

图编辑距离概述

徐周波 张 鷟 宁黎华 古天龙

(桂林电子科技大学广西可信软件重点实验室 广西 桂林 541004)

摘 要 图编辑距离是图模式匹配技术中常用的方法之一。基于图编辑距离的匹配方法能够处理多种类型的图数据,因而受到了学术界的广泛关注。首先介绍了图编辑距离的相关概念;然后简述了基于启发式搜索技术的精确图编辑距离算法,重点分析了基于二分图匹配的近似图编辑距离算法;最后对现存的一些图编辑问题进行了总结,并对未来的发展趋势进行了展望。

关键词 图编辑距离,二分图匹配, A^* 算法, Hausdorff 匹配

中图分类号 TP391 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2018.04.002

Summary of Graph Edit Distance

XU Zhou-bo ZHANG Kun NING Li-hua GU Tian-long

(Guangxi Key Laboratory of Trusted Software, Guilin University of Electronic Technology, Guilin, Guangxi 541004, China)

Abstract Graph edit distance is one of the most flexible and general graph pattern matching models available. This matching method has provoked wide concern from scholars owing to its capability to handle many kinds of graph data. Firstly, the related concepts of graph edit distance were introduced. Then the exact graph edit distance algorithms based on heuristic search technology were described briefly, and the inexact edit distance algorithms of bipartite graph matching was emphatically analyzed. Finally, some existing problems were summarized, and the future development trend was simply discussed.

Keywords Graph edit distance, Bipartite graph matching, A^* algorithm, Hausdorff matching

1 引言

由于图能很好地表示对象的属性以及对对象之间的关系,因此基于图的方法被广泛应用于模式识别领域^[1]。其中,图模式匹配技术^[2]作为实现图数据上高效查询的重要方法之一,被广泛应用于如生物信息学^[3]、图像分类^[4]及计算机网络分析^[5]等各个领域的实际问题中。

近年来,研究者们提出了大量的图匹配算法^[1,6]。其中,基于图编辑距离^[7](Graph Edit Distance, GED)的方法因具有良好的容错性且适用于多种类型的图,如有向图、无向图、属性图、无属性图等,受到了极大的关注。图编辑距离指将源图编辑成目标图所需的最小编辑操作代价值之和。其中,每步操作的代价值通过定义相应的代价函数得出,代价值即距离值。图编辑距离主要通过度量图之间的不相似度来衡量图之间的匹配度,而不相似度则是通过距离值进行衡量,距离值越大则不相似度越大,匹配度越低;反之亦然。根据求解的距离值是否精确,图编辑距离算法主要可分为两类^[8]:精确图编辑

距离(Exact Graph Edit Distance)和近似图编辑距离(Inexact Graph Edit Distance)。衡量图编辑距离算法优劣的主要指标是时间复杂度和分类实验的分类精确率。

在使用基于图编辑距离的方法进行图匹配时,仍面临着以下诸多问题:1)编辑操作的代价函数定义困难。代价函数反映相应编辑操作作用于底图的变形强度,而对分布于不同应用领域中的图数据,若相应编辑操作强度不同,其代价函数的定义也应不同。2)求解精确图编辑距离算法的复杂度较高。最坏情况下,其时间复杂度随图数据的规模呈指数级增长,且占用大量计算空间。3)近似图编辑距离算法的精确度不足。平衡时间复杂度和精确度是近似算法面临的主要问题。

针对以上 3 类问题,学者们主要从以下方面展开研究:1)针对代价函数定义问题,一方面设计了直接定义代价函数的方法,根据图标签的相关属性进行代价函数的定义;另一方面提出了众多学习方法,学习训练出代价函数模型^[9],自动生成代价函数。同时,对代价函数进行了大量的聚类研究,有助

到稿日期:2017-02-14 返修日期:2017-05-09 本文受国家自然科学基金(61572146, 61363030, U1501252, 61762027), 广西自然科学基金(2017GXNSFAA198172, 2015GXNSFAA139285, 2014GXNSFAA118354), 桂林电子科技大学研究生教育创新计划项目(2017YJXC08, 2017YJXC54)资助。

徐周波(1976—),女,博士,副教授,CCF 高级会员,主要研究领域为符号计算、智能规划与约束求解;张 鷟(1993—),男,硕士生,主要研究领域为大数据及智能制造;宁黎华(1981—),女,博士生,主要研究领域为智能计算、约束求解, E-mail: nlhau123@sina.com(通信作者);古天龙(1964—),男,博士,教授,博士生导师,主要研究领域为形式化方法、符号计算、知识工程。

于开发新的、更高效的图算法^[10]。2) 针对精确图编辑距离算法复杂度较高的问题,采用启发式搜索技术以及相关搜索策略^[11-14]等方法,提高搜索效率。3) 针对近似图编辑距离精度偏低的问题,引入二分图匹配^[15]和 Hausdorff 匹配^[16]等思想,同时采用多种算法及策略^[17-18]对其进行进一步优化。目前,这两种思想得到了广泛的研究,已成为计算近似距离的主要方法之一。

本文主要回顾并总结了图编辑距离中相关计算算法以及代价函数的定义。第2节介绍了有关图编辑距离的相关基本概念;第3节对代价函数的定义及研究进行了总结;第4节和第5节详细阐述了精确图编辑距离和近似图编辑距离的相关算法;最后,总结了图编辑距离现存的一些问题,并对未来的发展趋势做了简要的展望。

2 基本概念

2.1 图的定义

令 L_V 和 L_E 分别表示点和边的标签集合,图 g 表示一个四元组,即 $g=(V, E, \mu, \nu)$ 。其中, V 表示点的集合; $E \subseteq V \times V$ 表示边的集合; $\mu: V \rightarrow L_V$ 表示图中顶点到对应标签的映射; $\nu: E \rightarrow L_E$ 表示图中边到对应标签的映射。根据对标签函数的不同定义,可得到属性图或无属性图。

图的边是由其两端的端点来表示的。令点 u 和点 v 表示同一图 g 中的不同两点,两点间的无向边表示为 (u, v) ,此时点 u 到点 v 与点 v 到点 u 的标签一致,即 $\nu(u, v) = \nu(v, u)$;起点为 u 、终点为 v 的有向边表示为 $\langle u, v \rangle$,此时 $\nu(u, v) \neq \nu(v, u)$ 。

2.2 编辑操作

标准的编辑操作有6种,即点或边的删除、插入和替换。假设存在点 u, v 和空点 ϵ ,则点 u 的删除操作表示为 $(u \rightarrow \epsilon)$,点 v 的插入操作表示为 $(\epsilon \rightarrow v)$,点 u 替换为点 v 表示为 $(u \rightarrow v)$;边的3种操作的表示方法与此类似。此外,编辑操作还有点的嵌入和分裂^[19],但只适用于部分领域,本文不加以讨论。

