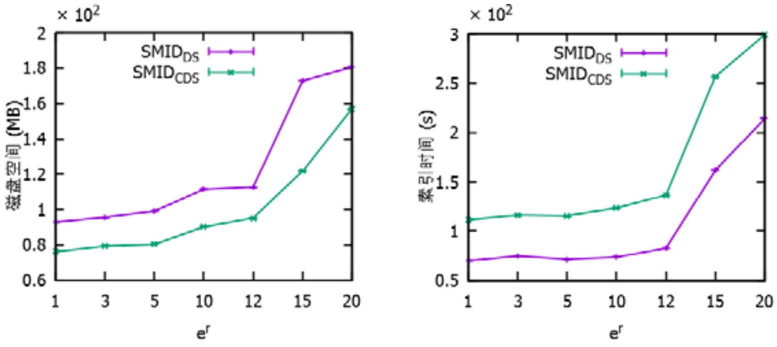
点容量越大时聚类时间越长,因此索引时间也在增长.整体而言,SMID_{CDS}的索引空间约为 SMID_{DS}索引空间的 76%~80%,因为压缩的 DS-Tree 对应的节点签名长度是未压缩的 75%,因此整体所占用的空间更小.SMID_{CDS}在索引建立的过程中需要进行折叠压缩操作,因此索引所需要的时间更长.一个有趣的现象是:当 s 过小(等于 30),索引文件大小和索引时间急剧上升,因为此时 DS-Tree 节点不断溢出和分裂.当 s=30 时,SMID_{DS}索引时间为 1 000s,索引文件达到 2G(为了使图更清楚,当 s=30 时的索引空间情况未在图中体现).因此,s 不能太小.

图 10 所示 S1M 数据集的索引大小和索引时间随 r 的变化情况.由图可知:随着 r 的增长,SMID_{DS}与 SMID_{CSD}索引空间和索引时间都呈上升趋势.因为 r 决定每一层容量的递减速度,r 越大,高层的节点容量就越小,产生的 DS-Tree 索引越大.当 r 大于一定值时,索引空间和索引时间上升得更快,因为此时高层节点不断分裂.同样,SMID_{CDS}索引空间约为 SMID_{DS}的 80%左右,因为对应的签名长度为 75%.由于压缩需要时间,因此 SMID_{CDS}需要更多的时间.



(a) s=50,索引空间随 r 变化 (b) s=50,索引时间随 r 变化

Fig.10 Impact of r on offline performance in dataset S1M

图 10 参数 r 对 S1M 数据集离线性能的影响

根据多组实验,我们对各数据集的 r,s 选择见表 2.图 11 显示了各数据集的索引空间和索引时间情况.

Table 2 Default value of r/s in each dataset

表 2 各数据集的默认参数设置

数据集	FB	DBP	S1M	S5M	S10M
s	20	20	50	50	50
e^r	3	3	10	10	10

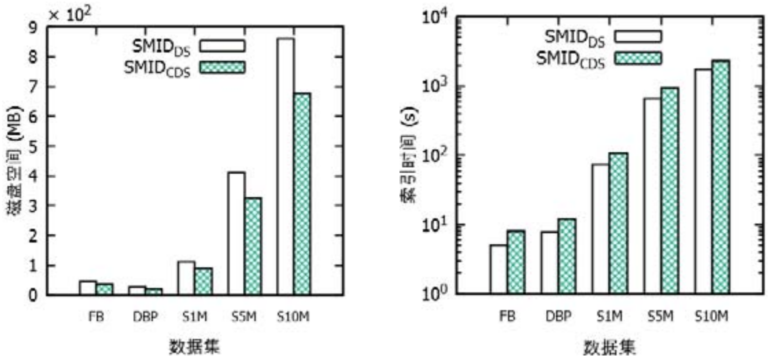


Fig.11 Index space and index time of different datasets

图 11 不同数据集的索引空间和索引时间

由图可知,索引空间和索引时间随着数据集的大小呈指数级增长.但即使对于 1000 000 000 节点的数据集,SMID$_{DS}$ 方法的索引空间约为 800M,索引时间不到 30 分钟,单机性能仍可接受.对于每个数据集,基于以上同样的原因,SMID$_{CDS}$ 方法的索引空间要小,而索引时间稍高.

