

引用格式:徐周波,黄文文,刘华东,等. 基于卡方统计的近似子图匹配[J]. 微电子学与计算机,2020,37(11):17-23. [XU Z B, HUANG W W, LIU H D, et al. Approximate subgraph matching based on Chi-squares statistics[J]. Microelectronics & Computer, 2020, 37(11): 17-23.]

基于卡方统计的近似子图匹配

徐周波¹, 黄文文¹, 刘华东^{1,2}, 杨 健¹

(1 桂林电子科技大学 广西可信软件重点实验室, 广西 桂林 541004;

2 桂林电子科技大学 机电工程学院, 广西 桂林 541004)

摘 要: 图查询的应用越来越广泛, 其中近似子图匹配是核心技术之一. 但是大规模图数据中噪音的存在对近似子图匹配精确度影响较大, 为进一步提高近似子图匹配算法的鲁棒性和实时性, 提出一种基于卡方统计的近似子图匹配改进算法. 在算法预处理阶段, 利用统一邻居随机游走距离和高斯影响函数将目标图划分, 使得划分后的子图在拓扑结构和标签属性之间达到最佳平衡. 在算法匹配阶段, 使用卡方统计量捕获的统计显著性来表征近似子图匹配结构相似度, 再结合权重系数 α 调整结构相似度和标签相似度所占比重, 其中统计显著性模型能够充分考虑背景结构和顶点邻域中的标签分布, 有效处理部分标签和结构失配, 从而得到最佳匹配子图. 真实数据集集中的实验结果表明, 该算法效果较好, 运算效率较高, 可以应用于 Top-k 近似子图匹配.

关键词: 图结构; 近似子图匹配; 统计显著性; 卡方统计; 图划分

中图分类号: TP311

文献标识码: A

文章编号: 1000-7180(2020)11-0017-07

Approximate subgraph matching based on Chi-squares statistics

XU Zhou-bo¹, HUANG Wen-wen¹, LIU Hua-dong^{1,2}, YANG Jian¹

(1 Guangxi Key Laboratory of Trusted Software, Guilin University of Electronic technology, Guilin 541004, China;

2 School of Mechanical & Electrical Engineering, Guilin University of Electronic technology, Guilin 541004, China)

Abstract: Graph query is more and more widely used, and approximate subgraph matching is one of the core technologies. However, the existence of noise in large-scale graph data has a great impact on the accuracy of approximate subgraph matching. In order to further improve the robustness and real-time performance of approximate subgraph matching algorithm, an improved algorithm of approximate subgraph matching based on Chi-square statistics is proposed. In the preprocessing stage, the target graph is partitioned by using the Unified Neighborhood Random Walk Distance and Gaussian Influence Function, so that the partitioned subgraph can achieve the best balance between the topological structure and label attributes. In the algorithm matching stage, the statistical significance captured by Chi-square statistics is used to characterize the subgraph similarity of approximate subgraph matching, and then the proportion of structural similarity and label similarity is adjusted by combining the weight coefficient α . The statistical significance model can takes into account the background structure and label distribution in the neighborhood of vertices, and effectively handles partial label and structural mismatches, so as to get the best matching subgraph. The experimental results in real dataset show that the algorithm is effective and efficient, and can be applied to Top-k approximate subgraph matching.

Key words: graph structure; approximate subgraph matching; statistical significance; chi-square statistics; graph partition

1 引言

随着社交网络、蛋白质交互网络^[1]、道路网络等图数据的激增,使得在大图中近似子图匹配成为研究的热点和难点. 目前存在多种相似度量标准的近似子图匹配算法^[2-7].

针对 VELSET 算法^[7]没有考虑避免构建不必要的节点对和根据不同数据集调整结构相似度和标签相似度所占比重的问题,本文首先通过将大型数据图分区,然后增加权重系数 α 调整结构相似度和标签相似度所占比重,有望提高算法的运行效率和匹配结果精确度. 具体改进如下: (1) 利用统一邻居随机游走距离将原始大图进行分区,使得分区后的子图在拓扑结构和标签属性之间可以达到最佳平衡状态; (2) 通过 Jaccard 相似性计算,决定查询图的匹配子图最大可能属于哪个分区,再构建相应节点对; (3) 在计算节点对匹配相似性程度的时候,通过增加权重系数 α ,来调整标签属性相似度和结构相似度所占比重.

基于以上 3 点改进,本文提出近似子图匹配算法 IMVELSET (Improved Vertex Label Similarity on Edge Triplets),该算法使得大图优化分解为小图,不仅使得复杂查询简单化,还提高了运算效率和匹配结果精确度.