2.3 编辑路径

将图 g_1 转化成图 g_2 所需的一系列编辑操作的序列集合记为 k , e_i 表示第 i 步编辑操作, $k=(e_1, \dots, e_i, \dots, e_n)$,则称 k 为图 g_1 和 g_2 间的完全编辑路径。部分编辑路径是序列 k 的子集,其只能将图 g_1 部分转化。下文提到的编辑路径或路径都为完全编辑路径。

2.4 图编辑距离

编辑距离的概念是从字符串编辑距离^[20]上延伸、拓展到树^[21]和图^[22]的数据结构上的。令源图 $g_1=(V_1, E_1, \mu_1, \nu_1)$,目标图 $g_2=(V_2, E_2, \mu_2, \nu_2)$ 。图编辑距离计算的是从图 g_1 转化为图 g_2 所需的变形强度或相异度,衡量的是两个图之间的不相似性。图编辑距离的定义如式(1)所示^[23]:

$$d_{\lambda_{\min}}(g_1, g_2) = \min_{\lambda \in \gamma(g_1, g_2)} \left\{ \sum_{e_j \in \lambda} c(e_j) \right\} \quad (1)$$

其中, $\gamma(g_1, g_2)$ 表示所有可从图 g_1 转化为图 g_2 的完全编辑路径; $c(e_j)$ 表示编辑操作 e_j 的编辑代价; λ_{\min} 表示所有完全编辑路径中代价和最小的路径。如图1所示,点和边的删除、点的替换构成了一条完全编辑路径 λ ,给每步编辑操作定义一个代价函数,若此时这条路径的代价函数值之和最小,则此代

价值就是图之间的编辑距离。

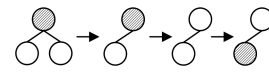


图1 完全编辑路径 λ

Fig. 1 A complete edit path λ

3 图编辑代价

在求解图的编辑距离前,需明确定义各编辑操作的代价函数,代价函数定义的合理性与充分性决定了编辑距离的有效性。换言之,定义给每种编辑操作上的代价值对图匹配结果有着重要的影响,在一种代价函数定义下的匹配结果与在另一种代价函数定义下的匹配结果不一定相同。本文记点的插入、删除、替换的代价函数分别为 c_{ni}, c_{nd}, c_{ns} ;边操作的代价函数表示与点类似,分别为 c_{ei}, c_{ed}, c_{es} 。

3.1 编辑代价函数的定义

大部分情况下,代价函数的定义与图标签距离有关。标签属性主要有3种:数值型、符号型以及这两者的结合^[23]。数值型标签距离一般可利用基于闵可夫斯基距离^[24]或欧氏距离等方法计算;符号型标签距离可利用基于字符串编辑距离^[25]或者狄克拉函数等方法求解;对于数值与符号的结合标签类型,可以分别采用各标签类型的标签距离求解方法,也可直接定义一个非负值而不考虑标签属性。

3.1.1 代价函数的直接定义

代价函数的直接定义是指根据图标签定义代价函数或直接人为设定代价函数。考虑到删除和添加操作是对称的,通常对这两者的操作代价与替换操作代价分别进行定义。

对于点替换操作代价,一种方式是定义 $C_{ns} = \{0, K_{ns}\}$,当两点间的标签距离小于人为设定的某一个阈值时,代价取值为 K_{ns} ,否则直接取为0,如文献^[26-27]定义 $C_{ns} = \{0, \infty\}$,在指纹识别^[28]中定义 $C_{ns} = \{0, 1\}$;另一种定义方式是取决于点上的标签以及一些自定义的参数 θ ^[29-31]。边的替换操作与此类似。对于点和边的删除、插入操作代价,多数情况下都直接赋予一个常量,但也可以根据标签距离进行定义^[32-34]。

直接给定代价函数简单方便,但只能部分反映图之间的属性关系,故其在实验中仍存在少量误差。

3.1.2 代价函数的学习定义

在某些情况下,有关图标签属性的先验知识并不清楚,或者为获得更准确的代价函数,可以利用自主学习的方式学习图标签属性,从而训练出代价模型。

文献^[31]中,图编辑过程被看成一个随机上下文过程,其估算编辑操作的最大似然参数,同时使用最大期望算法学习相应的变形模型,根据这个模型就可以得到所需的代价函数;文献^[35]提出了一种基于编辑操作分布估计的代价推理方法,该方法使用最大期望算法从属性样本图中学习混合密度来生成编辑代价;文献^[36]提出了使用自组织映射系统来表示点和边标签空间的距离测量过程,它的学习过程是基于自组织映射概念的,在学习过程中,它通过使来自相同类的图的相似性增加而来自不同类的图的相似性减小的方式来适配编辑代价;文献^[37]采用专家自定义的方法,并采用人机交互

和自适应的方法定义图编辑代价。

由学习方法训练出的代价模型相较于直接定义更加准确,但训练时需要大量样本图,且在面对不同数据集时需重新训练,因此在具体应用中使用时受到一定限制。

3.2 编辑代价函数的聚类研究

研究代价函数的性质一方面可用于提高现有图算法的效率,另一方面可以利用其开发出更高效的图算法^[10]。对编辑代价函数进行聚类研究能使相关性质更加直观。编辑代价函数同属一类,表示编辑代价函数的定义不同但得到的最终编辑路径相同。

文献[26]研究了代价函数对图之间匹配结果的影响,并根据同条件下的最优匹配结果将编辑代价进行分类;其主要分析了在不同约束条件下编辑距离与最大公共子图的关系、代价函数间的等价性以及更一般情况下的代价函数的定义等问题。定义代价函数 $C=(c_{ni},c_{nd},c_{ns},c_{es})$,另一种代价函数 $C'=(c'_{ni},c'_{nd},c'_{ns},c'_{es})$,当代价函数满足 $c_{nd}+c_{ni}\leq c_{ns}$ 和 $c_{ed}+c_{ei}\leq c_{es}$ 时,求解图编辑距离相当于求解图之间的最大公共子图^[27];若代价函数满足式(2):

$$\frac{c_{ni}+c_{nd}}{c_{ns}}=\frac{c'_{ni}+c'_{nd}}{c'_{ns}} \quad (2)$$

$$\frac{c_{es}}{c_{ns}}=\frac{c'_{es}}{c'_{ns}}$$

则得出结论:

1)在代价函数 C 下得出的最优匹配操作序列与在代价函数 C' 下得出的最优匹配操作序列一致,即相应点或边进行的编辑操作完全一致;

2)在代价函数 C 和 C' 下得出的最优匹配代价值之间存在一个比例系数;

3)最优匹配只取决于相关编辑代价的比例,而不是一个单一的编辑操作代价值。

文献[10]在文献[26]的基础上进一步研究了编辑代价间的相互关系,重点探究了在 $c_{nd}=c_{ni}=K_n, c_{ed}=c_{ei}=K_e (K_n, K_e \in \mathbb{R}^+)$ 时编辑代价函数的代价聚类问题。其提出了表示代价类别的新方法,并给出以下3个特性:

1)在由 K_n 和 K_e 构成的二维代价空间中,任何代价类的空间要么为空,要么呈现出多边形形状;

2)任一代价类中,若删除和插入操作的数量分别一致,且替换操作代价相同,则得到唯一一条编辑路径;

3)随 K_n 和 K_e 值的增大,编辑距离单调增加。

同时,作者使用相关实验对上述性质进行了验证说明,并指出,在约束条件过多的情况下探索编辑代价间的关系时,图之间大量的同构信息并不能被很好地反映出来。进一步的研究可以在松弛相关约束条件下展开。

4 精确图编辑距离

精确图编辑距离在求解过程中易受到噪声干扰,且其计算复杂度为指数级,因此只在少量对结果要求无误的应用中使用。

4.1 基于启发式搜索技术的精确图编辑距离

在精确图编辑距离的启发式搜索求解算法中,基于 A^*

算法的搜索技术^[11-12,38-40]使用得较为广泛。 A^* 算法^[41]是一类避免状态数目爆炸的经典方法,也是解决最短路径等问题的常用启发式算法之一。

基于 A^* 搜索技术的基本思路是通过逐个处理图中的点,使每个点可能的编辑操作都被考虑,动态构建出一个搜索树,遍历搜索树的过程等同于一个编辑路径求解的过程。搜索树中的内部节点相当于部分编辑路径,叶子节点相当于完全编辑路径。算法1给出精确图编辑距离算法的伪代码^[23]。

算法1 精确图编辑距离算法

输入:非空图 $g_1=(V_1,E_1,\mu_1,\nu_1),g_2=(V_2,E_2,\mu_2,\nu_2)$;其中, $V_1=\{u_1,\dots,u_n\},V_2=\{v_1,\dots,v_m\}$

输出:从图 g_1 到图 g_2 的最小代价路径 λ_{\min}

1. 初始化 OPEN 集合为空集;
2. 对于点 $w \in V_2$, 将替换操作 $\{u_1 \rightarrow w\}$ 插入 OPEN 集合;
3. 插入删除操作 $\{u_1 \rightarrow \epsilon\}$ 至 OPEN 集合;
4. loop
5. $\lambda_{\min} = \arg \min_{\lambda \in \text{OPEN}} \{g(\lambda) + h(\lambda)\}$, 从 OPEN 集合中提取 λ_{\min} ;
6. if λ_{\min} 是一个完全编辑路径 then
7. 返回 λ_{\min} 作为最终结果;
8. else
9. $\lambda_{\min} = \{u_1 \rightarrow v_{q_1}, \dots, u_k \rightarrow v_{q_k}\}$
10. if $k < n$ then
11. 对点 $w, w \in V_2 \setminus \{v_{q_1}, \dots, v_{q_k}\}$
12. 将 $\lambda_{\min} \cup \{u_{k+1} \rightarrow w\}$ 插入 OPEN 集合;
13. 将 $\lambda_{\min} \cup \{u_{k+1} \rightarrow \epsilon\}$ 插入 OPEN 集合;
14. else
15. 将 $\lambda_{\min} \cup w \in V_2 \setminus \{v_{q_1}, \dots, v_{q_k}\} \{ \epsilon \rightarrow w \}$ 插入 OPEN 集合;
16. end if
17. end if
18. end loop

算法1中,两个图的标号随机生成。算法首先初始化一个空集 OPEN,存储遍历过的所有点的操作及相应的代价值。第12行表示点的替换操作,第13行表示点的删除操作,这两个操作是同时考虑的,并且这两个操作在搜索树中产生了大量后继节点。算法第15行表示在对图 g_1 中所有点操作完成后,插入图 g_2 所剩余的点。最终得出图的编辑路径 λ_{\min} ,并依此得出距离值。

启发式搜索中,最重要的一步是如何选取下一步搜索的方向,其由算法中的启发函数决定,如算法1中的第5行所示。 A^* 算法的启发式函数如式(3)所示:

$$f(\lambda) = g(\lambda) + h(\lambda) \quad (3)$$

其中, $g(\lambda)$ 表示从起始点至目前所在的搜索点 λ 的累积距离值, $h(\lambda)$ 表示从目前所在的 λ 点至终点所估计的距离值。比较函数 $f(\lambda)$ 值的大小,选择函数值最小的点作为下一步的搜索点。如此迭代循环,直至搜索结束。由此,可以注意到启发式函数的定义唯一影响算法结果,尤其是估值函数 $h(\lambda)$ 的定义。为确保结果最优,必须保证估值函数 $h(\lambda)$ 不大于从目前所在的 λ 点至终点的实际距离值 $h'(\lambda)$,即 $h(\lambda) \leq h'(\lambda)$ 。

4.1.1 基于一般 A^* 算法的精确图编辑距离

A^* 算法中,当估值函数 $h(\lambda)$ 为0时,其相当于宽度优先搜索算法,至多需遍历完整个搜索树空间才能得到最优解,因此

称此搜索方法为 Plain-A* 搜索^[42]。其时间复杂度为指数级。

4.1.2 基于改进 A* 算法的精确图编辑距离

文献[39]使用 BP 代价阵框架的启发式函数代替了函数 $h(\lambda)$, 降低了启发式函数的时间复杂度, 加快搜索速度的同时能得出精确解。同时, 这种启发式函数的定义方法对于用其他图表示的多种机器学习或模式识别算法具有潜在的指导意义。

文献[40]利用 Hausdorff 编辑距离函数作为 A* 算法的启发式函数, 其优势在于其启发式函数的计算时间复杂度较低, 为 $O(n^2)$ 。实验结果也证实, 相比于 Plain-A* 搜索, 其在保证能得到最优结果的情况下, 大幅度减少了耗时。

4.2 基于深度优先搜索的精确图编辑距离

文献[13]提出了基于上下界剪枝策略的深度优先算法, 称为 DF 算法。与一般 A* 算法相比, DF 算法大幅度减小了所需拓展的集合空间, 减少了算法的计算时间。

文献[14]在 DF 算法的基础上, 使用分布式方法解决图编辑距离问题, 称为 D-DF 算法。其基本思路是将大量图数据分割成子图, 再将子图发送到各工作站利用 DF 算法求解, 最终汇集结果至主控制站。但算法的缺点在于, 由于工作站执行效率不同会导致部分工作站计算完成后出现空闲, 从而影响整体的求解时间。

5 近似图编辑距离

由于精确图编辑距离复杂度较高, 使得精确算法只适用于小规模图, 因此复杂度较低的近似图匹配技术受到学术界极大的关注。

5.1 二分图编辑距离

文献[15]提出将图编辑距离问题转化为基于指派的二分图匹配问题。在将源图 g_1 和目标图 g_2 边编辑代价整合进点编辑代价的情况下, 不考虑边结构, 将 g_1 和 g_2 中的点视为同一个新图中的点, 图 g_1 和 g_2 之间点的所有可能指派视为此新构成的图的边, 此时指派代价即为编辑操作代价, 模型化出一个完全二分图, 并构建一个二分代价阵框架 (BP 代价阵框架)。近年来, 多数近似图编辑距离问题的求解都是基于 BP 代价阵框架进行的。