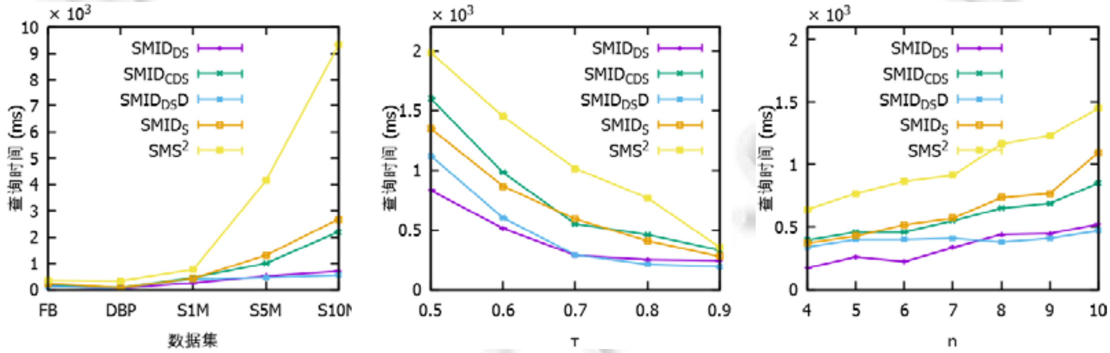
4.4 在线处理性能

在线处理阶段,我们比较 SMID$_{DS}$,SMID$_{S}$,SMID$_{CDS}$ 和 SMS^2 的查询响应时间.对于每组实验,从数据图中随机抽取 100 个特定大小的子图作为查询图.

图 12(a)表示包含度上限 τ=0.8、查询图大小 n=5 时,不同数据集上的平均查询时间.当数据量较小时,SMID$_{CDS}$,SMID$_{DS}$,SMID$_{S}$ 和 SMS^2 算法性能相差不大,但 SMID$_{CDS}$ 和 SMID$_{DS}$ 算法最佳,SMID$_{S}$ 次之,SMS^2 较差.随着数据量增大,SMID$_{DS}$ 优势愈发明显,表明 DS-Tree 索引的扩展性较好.SMID$_{CDS}$ 比 SMID$_{DS}$ 所需查询时间稍长,因为 SMID$_{CDS}$ 过滤的数据节点较少,产生的候选集较多,在验证同构时所需的时间更长.在小数据集上,SMID$_{DS}$D 查询时间比 SMID$_{DS}$ 稍长,当数据集较大时,SMID$_{DS}$D 算法略优.因为在小数据集上,验证子图同构的时间占主导地位.而大数据集对应的 DS-Tree 大,查询 DS-Tree 所需的时间更长.

图 12(b)显示的是在 S1M 数据集上,当 n=5 时,不同方法随着包含度阈值τ的变化情况.τ越小,每个节点的候选节点就越多,验证子图同构所需的时间就越多,因此查询时间越久.随着τ的增大,过滤的节点数就越多,因此总查询时间越少.对于 SMID$_{DS}$D 方法,当$\tau\geqslant$0.8 时,所需时间比 SMID$_{DS}$ 方法少,因为当包含度阈值越大,查到的候选节点少,由支配节点扩展找到非支配节点的候选节点也越少.此时,查找 DS-Tree 的时间占主要部分.随着包含度阈值增大,SMID$_{DS}$D 查找时间少于 SMID$_{DS}$ 时间.

图 12(c)比较了不同方法的查询时间随查询图大小的变化情况,此时选择数据集为 S1M,τ=0.8.对于各方法,容易理解:当查询图越小时,验证子图同构的时间越少,查询就越快.可以发现一个有趣的现象,SMID$_{DS}$D 方法随着查询图增大所需的查询时间变化不大.因为查询图变大时,支配子图的大小变化不大.因此,SMID$_{DS}$D 适合查询图较大的情况.