2 知识背景

统计显著性已被成功运用于图数据挖掘的多个领域,如挖掘连通公共子图^[8]、序列挖掘^[9]等. 从统计学角度来看,统计显著性与 p -value^[10] 紧密相关,一般 p -value < 0.05 表明差异性显著, p -value < 0.01 表明差异性极显著. 比如,在一次硬币投掷实验中,连续投掷 10 次,其中 8 次正面 (9 次, 10 次) 的概率是多少? 因为 10 次投掷中有 8 次正面事件发生的概率很低,即 p -value 值越小,则表明该事件的统计显著性越大. 同理,当一个带标签的查询图去大型带标签数据图中寻找最佳近似匹配子图时,如果大图中的子图与查询图匹配的统计显著性很大,则也可以表明该子图是最佳候选匹配子图. 但是在实际过程中,由于 p -value 的计算是指数级,因此本文采用 Pearson 卡方统计^[10]去替代 p -value. 卡方统计是计算观察值与期望值之间的差异,公式如下:

$$\chi^2 = \sum_{\forall i} \chi_i^2 = \sum_{\forall i} [(O_i - E_i)^2 / E_i] \quad (1)$$

式中, O_i 和 E_i 分别表示在一次实验中,事件 i 出现的观察次数 (值) 和期望次数 (值). 如果 p_i 表示事件 i 出现的概率,可观察的数量为 len ,则 $E_i = p_i \times \text{len}$. 综上所述, χ^2 值越大,则 p -value 越小,表明统计显著性越大. 因此,以上就形成了本文提出的近似子图匹配算法的基础.

问题描述: 给定一个带标签数据图 G 和一个带标签查询图 Q , 近似子图匹配问题为找到 G 中满足 (1) 的子图 G_m :

$$G_m = \arg\max_{G'} ss(Q, G') \quad (2)$$

式中, $G' \subseteq G$, 且 $G' \approx Q$, $ss(Q, G')$ 为计算 Q 与 G' 相似的统计显著性程度, 假设统计显著性程度值上界为 τ , 若 $ss(Q, G') = \tau$, 则表示 Q 与 G' 同构.

为了度量图之间相似的统计显著性程度, 定义如下:

$$ss(Q, G') = \sum_{v \in G', q \in Q} (\alpha \times JS(l_v^G, l_q^Q) + (1 - \alpha) \times \chi^2_{<v, q>}) \quad (3)$$

式中, $JS(l_v^G, l_q^Q) > \epsilon$ 表示节点标签之间的 Jaccard 相似性, ϵ 为标签相似度的阈值, $\chi^2_{<v, q>}$ 表示节点之间结构相似度的卡方统计值, $\alpha \in [0, 1]$, 为权重调节参数.

3 IMVELSET 算法

本文将近似子图匹配分为预处理、统计显著性计算和产生近似匹配三个阶段. 其中预处理阶段包括大型带标签数据图最佳平衡分区和创建索引列表.

3.1 预处理

本小节内容主要包括图分区算法和创建索引列表, 其主要目的是为了统计显著性的计算做准备工作. 与此同时, 为避免 VELSET 算法产生很多不必要的节点对, 图分区算法可以很大程度减少此步骤的时空消耗, 而索引列表的创建便于在构建节点对时快速查询.

3.1.1 图分区算法

给定一个带标签数据图 $G = (V, E, \sum)$, 其中 V 表示节点集, E 表示边集, $\sum = \{a_1, a_2, \dots, a_m\}$ 表示 m 个属性集. 数据图分区就是将数据图 G 划分为 k 个不相交的子图 $G_i = (V_i, E_i, \Sigma)$, 其中 $V = \bigcup_{i=1}^k V_i$ 且对任意 $i \neq j$, $V_i \cap V_j = \emptyset$. 注意到实际上任何满足此要求的图分区算法均适用. 希望划分得到的子图“高内聚、低耦合”原则, 即同一分区内部的边尽可能多, 而连接不同划分的边尽可能少. 但

是对于带标签数据图,如何做到在图分区后,在拓扑结构和标签属性之间达到最佳平衡是至关重要.文献[11]提出了统一邻居随机游走距离将数据图划分,使得划分后的子图在拓扑结构和标签属性之间达到最佳平衡的思想,启发了本文在近似子图匹配过程中,先将大图优化分解为小图,在小图上采用近似子图匹配算法,从而得到最佳匹配子图的思想.