5.1.1 基于二次指派问题的二分图编辑距离

二次指派问题^[43] (QAPs) 是一类 NP 完全问题。假设存在两个互斥集合 S 和 Q , $S = \{s_1, s_2, \dots, s_n\}$, $Q = \{q_1, q_2, \dots, q_n\}$, 二次指派问题指的是在一些边界约束 (计算要求) 下将集合 S 中的元素一一映射到集合 Q 中。在完成问题的求解后, Q 中元素重新排序, 成为集合 $\varphi = (\varphi_1, \dots, \varphi_n)$, 此时两集合元素一一映射, 即元素 $s_1 \in S$ 映射至 $q_{\varphi_1} \in Q$, 元素 $s_2 \in S$ 映射至 $q_{\varphi_2} \in Q$ 等。但将图编辑距离问题转化为二次指派问题时, 需要解决两个主要的先决条件:

1) 二次指派问题中, 两集合 (如 S, Q) 中元素基数一致, 但在图编辑距离问题中所涉及的两个图 (g_1 和 g_2) 的顶点数或边数大多数情况下都不相同;

2) 二次指派问题中, 两集合中元素间的映射都为双射, 而图编辑距离问题是一个更加常见的指派问题, 因其除了替换操作, 还有删除和添加操作, 存在对空点的操作, 不能达到映

射为双射的要求。

通过分别对图 g_1 和 g_2 添加一定数量的空点 ϵ , 能同时解决上述两个问题^[23]。令图 g_1 的顶点数 $|V_1| = n$, 图 g_2 的顶点数 $|V_2| = m$, 则可以分别添加空点 ϵ 到两个图中, 使得:

$$\begin{aligned} V_1^+ &= V_1 \cup \{\epsilon_1, \dots, \epsilon_m\} \\ V_2^+ &= V_2 \cup \{\epsilon_1, \dots, \epsilon_n\} \end{aligned} \quad (4)$$

图 g_1 和 g_2 可分别表示为 $g_1 = (V_1^+, E_1, \mu_1, \nu_1)$, $g_2 = (V_2^+, E_2, \mu_2, \nu_2)$ 。

基于以上两图的顶点集合, 可建立代价矩阵 C ^[15]:

$$C = \begin{array}{c} \begin{array}{cccc} \nu_1 & \nu_2 & \dots & \nu_m \end{array} \\ \begin{array}{cccc} \epsilon_1 & \epsilon_2 & \dots & \epsilon_n \end{array} \end{array} \begin{array}{c} \begin{array}{cccc|cccc} u_1 & c_{11} & c_{12} & \dots & c_{1m} & c_{1\epsilon} & \infty & \dots & \infty \\ u_2 & c_{21} & c_{22} & \dots & c_{2m} & \infty & c_{2\epsilon} & \ddots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \ddots & \infty \\ u_n & c_{n1} & c_{n2} & \dots & c_{nm} & \infty & \dots & \infty & c_{n\epsilon} \\ \hline \epsilon_1 & c_{\epsilon 1} & \infty & \dots & \infty & 0 & 0 & \dots & 0 \\ \epsilon_2 & \infty & c_{\epsilon 2} & \ddots & \vdots & 0 & 0 & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \infty & \vdots & \ddots & \ddots & 0 \\ \epsilon_m & \infty & \dots & \infty & c_{\epsilon m} & 0 & \dots & 0 & 0 \end{array} \end{array}$$

其中, c_{ij} 表示点替换操作 ($u_i \rightarrow \nu_j$) 的代价值, $c_{i\epsilon}$ 表示点删除操作 ($u_i \rightarrow \epsilon$) 的代价值, $c_{\epsilon j}$ 表示点插入操作 ($\epsilon \rightarrow \nu_j$) 的代价值。

可知, 上述代价阵左上角部分元素表示所有点的替换代价; 右上角部分元素表示点删除的代价值; 左下角部分元素表示点插入的代价值; 右下角部分元素表示 ($\epsilon \rightarrow \epsilon$) 的代价值, 故全为 0。由于每个点只能删除或插入一次, 因此右上角和左下角元素中只有对角线元素为实数, 其余元素直接设定为 ∞ 。

二次指派问题的图编辑路径公式如下^[23]:

$$(\varphi_1, \dots, \varphi_{(n+m)}) = \arg \min_{(\varphi_1, \dots, \varphi_{(n+m)}) \in \mathcal{D}_{(n+m)}} \left[\sum_{i=1}^{n+m} c_{i\varphi_i} + \sum_{i=1}^{n+m} \sum_{j=1}^{n+m} c(a_{ij}) \rightarrow b_{\varphi_i \varphi_j} \right]$$

其中, $\mathcal{D}_{(n+m)}$ 表示图标号序列 $(1, 2, \dots, (n+m))$ 所有的 $(n+m)!$ 种可能的编辑排序。

在基于二次指派问题的图编辑距离中, 首先对点的代价进行优化, 得出点的最优编辑路径, 再根据点最优编辑路径确定边的编辑路径, 并不是动态考虑点和边的匹配, 因此其比精确编辑距离得出的值稍大。

文献[44]在二分图编辑距离的基础上提出另一种二次分配模型, 其时间复杂度在最坏情况下为 $O(n^4)$ 。实验结果表明, 在处理 500 个顶点以内的图时, 该模型的结果接近精确解。

5.1.2 基于线性和分配问题的二分图编辑距离

线性和分配问题 (LSAPs) 类似于二次指派问题, 也是对序列中的元素进行重组的一类分配问题。然而, 与二次指派问题的不同在于, 其只对线性项进行重组, 即线性和分配问题只能处理无边界约束条件下的单一代价阵。

但是, 在线性和分配问题中, 如果未考虑对二次项的求解, 也就意味着没有考虑图中点之间的边结构关系, 使得问题的求解并不完整。因此, 对于线性和分配问题中的代价阵元素, 需要添加相应边的编辑操作代价, 构建一个新的类似于代价阵 C 的代价阵 C^* 。对于替换操作代价, 对相应点的相邻边再建立一个类似 BP 代价阵的矩阵, 先行求解边的最小编辑距离, 再将其加入点替换操作代价值中, 构成代价阵 C^* 左

上角区域的替换代价;代价阵 C^* 中的删除代价表示删除点及其所有相邻边的代价和;对于插入代价,代价阵 C^* 中则更新为插入点及其所有相邻边的代价和。即线性和分配问题中,代价阵元素为^[23]:

$$c_{ij}^* = c_{ij} + \min_{(\varphi_1, \dots, \varphi_{i+m}) \in \mathcal{D}_{(i+m)}} \sum_{k=1}^{n+m} c(a_{ik} \rightarrow b_{j\varphi_k}) \quad (5)$$

