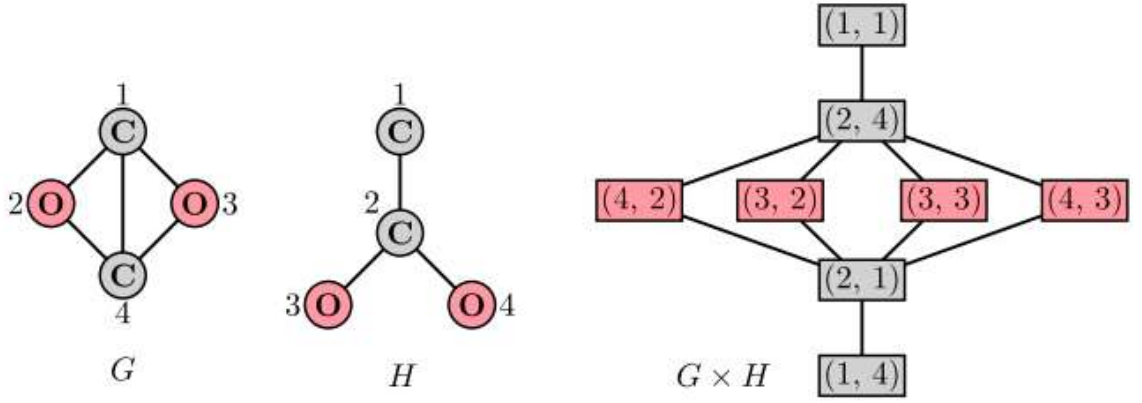


Fig. 6 Direct product graph. Two labeled graphs *G*, *H* and their direct product graph $G \times H$. The vertices of *G* and *H* are labeled with 'C' (gray) and 'O' (red). In the direct product graph, there is a vertex for all pairs of vertices of *G* and *H* with the same label. Two vertices in the direct product graph are adjacent if and only if the associated pairs of vertices are adjacent in *G* and *H*

Direct Product & Geometric Random Walk kernel

$G \times H = (\mathcal{V}, \mathcal{E})$

- The direct product kernel is then defined as

$$K_{\mathrm{RW}}(G,H) = \sum_{i,j=1}^{|\mathcal{V}|} \left[\sum_{l=0}^{\infty} \lambda_l A_{\times}^{l} \right]_{ij}$$

- $\boldsymbol{A}_{\times}$表示直积图的邻接矩阵
- λ_l需要使得级数收敛

- The geometric random walk kernel is

$$K_{\mathrm{GRW}}(G,H) = \sum_{i,j=1}^{|\mathcal{V}|} \left[(\boldsymbol{I} - \gamma \boldsymbol{A}_{\times})^{-1} \right]_{ij}$$

Graph Kernel Conclusion

Expressivity

- Complete graph kernel对应的feature map是一个单射(injection)
- 如果一个graph kernel不是complete的，则表示存在两个非同构(non-isomorphic)的图G, H，使得$\phi(G) = \phi(H)$，即这两个图不能被此kernel区分
- 特别是Oneto等人，从统计的角度研究了图内核的表达性

Applications