在文献[11]算法中,为衡量图分区质量,提出了统一随机游走距离的概念,在属性增强图中,任意两个节点 v_i 和 v_j 之间的紧密性程度定义如下:

$$d(v_i, v_j) = \sum_{\substack{\tau: v_i, v_j \\ \text{length}(\tau) \leq l}} p(\tau) c (1-c)^{\text{length}(\tau)} \quad (4)$$

式中, τ 表示从 v_i 到 v_j 的距离长度为 $\text{length}(\tau)$, 转移概率为 $p(\tau)$, $c(1-c)^{\text{length}(\tau)}$ 表示随机游走 τ 步之后没有重新跳回起点的概率. 距离值越大表明该节点对之间越紧密. 首先分别累加计算每一个节点与剩余其他节点之间的随机游走距离值, 然后从中选取前 Top- k 个最大随机游走距离值的节点作为 k 个分区的质心点, 最后根据非质心点与质心点之间的随机游走距离值决定非质心点所属哪个分区. 算法采用迭代方式进行处理, 迭代 t 次直至所有分区内平均随机游走距离值最大且目标函数收敛结束. 如算法 1 所示.

算法 1. 图分区算法

输入: 一个带标签属性图, 随机游走距离长度 l , 重新开始概率 c 和一个分区数 k

输出: k 个不相交的子图 $V_1, \dots, V_K$

1. 初始化属性增强图中边的权重 $w_1 = w_2 = \dots = w_m = 1.0$, $w_0 = 1.0$, 计算统一随机游走距离矩阵 R_1^A .
2. 根据高斯影响函数和稠密度函数选取 k 个质心点.
3. 重复步骤 4, 直至目标函数收敛:
4. 根据 $c = \arg \max_j d(v_i, v_j)$ 计算剩余节点与质心点之间的平均最大距离值, 从而决定剩余节点所属分区, 而后更新属性边的权重 $w_1, w_2, \dots, w_m$, 重新计算统一随机游走距离矩阵 R_1^A .

5. 输出 k 个分区 $V_1, \dots, V_K$.

3.1.2 创建索引列表

本小节算法会构建不同的索引列表来概括节点的标签信息和邻居节点信息. 如图 1 所示, 分别对数据图 G 和查询图 Q 创建倒排索引列表 IL^G 、 IL^Q 和标签邻居列表 LNL^G 、 LNL^Q . 比如数据图 G 中含 A 标签, 可表示为 $IL_{v_1}^G$ (因为数据图含 A 标签只有节点 v_1), 简化为 $IL_{v_1}^G$. 对于任意 $v \in G$, 则节点 v 标签邻居列表可表示为 LNL_v^G . 最后根据分区结果为不同的分区子图分别创建索引列表信息.

3.2 统计显著性计算

3.2.1 节点对的构建

同计算标签相似性一致, 本文采用 Jaccard 相似性确定查询图最大可能与数据图的哪个分区子图相似, 假如 $JS(Q, G_i)$ 相似度值最大, 则 Q 首先与分区 G_i 进行后续匹配计算. 然后对于查询图 Q 中的每一个标签, $IMVELSET$ 都会在该分区子图中寻找与之对应的相似性标签. 定义一个阈值 ϵ , 用 Jaccard 相似性来衡量标签的相似性, 比如计算两个标签字符串 s 和 t 的 Jaccard 相似性, $JS(s, t) = |s \cap t| / |s \cup t|$. 如果查询图 Q 中的节点 i 标签 l_i^Q 与数据图 G 中的节点 j 标签 l_j^G Jaccard 相似度大于 ϵ , 即表示为 $JS(l_i^Q, l_j^G) > \epsilon$, 说明 l_i^Q 相似性于 l_j^G , 简写为 $l_i^Q \sim l_j^G$, 不相似简写为 $l_i^Q \not\sim l_j^G$. 如图 1 数据图 G 所示, 假设 $\epsilon = 0.5$, 那么 EA 标签相似性于 A 标签. 此时通过计算会得到查询图 Q 与子图 G_i 形成节点对 $\langle v, q \rangle$, 定义如下:

$$VP = \{ \langle v, q \rangle \mid v \in G_i \wedge q \in Q \wedge l_v^G \sim l_q^Q \} \quad (5)$$

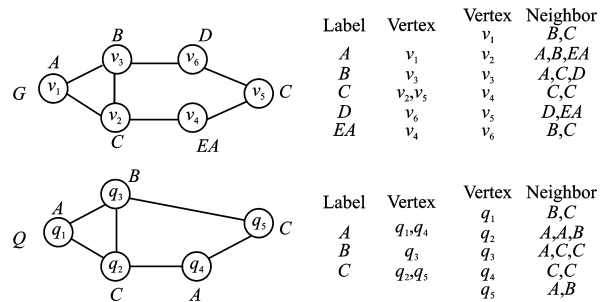


图 1 数据图和查询图的索引构建

3.2.2 统计显著性计算

将介绍如何利用卡方统计, $\chi^2_{\langle v, q \rangle}$ 来计算节点对 $\langle v, p \rangle \in VP$ 的统计显著性.