可知,线性和分配问题中建立的代价阵与二次指派问题所建代价阵的结构及元素表示的功能类似,只在元素构建时稍有差异。

5.1.3 基于二分代价阵的改进矩阵框架

文献[45]基于 BP 代价阵框架,利用指派问题矩阵的一个性质,即若从指派问题系数矩阵的某行或某列分别减去一个常数 k ,得到一个新的矩阵,其与原矩阵具有相同的最优解,从而构建了一种 $n \times m$ 阶的新代价阵,称为 FBP 算法。但构建此代价阵时,需满足距离量度的 4 条公理,即:

- 1) 同一点或边的替换代价不大于其删除再插入的代价和(三角不等式);
- 2) 同一点或边的删除代价与插入代价相等(对称性);
- 3) 在属性图中,替换前后的点或边标签相同,其替换代价值为零(同一模糊);
- 4) 所有操作的代价值为非负数(非负性)。

其使用的求解算法与对 BP 代价阵的求解算法类似^[46]。FBP 算法构建的代价阵如下:

$$C^{FBP} = n \begin{Bmatrix} \overbrace{c_{11} - (c_{1\epsilon} + c_{\epsilon 1}) \quad \cdots \quad c_{1m} - (c_{1\epsilon} + c_{\epsilon m})}^m \\ \vdots \\ \vdots \\ \underbrace{c_{n1} - (c_{n\epsilon} + c_{\epsilon 1}) \quad \cdots \quad c_{nm} - (c_{n\epsilon} + c_{\epsilon m})}_m \end{Bmatrix}$$

文献[47]基于 BP 代价阵提出了一种新的代价阵,其求解算法称为 SFBP 算法。算法的前提也是需满足上述距离量度的 4 条公理。当 $m \geq n$ 时,新建代价阵如下:

$$C_{m \geq n} = m \begin{Bmatrix} \overbrace{c_{11} \quad c_{12} \quad \cdots \quad c_{1m}}^m \\ c_{21} \quad c_{22} \quad \cdots \quad c_{2m} \\ \vdots \\ c_{n1} \quad c_{n2} \quad \cdots \quad c_{nm} \\ \hline c_{\epsilon 1} \quad c_{\epsilon 2} \quad \cdots \quad c_{\epsilon m} \\ c_{\epsilon 1} \quad c_{\epsilon 2} \quad \cdots \quad c_{\epsilon m} \\ \vdots \\ c_{\epsilon 1} \quad c_{\epsilon 2} \quad \cdots \quad c_{\epsilon m} \end{Bmatrix}$$

当 $n \geq m$ 时,新建代价阵如下:

$$C_{n \geq m} = n \begin{Bmatrix} \overbrace{c_{11} \quad c_{12} \quad \cdots \quad c_{1m}}^m & c_{1\epsilon} & c_{1\epsilon} & \cdots & c_{1\epsilon} \\ c_{21} & c_{22} & \cdots & c_{2m} & c_{2\epsilon} & c_{2\epsilon} & \cdots & c_{2\epsilon} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ c_{n1} & c_{n2} & \cdots & c_{nm} & c_{n\epsilon} & c_{n\epsilon} & \cdots & c_{n\epsilon} \end{Bmatrix}$$

其中,各项元素的含义与线性和分配问题中的元素含义一致。

文献[48]基于代价阵 C^* 提出了效益矩阵的概念。此时,矩阵框架中的元素并不是代价值,而是对矩阵框架中行和列的元素分别进行以下处理:

$$\begin{aligned} row\text{-}utility_{ij} &= \frac{row\text{-}max_i + row\text{-}min_i - 2c_{ij}^*}{row\text{-}max_i - row\text{-}min_i} \\ col\text{-}utility_{ij} &= \frac{col\text{-}max_j + col\text{-}min_j - 2c_{ij}^*}{col\text{-}max_j - col\text{-}min_j} \end{aligned} \quad (6)$$

其中, $row\text{-}max_i$ 和 $row\text{-}min_i$ 分别是代价阵 C^* 中第 i 行的最大元素和最小元素;列元素的定义与此类似。新建的效益矩阵中的第 i 行第 j 列的元素是 $(row\text{-}utility_{ij} + col\text{-}utility_{ij})$ 。此方法提出的目标在于提高选择一个更好的点编辑操作的可能性,其采用的求解算法与 BP 代价阵的求解一致。通过实验将其与基于 C^* 代价阵的 BP 算法和基于 C^* 代价阵的贪心算法进行比较,结果表明基于效益阵的方法能在提高距离精度的同时几乎不对算法的计算时间产生影响。

此外,文献[17]提出并比较了 8 种构建 BP 代价阵元素的方法,囊括了从最简单的、只包含点信息的元素的构建,到最复杂的、包含整团或特征向量结构信息的元素的构建方法。实验证明,针对不同数据类型采用不同构建元素的方法,匹配结果最为有效。

5.1.4 二分图编辑距离求解算法及策略

对于匹配问题,存在众多的求解算法^[49]。将 BP 代价阵及其改进矩阵框架的求解算法统称为 BP 算法(Bipartite Algorithm)。文献[50]基于代价阵 C^* ,分别从理论与实验上比较了 Hungarian 算法^[51]、Munkres 算法^[52]、Volgenant-Jonker 算法^[53-54]的计算效率,尽管理论上三者的时间复杂度同为 $O(n^3)$,但 Volgenant-Jonker 算法能同时适用于稀疏图和稠密图,且实验结果表明其效率要优于其他两种算法。文献[55]提出了基于贪婪指派问题的算法,其算法时间复杂度为 $O(n^2)$;同时针对计算时节点的排序对结果的影响问题提出了 3 种规则,对贪婪指派算法加以改进,优化出 4 种改进算法,实验表明其精确度相较于 BP 算法依然保持良好。

文献[56]使用了 6 种搜索策略对代价阵进行求解,它们分别是迭代搜索、浮动搜索、遗传搜索、贪婪搜索、基于交换方法的遗传搜索、束搜索等。使用这些搜索策略能大大提高计算精度。

表 1 总结了以上部分算法的计算复杂度。

表 1 相关算法的时间复杂度
Table 1 Time complexity of related algorithms

Reference	Algorithm	Time
文献[15]	BP	$O(n+m)^3$
文献[45]	FBP	$O(\max(n,m))^3$
文献[47]	SFBP	$O(\max(n,m))^3$
文献[55]	Greedy-based	$O(n+m)^2$
文献[48]	Based on Utility Matrix	$O(n+m)^3$

5.2 上下界方法

图编辑距离上界是指算法得出的距离值比精确图编辑距离稍大;同理,下界指算法得出的距离值稍小于精确距离。上下界方法利用生成距离的上下界来逼近精确编辑距离。使上下界尽可能小、下界尽可能大,是上下界方法面临的主要问题。