(a) 查询时间 VS.数据集 (b) 查询时间 VS.包含度阈值 (c) 查询时间 VS.查询图大小

Fig.12 Query time of different methods

图 12 不同方法的查询时间

上述实验结果表明:在大数据图、高包含度阈值、查询图节点较多的情况下,基于支撑节点的 SMID$_{DS}$D 方法要优于 SMID$_{DS}$ 方法.此外,本文提出的 DS-Tree 压缩方法能够在对查询时间影响不大的情况下,缩小 DS-Tree 的内存空间,这对数据图较大的情况非常有用.最后,SMID$_{DS}$ 优于基于 S-Tree 索引的查询方法 SMID$_{S}$.原因是:对于 DS-Tree,每层节点的容量是不同的,越往高层的容量越小.这符合 DS-Tree 的增长规律,即,越往高层的节点越难装满.同时,SMID$_{DS}$ 方法优于 SMS^2 方法,因为 DS-Tree 相对于 SMS^2 中的签名桶具有以下优势.

(1) 更好的适应性.签名桶基于局部敏感哈希(locality sensitive hashing,简称 LSH)[38],但 LSH 与特定应用相关,找到一个合适的哈希函数并不容易.而 DS-Tree 无需特定其他函数,能够适用不同应用的需求;

(2) 更大的灵活性.签名桶的层数需要事先指定,而层数的多少取决于数据图的大小,在创建之前难以预估.而 DS-Tree 动态增长,无需实现指定层数;

(3) 更好的扩展性.同一个数据节点可能存在于多个签名桶中,这会造成存储空间浪费.当数据量大的时候,造成索引文件非常大.而对于 DS-Tree,一个数据节点只会存在于一个 DS-Tree 的叶节点中.

5 结 论

本文提出了一种应用广泛的子图查询问题,即基于包含度的子图匹配 SMID.在 SMID 查询中,图结构必须同构,且对应节点的加权包含度大于用户给定阈值.此外,给出了一种基于 DS-Tree 和最小支配子图的 SMID 查询算法.最小支配子图算法在数据图较大、包含度阈值较高和查询图较大的情况下,能够加快查询效率.针对 DS-Tree 占用内存空间高的问题,提出了一种 DS-Tree 的压缩算法,在对查询效率影响不大的情况下,缩小了内存空间.通过实验证明,本文给出的 SMID 算法在单机上具有较高的查询效率和较好的扩展性.下一步工作,将集中精力在计算机集群中实现 SMID 查询.

References:

[1] Yu X, Sun Y, Zhao P, Han J. Query-Driven discovery of semantically similar substructures in heterogeneous networks. In: Proc. of the ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining. New York: ACM Press, 2012. 1500−1503. [doi: 10.1145/2339530.2339765]

[2] Zou L, Mo J, Chen L, Özsu MT, Zhao D. gStore: Answering SPARQL queries via subgraph matching. Proc. of the VLDB Endowment, 2011,4(8):482−493. [doi: 10.14778/2002974.2002976]

[3] Tian Y, Patel JM. TALE: A tool for approximate large graph matching. In: Proc. of the 2008 IEEE 24th Int'l Conf. on Data Engineering. IEEE Computer Society, 2008. 963−972. [doi: 10.1109/ICDE.2008.4497505]

[4] Zhao P, Han J. On graph query optimization in large networks. Proc. of the VLDB Endowment, 2010,3(1-2):340−351. [doi: 10.14778/1920841.1920887]

[5] Zhang S, Yang J, Jin W. Sapper: Subgraph indexing and approximate matching in large graphs. Proc. of the VLDB Endowment, 2010,3(1-2):1185−1194. [doi: 10.14778/1920841.1920988]

[6] Sun Z, Wang H, Wang H, Shao B, Li J. Efficient subgraph matching on billion node graphs. Proc. of the VLDB Endowment, 2012, 5(9):788−799. [doi: 10.14778/2311906.2311907]