(1) 标签元组的产生

首先对任意一个节点 $v \in G_i$ 构建一个标签三元组, 其中 v 的邻居为 x 和 y , 则对应的标签三元组记为 $\langle l_x^G, l_v^G, l_y^G \rangle$, 邻居的顺序不影响后续卡方值的计算, 即 $\langle l_x^G, l_v^G, l_y^G \rangle \equiv \langle l_y^G, l_v^G, l_x^G \rangle$. 例如, 图 1 中的节点 v_3 的标签为 B , 其中 v_3 的邻居 v_1, v_2, v_6 的标签分别为 A, C, D . 因此, 以 v_3 为中心的标签三元组有 $\langle A, B, C \rangle, \langle C, B, D \rangle, \langle D, B, A \rangle$. 类似的有, 以 q_5 为中心标签三元组为 $\langle B, C, A \rangle$.

(2) 标签重叠度的计算

两个标签元组的相似性可以通过标签重叠度来衡量. 令节点对 $\langle v, p \rangle \in VP$ (其中 $l_v^G \sim l_p^G$), 分别

以 v 和 q 构建的标签三元组为 $\langle l_x^Q, l_q^Q, l_y^Q \rangle, \langle l_a^G, l_v^G, l_b^G \rangle$. 基于标签相似性, 标签重叠度可以分为 3 类(分别用字符 s_2, s_1, s_0 表示) 如下:

s_2 : 节点 v 和 q 的邻居标签均相似:

$$s_2 = (l_x^Q \sim l_a^G \wedge l_y^Q \sim l_b^G) \quad (6)$$

s_1 : 只有一个邻居标签相似:

$$s_1 = (l_x^Q \sim l_a^G \wedge l_y^Q \neq l_b^G) \vee (l_x^Q \neq l_a^G \wedge l_y^Q \sim l_b^G) \quad (7)$$

s_0 : 邻居标签都不相似:

$$s_0 = (l_x^Q \neq l_a^G \wedge l_y^Q \neq l_b^G) \quad (8)$$

如果标签重叠匹配程度越高(即 s_2 的数量越多), 则结构越相似. 如图 1 中的节点对 $\langle v_3, q_3 \rangle$, 其中一个以 q_3 为中心的标签元组 $\langle A, B, C \rangle$ 精确匹配 $\langle v_1, v_3, v_2 \rangle$ 构建的标签元组 $\langle A, B, C \rangle$, 因此用字符 s_2 表示标签重叠度. 同理, $\langle C, B, C \rangle$ 匹配 $\langle C, B, D \rangle$, $\langle C, B, A \rangle$ 匹配 $\langle D, B, A \rangle$, 分别用 s_1 表示. 综上, v_3 匹配 q_3 标签重叠度(观察 \ 实际标签重叠度) 的字符序列表示为 s_2, s_1, s_1 .

(3) 期望标签重叠度

对于查询图 Q 节点 $q (q \in Q)$ 与数据图 G 的节点 $v (v \in G)$ 实际观察匹配相似度可以用 O_u 表示(如 $O_{\langle v_3, q_3 \rangle} = s_2, s_1, s_1$). 则统计显著性可以通过卡方统计中的 O_u (实际 \ 观察值)、 E_u (期望值) 来计算. 然而期望匹配字符出现的次数是不确定的. 假定有 L 组可能的标签分组, 组内的标签都是相似的, 也可认为组内的标签都是可匹配的.