文献[57]将图编辑距离问题视为二次线性规划问题,但只能处理无向无标签图,其生成上下界的时间复杂度分别为 $O(n^7)$ 和 $O(n^3)$ 。

文献[58]基于 BP 代价阵提出时间复杂度为 $O(n^3)$ 的计

算上下界的方法。求解二次指派问题的方法可直接作为求解上界的方法,此时二次指派得出的编辑距离可写为 $d_\psi(g_1, g_2) = C_\psi^{(V)} + C_\psi^{(E)}$;下界值为 $d_\psi'(g_1, g_2) = C_\psi^{(V)} + C_\psi^{(E)}/2$,其中, $C_\psi^{(E)}$ 是由线性和问题求解得出的所有边的操作代价和。此时,精确图编辑距离 $d_{\lambda\min}(g_1, g_2)$ 满足:

$$d_\psi'(g_1, g_2) \leq d_{\lambda\min}(g_1, g_2) \leq d_\psi(g_1, g_2) \tag{7}$$

5.3 基于 Hausdorff 匹配的图编辑距离

5.3.1 Hausdorff 距离

Hausdorff 距离^[59]是描述两组点集之间相似程度的一种度量,也是两个点集之间距离的一种定义形式。令有两组集合 $A = \{a_1, \dots, a_m\}, B = \{b_1, \dots, b_n\}$,则这两个点集之间的 Hausdorff 距离定义为:

$$H(A, B) = \max(h(A, B), h(B, A)) \tag{8}$$

其中:

$$h(A, B) = \max_{a \in A} \min_{b \in B} \|a - b\| \tag{9}$$

$$h(B, A) = \max_{a \in B} \min_{b \in A} \|a - b\|$$

其中, $\|\cdot\|$ 表示点集 A 和点集 B 间的距离范式(如欧氏距离等)。 $H(A, B)$ 称为双向 Hausdorff 距离,是 Hausdorff 距离的基本形式,它度量了两个点集间的最大不匹配程度。 $h(A, B)$ 和 $h(B, A)$ 则分别指集合 A 到集合 B 和集合 B 到集合 A 的单向 Hausdorff 距离。

5.3.2 基于 Hausdorff 编辑距离的匹配

通过对 Hausdorff 距离进行变形,文献[16]就图编辑问题提出一种考虑了所有最近邻距离的距离函数,称为 Hausdorff 编辑距离(HED)。类似于在一个度量空间使用 Hausdorff 距离,Hausdorff 编辑距离是将源图的每个元素与目标图的每个元素分别进行比对。

类似于指派问题,基于 Hausdorff 距离的分配也是由两部分构成,每个图的点都包含一条指派边;但不同之处在于,基于 Hausdorff 距离的指派是有向的,且存在对一个点的多种指派,而指派编辑距离中的指派是无向的,且点的指派是唯一的。其指派图如图 2、图 3 所示。

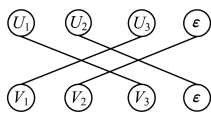


图 2 基于指派编辑距离的分配

Fig. 2 Allocation based on assignment edit distance

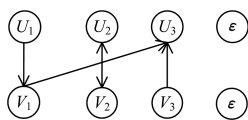


图 3 基于 Hausdorff 距离的分配

Fig. 3 Allocation based on Hausdorff distance

定义图 g_1, g_2 中的点集分别为 $u \in V_1, v \in V_2$,代价函数为 C ,则其间的 Hausdorff 编辑距离如下:

$$HED(g_1, g_2, C) = \sum_{u \in V_1} \min_{v \in V_2 \cup \{\epsilon\}} C_n^*(u, v) + \sum_{v \in V_2} \min_{u \in V_1 \cup \{\epsilon\}} C_n^*(u, v) \tag{10}$$

因 Hausdorff 距离是双映射且有向的,故对于替换操作,其每项代价函数的定义各取一半,而插入和添加操作只会

上式中的一项出现,因此其代价函数可全取,即点的代价函数定义为:

$$C_n^*(u, v) = \begin{cases} \frac{C_n(u, v)}{2} \\ C_n(u, v) \end{cases} \tag{11}$$

不同于分配编辑距离,Hausdorff 编辑距离对边也一一进行 Hausdorff 距离比较,其代价函数的定义与边类似。

概言之,分配编辑距离遍历寻找的是所有相连点的最优分配,而 Hausdorff 距离遍历寻找的则是所有独立点的最优分配。当一个点的最优分配是删除或插入时,Hausdorff 距离比分配编辑距离能更准确地确定操作;当是替换操作时,即替换操作代价值的一半还小于删除或插入操作代价值,Hausdorff 编辑距离则只能估算相应的边代价且小于真实的代价,故 Hausdorff 编辑距离计算值不大于精确图编辑距离。

文献[18]将基于 Hausdorff 编辑距离的匹配方法与贪心分配算法结合使用,提出一种新的时间复杂度为 $O(n^2)$ 的生成上界的方法。实验结果表明,其计算精确度不低于 BP 算法得出的精确度,甚至在某些情况下更优。

5.4 其他近似算法

除了上述近似算法,还存在其他一些时间复杂度为多项式的近似算法。如,概率松弛标签方法^[60-61]从贝叶斯概率公式出发,迭代应用编辑操作以提高最大后验准则;文献[62]提出基于遗传算法的优化方法,这种方法比 A^* 搜索更有效,但只是在搜索空间中逐步寻找局部最优解,最终得出次优解;文献[63-64]通过初始映射的随机构造,再结合局部搜索求解编辑路径距离。

文献[42]提出了对 A^* 算法中启发式函数的两种改进,一是对估值函数 $h(\lambda)$ 限定搜索步数而不是直接搜索到终点,这样可大大减少遍历时间;二是重新将启发式函数定义为:

$$f(\lambda) = \frac{g(\lambda) + h(\lambda)}{t^{|\lambda|}} \tag{12}$$

其中,人为设定的参数 $t > 1, |\lambda|$ 表示从起始点到目前所在点 λ 所形成的路径中编辑操作的数量。实验证明,上述改进方法能有效加快计算速度并得到次优解。

结束语 图编辑距离作为度量图相似性的一种有效且灵活的方法,在模式识别和图像检索领域有着广泛的应用。近年来,基于指派问题的近似图编辑距离方法得到了极大的关注,通过实验也验证了这种方法的适用性及优良性。

在基于图编辑距离的方法中,研究方向和方法主要集中于两点:

1) 代价函数的定义。代价函数定义的优劣直接决定了图相似性比较的最终结果。目前有关定义代价函数的方法具有一定的局限性,或者有一定的约束要求。随着深度学习与机器学习的不断发展,如何构建更加高效的代价模型值得关注;同时,在给定更少的约束条件下,代价函数的聚类问题也值得进一步研究。