[7] Qu KS, Zhai YH. Posets, inclusion degree theory and FCA. Chinese Journal of Computers, 2006,29(2):219−226 (in Chinese with English abstract). [doi: 10.3321/j.issn:0254-4164.2006.02.004]

[8] Ren X, Liu J, Yu X, Khandelwal U, Wang L, Han J. Cluscite: Effective citation recommendation by information network-based clustering. In: Proc. of the 20th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining. ACM Press, 2014. 821−830. [doi: 10.1145/2623330.2623630]

[9] Duan L, Street WN, Liu Y, Lu H. Community detection in graphs through correlation. In: Proc. of the 20th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining. ACM Press, 2014. 1376−1385. [doi: 10.1145/2623330.2623629]

[10] Cordella LP, Foggia P, Sansone C, Vento M. A (sub) graph isomorphism algorithm for matching large graphs. IEEE Trans. on Pattern Analysis and Machine Intelligence, 2004,26(10):1367−1372. [doi: 10.1109/TPAMI.2004.75]

[11] Zhao X, Xiao C, Lin X, Wang W, Ishikawa Y. Efficient processing of graph similarity queries with edit distance constraints. VLDB Journal, 2013,22(6):727−752. [doi: 10.1007/s00778-013-0306-1]

[12] Ullmann JR. An algorithm for subgraph isomorphism. Journal of the ACM, 1976,23(23):31−42. [doi: 10.1145/321921.321925]

[13] Shang H, Zhang Y, Lin X, Yu J. Taming verification hardness: An efficient algorithm for testing subgraph isomorphism. Proc. of the VLDB Endowment, 2008,1(1):364−375. [doi: 10.14778/1453856.1453899]

[14] Han WS, Lee J, Lee JH. Turbo iso: Towards ultrafast and robust subgraph isomorphism search in large graph databases. In: Proc. of the ACM SIGMOD Int'l Conf. on Management of Data. 2013. 337−348. [doi: 10.1145/2463676.2465300]

[15] Ma H, Shao B, Xiao Y, Chen LJ, Wang H. G-SQL: Fast query processing via graph exploration. Proc. of the VLDB Endowment, 2016,9(12):900−911. [doi: 10.14778/2994509.2994510]

[16] He H, Singh AK. Closure-Tree: An index structure for graph queries. In: Proc. of the 22nd Int’l Conf. on Data Engineering (ICDE 2006). IEEE, 2006. 38−38. [doi: 10.1109/ICDE.2006.37]

[17] Zhang XC, Yu H, Gong XJ. A random walk based iterative weighted algorithm for sub-graph query. Journal of Computer Research and Development, 2015,52(12):2824−2833 (in Chinese with English abstract). [doi: 10.7544/issn1000-1239.2015.20140801]

[18] Yuan Y, Wang G, Xu JY, Chen L. Efficient distributed subgraph similarity matching. The VLDB Journal, 2015,24(3):369−394. [doi: 10.1007/s00778-015-0381-6]

[19] Gupta M, Gao J, Yan X, Cam H, Han J. Top-*k* interesting subgraph discovery in information networks. In: Proc. of the 2014 IEEE 30th Int’l Conf. on Data Engineering. IEEE, 2014. 820−831. [doi: 10.1109/ICDE.2014.6816703]

[20] Rangapuram SS, Bühler T, Hein M. Towards realistic team formation in social networks based on densest subgraphs. In: Proc. of the 22nd Int’l Conf. on World Wide Web. ACM Press, 2013. 1077−1088. [doi: 10.1145/2488388.2488482]

[21] Khan A, Wu Y, Aggarwal CC, Yan X. Nema: Fast graph search with label similarity. Proc. of the VLDB Endowment, 2013,6(3): 181−192. [doi: 10.14778/2535569.2448952]

[22] Zou L, Chen L, Lu Y. Top-*k* subgraph matching query in a large graph. In: Proc. of the ACM First Ph. D. Workshop in CIKM. ACM Press, 2007. 139−146. [doi: 10.1145/1316874.1316897]