假如节点 v 的邻居标签与查询图节点 q 的邻居标签 l_x^G 不相似, 该事件出现的概率为 $(1 - 1/L)$. 又因 $d = \deg(v)$ 为 v 的邻居个数(即节点 v 的度), 则所有 v 的邻居标签均不相似于 l_x^G 的概率为 $(1 - 1/L)^d$. 考虑到查询标签 l_x^G 和 l_y^G 相互独立, 因此期望元组标签匹配字符为 s_0 定义如下:

$$p(s_0) = ((1 - 1/L)^d)^2 \quad (9)$$

同理, 期望元组标签匹配字符为 s_1 定义如下:

$$p(s_1) = 2 \cdot ((1 - 1/L)^d) \cdot (1 - (1 - 1/L)^d) \quad (10)$$

期望元组标签匹配字符为 s_2 定义如下:

$$p(s_2) = (1 - (1 - 1/L)^d)^2 \quad (11)$$

这里需要声明的是, 上面三个匹配字符事件的概率空间为:

$$p(s_2) + p(s_1) + p(s_0) = 1 \quad (12)$$

(4) 卡方值的计算

IMVELSET 计算节点 v 与 q 匹配的期望标签重叠度的字符序列出现的概率用公式(9) ~ (11) 可计

算得出结果. 其中, 每个字符出现的观察值用实际匹配来计算. 因此, 由公式(3) 可计算节点对 $\langle v, q \rangle$ 的卡方值 $\chi^2_{\langle v, q \rangle}$.

然而, 本文中在计算节点匹配相似度的过程中, 不仅考虑结构信息, 而且也结合了标签属性信息. 因此, 对任意节点对 $\langle v, p \rangle \in VP$, 最终节点对匹配相似度由元组 $ss(v, q) = \langle v, q, (\alpha \times JS(l_v^G, l_q^G) + (1 - \alpha) \times \chi^2_{\langle v, q \rangle}) \rangle$ 计算得出.

例如, 如图 1 中的节点对 $\langle v_3, q_3 \rangle$, 对节点 v_3 而言, 实际匹配字符序列为 s_2, s_1, s_1 , 因此实际观察值 O_u 的次数 $O[l_{v_3}^G, l_{q_3}^Q] = [0, 2, 1]$. 假定 $|L| = 4, d = \deg(q_3) = 3, \alpha = 0.5$, 则根据公式(9) ~ (11), 节点 v_3 与节点 q_3 的期望匹配字符序列 $E(l_{v_3}^G, l_{q_3}^Q) = 3 \times [0.178, 0.488, 0.334]$. 因此, $ss(v_3, q_3) = 0.865$.

3.3 近似匹配结果的产生

将所有的节点对相似度值 ss 插入大顶堆中, 然后从大顶堆中选择 $ss(v, q)$ 值最大的作为种子节点对. 再重新计算所有节点节点对 v 和 q 的邻居节点对的 ss 值放置另一个大顶堆中, 从该邻居大顶堆中同样选择 ss 值最大的作为种子节点对的邻居节点匹配对. 重复以上步骤, 直至查询图匹配完.

4 实验结果及分析

4.1 实验准备

实验环境: Core-i5-6500@3.2 GHz, 16 GB 内存, 操作系统是 Ubuntu12.04 LTS 64 位, 编译环境为 gcc/c++. 实验使用了 YAGO^[12]、DBLP、Protein 这三个数据集. YAGO 是从维基百科中提取的语义知识库数据集(<http://www.yago-knowledge.org>): 节点表示实体, 边表示实体之间的关系、标签表示实体的属性. DBLP 是参考文献数据集(<http://www.informatik.uni-trier.de/ley/db/>), 其中主要包含数据库研究(DB)、数据挖掘(DM)、信息检索(IR)和人工智能(AI)四个领域, 节点表示作者, 边表示作者之间的关系, 标签表示作者的研究领域. Protein 是蛋白质交互网络数据集(<https://www.uniprot.org/>), 节点表示蛋白质, 边表示蛋白质之间的关系. 处理后的数据集的详细统计信息见表 1.

表 1 数据集信息统计

	YAGO	DBLP	Protein
节点数	12 811 149	797 787	19 428
边数	18 282 215	1 301 298	36 631
标签数	9 505 202	3 752	19 428

查询图是由目标图的子集加上一部分噪音构成的,噪音主要有结构噪音和标签噪音,结构噪音是从目标图中随机选取一部分节点(不超过 30%),然后进行统一的插入或删除操作,标签噪音也是从目标图中随机选取一部分节点(不超过 30%),然后进行统一插入任意字符串操作。

实验过程中对比的算法主要有 NeMa^[4]、SIM-T^[5],以及本文所改进前的 VELSET^[7]算法,之所以选 NeMa 和 SIM-T,主要是因为是目前就运算效率和精确度而言,它们是最好的。