2) 编辑距离的求解。模式匹配存在精确和非精确匹配,现有大量匹配算法致力于降低精确匹配的复杂度,提高非精确匹配的精确度。但如何更好地表征源图与目标图之间的相关性,尤其是图中边之间的相关性,值得进一步探究。同时,

如何针对大数据环境下出现的大规模图数据,运用各种算法和策略,在降低算法复杂度的同时稳定地提高匹配精确度也是下一步值得探究的方向之一。

参 考 文 献

- [1] FOGGIA P, PERCANNELLA G, VENTO M. Graph Matching and Learning in Pattern Recognition in the Last 10 Years[J]. *International Journal of Pattern Recognition & Artificial Intelligence*, 2014, 28(1): 178-215.
- [2] YU J, LIU Y B, ZHANG Y, et al. Survey on Large-Scale Graph Pattern Matching[J]. *Journal of Computer Research and Development*, 2015, 52(2): 391-409. (in Chinese)
于静, 刘燕兵, 张宇, 等. 大规模图数据匹配技术综述[J]. *计算机研究与发展*, 2015, 52(2): 391-409.
- [3] BORGWARDT K M, ONG C S, SCHÖNAUER S, et al. Protein function prediction via graph kernels[J]. *Oral Radiology*, 2005, 6(2): 29-35.
- [4] HARCHAOUZI Z, BACH F. Image Classification with Segmentation Graph Kernels[C]// *IEEE Conference on Computer Vision & Pattern Recognition (Cvpr 07)*. IEEE, 2007: 1-8.
- [5] SHOUBRIDGE P, KRAETZL M, WALLIS W, et al. Detection of Abnormal Change in A Time Series of Graphs[J]. *Journal of Interconnection Networks*, 2002, 3(3): 85-101.
- [6] CONTE D, FOGGIA P, SANSONE C, et al. Thirty Years of Graph Matching in Pattern Recognition[J]. *International Journal of Pattern Recognition & Artificial Intelligence*, 2011, 18(3): 265-298.
- [7] BUNKE H, ALLERMANN G. Inexact graph matching for structural pattern recognition[J]. *Pattern Recognition Letters*, 1983, 1(4): 245-253.
- [8] GAO X, XIAO B, TAO D, et al. A survey of graph edit distance [J]. *Pattern Analysis and Applications*, 2010, 13(1): 113-129.
- [9] CORTÉS X, SERRATOSA F. Learning graph-matching edit-costs based on the optimality of the oracle's node correspondences[J]. *Pattern Recognition Letters*, 2015, 56: 22-29.
- [10] SOLÉ-RIBALTA A, SERRATOSA F, SANFELIU A. On the Graph Edit Distance cost: Properties and Applications[J]. *International Journal of Pattern Recognition & Artificial Intelligence*, 2012, 25(5): 53-61.
- [11] BERRETTI S, BIMBO A D, VICARIO E. Efficient Matching and Indexing of Graph Models in Content-Based Retrieval[J]. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 2001, 23(10): 1089-1105.
- [12] GREGORY L, KITTLER J. Using Graph Search Techniques for Contextual Colour Retrieval [C] // *Joint Iapr International Workshop on Structural, Syntactic, and Statistical Pattern Recognition*. Springer-Verlag, 2002: 186-194.
- [13] ABU-AISHEH Z, RAVEAUX R, RAMEL J Y, et al. An Exact Graph Edit Distance Algorithm for Solving Pattern Recognition Problems[C]// *4th International Conference on Pattern Recognition Applications and Methods*. Lisbon, Portugal, 2015: 271-278.
- [14] ABU-AISHEH Z, RAVEAUX R, RAMEL J Y, et al. A Distributed Algorithm for Graph Edit Distance[C]// *The Eighth International Conference on Advances in Databases, Knowledge, and Data Applications*. Lisbon, Portugal, 2016: 66-71.
- [15] RIESEN K, BUNKE H. Approximate graph edit distance computation by means of bipartite graph matching[J]. *Image & Vision Computing*, 2009, 27(7): 950-959.
- [16] FISCHER A, SUEN C Y, FRINKEN V, et al. Approximation of graph edit distance based on Hausdorff matching[J]. *Pattern Recognition*, 2015, 48(2): 331-343.
- [17] SERRATOSA F, CORTÉS X. Graph Edit Distance: Moving from global to local structure to solve the graph-matching problem [J]. *Pattern Recognition Letters*, 2015, 65: 204-210.
- [18] FISCHER A, RIESEN K, BUNKE H. Improved Quadratic Time Approximation of Graph Edit Distance by Combining Hausdorff Matching and Greedy Assignment[J]. *Pattern Recognition Letters*, 2017, 87: 55-62.
- [19] AMBAUEN R, FISCHER S, BUNKE H. Graph Edit Distance with Node Splitting and Merging, and Its Application to Diatom Identification[C]// *Iapr International Conference on Graph Based Representations in Pattern Recognition*. Springer-Verlag, 2003: 95-106.
- [20] LEVENSHTAIN V I. Binary codes capable of correcting deletions, insertions and reversals [J]. *Problems of Information Transmission*, 1966, 10(1): 707-710.
- [21] PROBLEM T T E. The tree-to-tree editing problem[J]. *Information Processing Letters*, 1977, 6(6): 184-186.
- [22] ESHERA M A, FU K S. A graph distance measure for image analysis[J]. *IEEE Transactions on Systems Man & Cybernetics*, 1984, 14(3): 398-408.
- [23] RIESEN K. *Structural Pattern Recognition with Graph Edit Distance: Approximation Algorithms and Applications [M]*. Springer Publishing Company, Incorporated, 2016: 14-135.
- [24] WEI J. Markov edit distance[J]. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 2004, 26(3): 311-321.
- [25] WAGNER R A, FISCHER M J. The String-to-String Correction Problem[J]. *Journal of the ACM*, 1974, 21(1): 168-173.
- [26] BUNKE H. Error Correcting Graph Matching: On the Influence of the Underlying Cost Function[J]. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 1999, 21(9): 917-922.
- [27] BUNKE H. On a relation between graph edit distance and maximum common subgraph[J]. *Pattern Recognition Letters*, 1997, 18(9): 689-694.
- [28] MALTONI D, MAIO D, JAIN A K, et al. *Handbook of Fingerprint Recognition [M]*. New York: Springer-Verlag, 2009: 12-50.
- [29] CAETANO T S, MCAULEY J J, CHENG L, et al. Learning graph matching [J]. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 2008, 31(6): 1048-1058.
- [30] LLADÓS J, MARTÍ E, VILLANUEVA J J. Symbol Recognition by Error-Tolerant Subgraph Matching between Region Adjacency Graphs[J]. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 2001, 23(10): 1137-1143.
- [31] NEUHAUS M, BUNKE H. Automatic learning of cost functions for graph edit distance[J]. *Information Sciences*, 2007, 177(1): 239-247.
- [32] WONG A K C, YOU M. Entropy and Distance of Random Graphs with Application to Structural Pattern Recognition[J]. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 1985, 7(5): 599-609.