表 2 YAGO 数据集的平均精确度、召回率和 F_1 得分/%

查询情形	NeMa			SIM-T			VELSET			IMVELSET		
	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1
无噪音	98.96	86.04	92.25	81.65	81.65	81.65	96.15	96.15	96.15	97.03	97.03	97.03
只有结构噪音	98.32	86.14	92.23	86.59	86.59	86.59	96.74	96.74	96.74	98.11	98.11	98.11
只有标签噪音	97.44	85.15	91.19	70.26	70.26	70.26	95.26	95.26	95.26	97.56	97.56	97.56
两种噪音都有	99.01	85.99	92.34	71.22	71.22	71.22	94.68	94.68	94.68	96.98	96.98	96.98
平均	98.43	85.83	92.00	77.43	77.43	77.43	95.70	95.70	95.70	97.42	97.42	97.42

表 3 DBLP 数据集的平均精确度、召回率和 F_1 得分/%

查询情形	NeMa			SIM-T			VELSET			IMVELSET		
	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1
无噪音	96.25	85.16	90.36	83.24	83.24	83.24	95.63	95.63	95.63	98.92	98.92	98.92
只有结构噪音	95.69	86.26	90.73	87.26	87.26	87.26	95.74	95.74	95.74	98.15	98.15	98.15
只有标签噪音	95.03	85.04	89.75	72.69	72.69	72.69	95.16	95.16	95.16	97.56	97.56	97.56
两种噪音都有	96.86	86.03	91.12	73.51	73.51	73.51	95.68	95.68	95.68	98.98	98.98	98.98
平均	95.96	85.62	90.49	79.18	79.18	19.18	95.55	95.55	95.55	98.4	98.4	98.4

表 4 Protein 数据集的平均精确度、召回率和 F_1 得分/%

查询情形	NeMa			SIM-T			VELSET			IMVELSET		
	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1
无噪音	90.16	90.01	90.08	77.43	77.43	77.43	96.01	96.01	96.01	97.02	97.02	97.02
只有结构噪音	88.43	88.36	88.39	75.12	75.12	75.12	97.23	97.23	97.23	97.95	97.95	97.95
只有标签噪音	90.13	55.19	68.45	65.43	65.43	65.43	96.95	96.95	96.95	96.99	96.99	96.99
两种噪音都有	90.25	58.63	71.08	69.21	69.21	69.21	97.21	97.21	97.21	97.86	97.86	97.86
平均	89.74	73.05	45.43	71.79	71.79	71.79	96.85	96.85	96.85	97.46	97.46	97.46

从表 2、表 3 和表 4 可以看出,当查询情形是无噪音查询、只有结构噪音时、只有标签噪音和两种噪音都有这四种情形时,IMVELSET 和 VELSET 均呈现出较高的精确度、召回率和 F_1 得分. 这表明在各种噪音的情形下节点匹配都具有较大的卡方统计值得以区分,但是 IMVELSET 相较于 VELSET 的在精确度标准的衡量下优势并不是很突出. 再结合图 1、图 2、图 3 来看,当节点查询数量分别是 3、6、9、12、15、18 时,此时就平均查询时间(10 组)而言,IMVELSET 算法遥遥领先. 结合二者来看,充

4.2 精确度和运算效率指标的评测实验及分析

为了多角度验证 IMVELSET 算法的好坏,首先将查询图分为四种情形:不包含噪音、只包含结构噪音、只包含标签噪音和两种噪音都包含;其次查询图的节点数量共六种,分别选取 3,6,9,12,15,18;最后使用 NeMa、SIM-T、VELSET 和 IMVELSET 算法在三种数据集上进行 240(4×6×10)组查询操作. 实验结果的度量主要用精确度(P)、召回率(R)和结合精确度-召回率的调和平均值(F_1),以及运算所需要的时间(s).

分表明改进后的 IMVELSET 算法优势在于在算法前期节点对构建过程中,由于将大图分区的效果,避免了很多不必要的节点对构建和搜索节点对时间,从而为后续根据标签重叠度计算卡方值时节省了大量的时间。

然而就 Protein 数据集 IMVELSET 依然保持较好性能,而 NeMa 的召回率和 F_1 得分急剧下降,这说明 IMVELSET 算法的适用范围更广,匹配结果更准确. 原因主要有两点:①IMVELSET 算法默认对实体本身就是标签这种情形是忽视的;②IM-

VELSET 在计算元祖重叠度时,对于少量的噪声,不能很大程度影响卡方值的计算.与此同时,从图 2~图 4 可以看出,NeMa 运算效率相较于 VELSET 占优势,对于 IMVELSET 来说,IMVELSET 要优于 NeMa,进一步说明平衡结构和标签分区的意义很大. SIM-T 不论就精确度、还是运算效率,效果都不尽人意,其主要归咎于 tabu 算法本身的复杂性.