- [33] SERRATOSA F, ALQUEZAR R, SANFELIU A. Function-described graphs for modelling objects represented by sets of attributed graphs[J]. *Pattern Recognition*, 2003, 36(3): 781-798.
- [34] SANFELIU A, SERRATOSA F, ALQUÉZAR R. Second-Order Random Graphs for modelling sets of Attributed Graphs and their application to object learning and recognition[J]. *International Journal of Pattern Recognition & Artificial Intelligence*, 2011, 18(3): 375-396.
- [35] NEUHAUS M, BUNKE H. A probabilistic approach to learning costs for graph edit distance[C]// *Proceedings of IEEE International Conference on Pattern Recognition*. 2004: 389-393.
- [36] NEUHAUS M, BUNKE H. Self-organizing maps for learning the edit costs in graph matching[J]. *IEEE Transactions on Systems Man & Cybernetics Part B Cybernetics A Publication of the IEEE Systems Man & Cybernetics Society*, 2005, 35(3): 503-514.
- [37] SERRATOSA F, SOLÉRIBALTA A, CORTÉS X. Automatic Learning of Edit Costs Based on Interactive and Adaptive Graph Recognition [C] // *International Conference on Graph-Based Representations in Pattern Recognition*. Springer-Verlag, 2011: 73-78.
- [38] DUMAY A C M, GEEST R J V D, GERBRANDS J J, et al. Consistent inexact graph matching applied to labelling coronary segments in arteriograms[C]// *Iapr International Conference on Pattern Recognition*. 1992: 439-442.
- [39] RIESEN K, FANKHAUSER S, BUNKE H. Speeding Up Graph Edit Distance Computation with a Bipartite Heuristic [C] // *Proc. 5th Int. Workshop on Mining and Learning with Graphs*. Firenze, Italy, 2007: 21-24.
- [40] FISCHER A, PLAMONDON R, SAVARIA Y, et al. A Hausdorff Heuristic for Efficient Computation of Graph Edit Distance [C]// *Int. Workshop on Structural and Syntactic Pattern Recognition*. 2014: 83-92.
- [41] HART P E, NILSSON N J, RAPHAEL B. A Formal Basis for the Heuristic Determination of Minimum Cost Paths[J]. *IEEE Transactions on Systems Science & Cybernetics*, 1968, 4(2): 100-107.
- [42] NEUHAUS M, RIESEN K, BUNKE H. Fast suboptimal algorithms for the computation of graph edit distance[C]// *Joint Iapr International Conference on Structural, Syntactic, and Statistical Pattern Recognition*. Springer-Verlag, 2006: 163-172.
- [43] KOOPMANST C, BECKMANN M. Assignment Problems and the Location of Economic Activities[J]. *Cowles Foundation Discussion Papers*, 1955, 25(1): 53-76.
- [44] BOUGLEUX S, BRUN L, CARLETTI V, et al. Graph Edit Distance as a Quadratic Assignment Problem[J]. *Pattern Recognition Letters*, 2016, 3(4): 1-9.
- [45] SERRATOSA F. Fast computation of Bipartite graph matching [J]. *Pattern Recognition Letters*, 2014, 45(8): 244-250.
- [46] BOURGEOIS F, LASSALLE J C. An extension of the Munkres algorithm for the assignment problem to rectangular matrices [J]. *Communications of the ACM*, 1971, 14(12): 802-804.
- [47] SERRATOSA F. Speeding up Fast Bipartite Graph Matching Through a New Cost Matrix[J]. *International Journal of Pattern Recognition & Artificial Intelligence*, 2014, 29(2): 1550010.
- [48] RIESEN K, FISCHER A, BUNKE H. Approximation of Graph Edit Distance by Means of a Utility Matrix[C]// *Artificial Neural Networks in Pattern Recognition*. Springer International Publishing, 2016: 185-194.
- [49] BURKARD R E, DELL'AMICRO M, MARTELLO S. Assignment Problems[M]. *Society for Industrial and Applied Mathematics*, 2009: 9-100.
- [50] FANKHAUSER S, RIESEN K, BUNKE H. Speeding up graph edit distance computation through fast bipartite matching[J]. *Lecture Notes in Computer Science*, 2011, 6658(1): 102-111.
- [51] KUHN H W. The Hungarian method for the assignment problem[J]. *Naval Research Logistics*, 2005, 52(1): 83-97.
- [52] MUNKRES J. Algorithms for the assignment and transportation problems[J]. *Journal of the Society for Industrial & Applied Mathematics*, 1957, 5(1): 32-38.
- [53] JONKER R, VOLGENANT A. A shortest augmenting path algorithm for dense and sparse linear assignment problems[J]. *Computing*, 1987, 38(4): 325-340.
- [54] JONES W, CHAUDHARY A, KING A. Optimizing the Volgenant-Jonker algorithm for approximating graph edit distance [J]. *Pattern Recognition Letters*, 2016, 13(27): 1-8.
- [55] RIESEN K, FERRER M, BUNKE H. Approximate Graph Edit Distance in Quadratic Time [J]. *IEEE/ACM Transactions on Computational Biology & Bioinformatics*, 2014, 13(9): 1-12.
- [56] RIESEN K, BUNKE H. Improving bipartite graph edit distance approximation using various search strategies [J]. *Pattern Recognition*, 2014, 48(4): 1349-1363.
- [57] JUSTICE D, HERO A. A binary linear programming formulation of the graph edit distance[J]. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 2006, 28(8): 1200-1214.
- [58] RIESEN K, FISCHER A, BUNKE H. Computing Upper and Lower Bounds of Graph Edit Distance in Cubic Time[C]// *Int. Workshop on Artificial Neural Networks in Pattern Recognition*. 2014: 129-140.
- [59] HUTTENLOCHER D P, KLANDERMAN G, RUCKLIDGE W J. Comparing images using the Hausdorff distance [J]. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 1993, 15(9): 850-863.
- [60] WILSON R C, HANCOCK E R. Structural Matching by Discrete Relaxation[J]. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 1997, 19(6): 634-648.
- [61] MYERS R, WILSON R C, HANCOCK E R. Bayesian Graph Edit Distance[J]. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 2000, 22(6): 628-635.
- [62] CROSS A D J, WILSON R C, HANCOCK E R. Inexact graph matching using genetic search[J]. *Pattern Recognition*, 1997, 30(6): 953-970.
- [63] BOERES M C, RIBEIRO C C, BLOCH I. A Randomized Heuristic for Scene Recognition by Graph Matching[C]// *Experimental and Efficient Algorithms*, Third International Workshop, WEA 2004. Angra dos Reis, Brazil, 2004: 100-113.
- [64] SORLIN S, SLONON C. Reactive Tabu Search for Measuring Graph Similarity[C]// *Iapr International Conference on Graph-Based Representations in Pattern Recognition*. Springer-Verlag, 2005: 172-182.