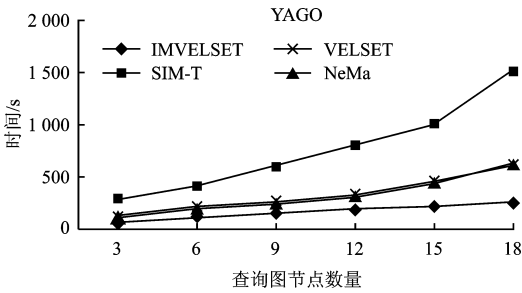


图 2 YAGO 数据集的平均运行效率

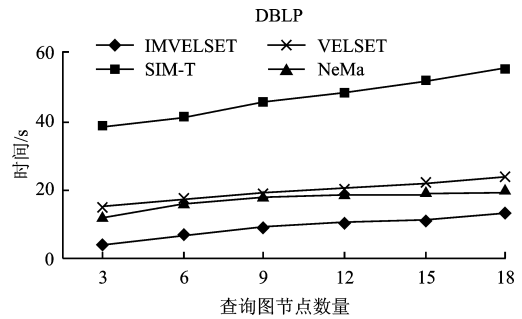


图 3 DBLP 数据集的平均运行效率

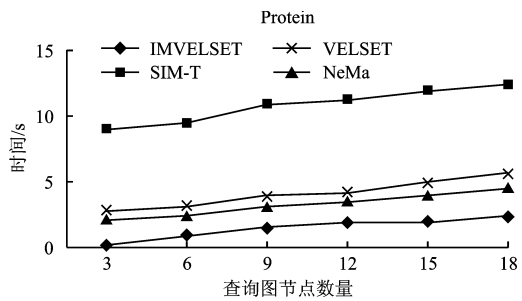


图 4 Protein 数据集的平均运行效率

综上所述、从总体平均情况而言, SIM-T 效果一直较差, NeMa 只有在无噪音和只有结构噪音的情形下才能和 VELSET、IMVELSET 算法具有可比性. 然而 IMVELSET 就精确度始终较 VELSET 保持较低的优势, 但是运算效率, IMVELSET 具有很大优势, 这说明在有效的处理噪音复杂的情形时, 选择 IMVELSET 更好.

4.3 参数的影响

这小节我们将会探索随着 α 参数的变化, 对

IMVELSET 在 YAGO、DBLP 和 Protein 这三个数据集 F_1 得分的表现. 从图 5 可以看出, 对于 YAGO 和 DBLP 数据集而言, 当 α 逐渐从 0.1 升至 0.5 附近的时候, F_1 得分也渐渐变高直到最高, 当 α 慢慢趋于 1 时, F_1 得分也会越来越小, 这表明算法在设计的时候是充分结合结构信息和标签属性信息, 只有当二者平衡达到最佳时(即结构相似度和标签相似度各占一半时), F_1 得分才会达到最高. 然而对于 Protein 数据集来说, α 位于 0.3 附近时, F_1 得分最大. 之所以呈现这样的情况, 其主要是因为 PPI (Protein-ProteinInteraction) 在匹配过程中, 网络拓扑结构信息更加重要, 结果也更加说明符合 PPI 全局比对的实际情况.

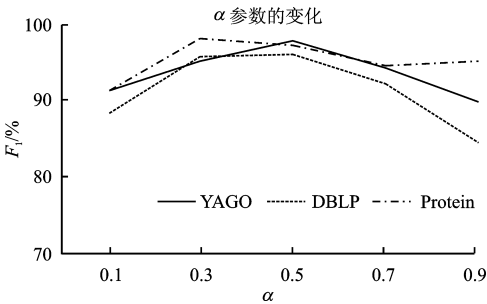


图 5 参数 α 从 0.1 变化到 0.9 对 F_1 值的影响

5 结束语

本文提出了采用卡方统计捕获的统计显著性来表征近似子图匹配相似度, 首先在算法预处理阶段, 利用统一邻居随机游走距离将目标图划分, 使得基于拓扑结构和标签属性划分平衡达到最佳, 可以得到数据图中和查询图最相似的 Top-k (本文只取 $k=1$, 即可以递归计算 k 次) 的匹配子图. 实验证明, 算法在结构噪音或标签噪音存在的情况下, 匹配精确度仍然保持较高, 与此同时, 算法的运算效率也有较大的提高. 在未来的工作中, 本文将着重探讨卡方分布能否成功运用其他领域, 比如生物蛋白网络的全局比对等.

参考文献:

[1] ELMSALLATI A, CLARK C, KALITA J. Global-alignment-of-protein-protein-interaction-networks; asurvey [J]. IEEE/ACM Transaction on Computational Biology and Bioinformatics, 2016, 13(4): 689-705.

[2] TIANY Y, PATEL J M. TALE: a tool for approximate large graph matching [C]//Proceedings of the 2008 IEEE 24th International Conference on Data Engineer-

- ing. Cancun, Mexico:IEEE, 2008: 963-972.
- [3] ZHU L H, NGW K, CHENG J. Structure and attribute index for approximate graph matching in large graphs [J]. Information-Systems, 2011, 36(6): 958-972.
- [4] KHAN A, LI N, YANX F, et al. Neighborhood-based fast graph search in large networks [C]//Proceedings of 2011 ACM SIGMOD International Conference on Management of Data. Athens, ACM, 2011: 901-912.
- [5] KPODJED S, GALINIER P, ANTONIOL G. Using local similarity measures to efficiently address approximate graph matching [J]. Discrete Applied Mathematics, 2014(164): 161-177.
- [6] GLOVER F. Tabu search-Part I [J]. ORSA Journal on Computing, 1989, 1(3): 135-206.
- [7] DUTTA S, NAYEK P, BHATTACHARYA A. Neighbor aware search for approximate labeled graph-matching using the chi-square statistics [C]//Proceedings of the 26th International Conference on World-Wide Web. Perth: ACM, 2017: 1281-1290.
- [8] ARORA A, SACHAN M, BHATTACHARYA A. Mining statistically significant connected subgraph in vertex-labeled graphs [C]//Proceedings of 2014 ACM SIGMOD International Conference on Management of Data. Snowbird: ACM, 2014: 1003-1014.
- [9] DUTTA S, BHATTACHARYA A. Most significant substring mining based on chi-square measure [C]//Proceedings of 14th Pacific-Asia Conference Knowledge Discovery and Data Mining. Hyderabad: Springer, 2010: 21-24.
- [10] READ T R C, CRESSIE A C. Goodness of fit statistics for discrete multivariate data [M]. America: Springer, 1989: 4617-4578.
- [11] CHENG H, ZHOU Y, YU J X. Clustering large attributed graphs: a balance between structural and attribute similarities [J]. ACM Transactions on Knowledge Discovery from Data, 2015; 5(2): 12.
- [12] SUCHANEK F M, KASNECI G, WEIKUM G. Yago: a core semantic knowledge [C]//Proceedings of the 16th International Conference on World Wide Web. Banff: ACM, 2007: 679-706.

作者简介:

徐周波 女, (1976-), 博士, 教授. 研究方向为符号计算、智能规划与约束求解.

黄文文 男, (1994-), 硕士研究生. 研究方向为图数据.

刘华东(通讯作者) 男, (1978-), 讲师. 研究方向为图数据表示、符号计算. E-mail: ldd@guet.edu.cn.

杨健 男, (1994-), 硕士研究生. 研究方向为图数据挖掘.

(上接第 16 页)

- [8] KUMARI B, MAURYA A, PATTANAIK M, et al. An efficient algebraic codebook structure for CS-ACELP based speech codecs [C]//2017 8th International Conference on Computing, Communication and Networking Technologies (ICCCNT). Delhi, India. IEEE, 2017: 1-6.
- [9] HAN K. A fast search method of algebraic codebook by reordering search sequence [C]//1999 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings. ICASSP99 (Cat. No. 99CH36258). Phoenix, AZ, USA: IEEE, 1999: 21-24.
- [10] 张昕源, 高绍全, 姜汉钧, 等. 基于 G. 726 的语音无线采集芯片发射机基带电路设计 [J]. 微电子学与计算机, 2018, 35(1): 31-35.
- ZHANG X Y, GAO S Q, JIANG H J, et al. Base-

band circuit design for wireless speech acquisition transmitter based on G. 726 coding [J]. Microelectronics & Computer, 2018, 35(1): 31-35.

作者简介:

周瑾 女, (1995-), 硕士研究生. 研究方向为语音信号处理、专用集成电路设计.

乔树山 男, (1981-), 研究员. 研究方向为人工智能、低功耗集成电路设计方法学等.

凌康 男, (1981-), 硕士. 研究方向为集成电路设计.

詹毅 男, (1973-), 博士, 高级工程师. 研究方向为语音信号处理.

蒋见花(通讯作者) 女, (1976-), 博士, 副研究员. 研究方向为大规模集成电路设计技术、混合信号电路设计、电路加固技术. E-mail: jiangjianhua@ime.ac.cn.