[23] Peng P, Zou L, Özsu MT, Chen L. Processing SPARQL queries over distributed RDF graphs. Computer Science, 2016,25(2):1−26. [doi: 10.1007/s00778-015-0415-0]

[24] Zheng W, Zou L, Lian X, Wang D, Zhao D. Efficient graph similarity search over large graph databases. IEEE Trans. on Knowledge and Data Engineering, 2015,27(4):964−978. [doi: 10.1109/TKDE.2014.2349924]

[25] Zheng W, Zou L, Peng W, Yan X, Song S, Zhao D. Semantic SPARQL similarity search over RDF knowledge graphs. Proc. of the VLDB Endowment, 2016,9(11):840−851. [doi: 10.14778/2983200.2983201]

[26] Lian X, Chen L, Wang G. Quality-Aware subgraph matching over inconsistent probabilistic graph databases. IEEE Trans. on Knowledge and Data Engineering, 2016,28(6):1560−1574. [doi: 10.1109/TKDE.2016.2518683]

[27] Hong L, Zou L, Lian X, Philip SY. Subgraph matching with set similarity in a large graph database. IEEE Trans. on Knowledge & Data Engineering, 2015,27(9):2507−2521. [doi: 10.1109/TKDE.2015.2391125]

[28] Garey MR, Johnson DS. Computers and Intractability: A Guide to the Theory of NP-Completeness. W.H. Freeman and Company, 1979.

[29] Deppisch U. S-Tree: A dynamic balanced signature index for office retrieval. In: Proc. of the 9th Annual Int’l ACM SIGIR Conf. on Research and Development in Information Retrieval. ACM Press, 1986. 77−87. [doi: 10.1145/253168.253189]

[30] Mamoulis N, Cheung DW, Lian W. Similarity search in sets and categorical data using the signature tree. In: Proc. of the 19th Int’l Conf. on Data Engineering. IEEE, 2003. 75−86. [doi: 10.1109/ICDE.2003.1260783]

[31] Chen Y, Chen Y. On the signature tree construction and analysis. IEEE Trans. on Knowledge and Data Engineering, 2006,18(9): 1207−1224. [doi: 10.1109/TKDE.2006.146]

[32] Kontaki M, Manolopoulos Y, Nanopoulos A. Compressing large signature trees. Lecture Notes in Computer Science, 2003,2798: 163−177. [doi: 10.1007/978-3-540-39403-7_14]

[33] Fomin FV, Grandoni F, Pyatkin AV, Stepanov AA. Combinatorial bounds via measure and conquer: Bounding minimal dominating sets and applications. ACM Trans. on Algorithms, 2008,5(1):596−600. [doi: 10.1145/1435375.1435384]

[34] Couturier JF, Heggernes P, van’t Hof P, Krastsch D. Minimal dominating sets in graph classes: Combinatorial bounds and enumeration. Theoretical Computer Science, 2013,487:82−94. [doi: 10.1016/j.tcs.2013.03.026]

[35] Rooij JMMV. Exact Exponential-Time Algorithms for Domination Problems in Graphs. Boxpress, 2011.

[36] Yang Y, Pedersen JO. A comparative study on feature selection in text categorization. In: Proc. of the 14th Int’l Conf. on Machine Learning. Morgan Kaufmann Publishers Inc., 1998. 412−420.

[37] Viger F, Latapy M. Efficient and simple generation of random simple connected graphs with prescribed degree sequence. 2005, 3595:440−449. [doi: 10.1007/11533719_45]

[38] Chakrabarti A, Parthasarathy S. Sequential hypothesis tests for adaptive locality sensitive hashing. Computer Science, 2014. [doi: 10.1145/2736277.2741665]

附中文参考文献:

[7] 曲开社,翟岩慧.偏序集、包含度与形式概念分析.计算机学报,2006,29(2):219−226. [doi: 10.3321/j.issn:0254-4164.2006.02.004]

[17] 张小驰,于华,宫秀军.一种基于随机游走的迭代加权子图查询算法.计算机研究与发展,2015,52(12):2824−2833. [doi: 10.7544/issn1000-1239.2015.20140801]

李瑞远(1990－),男,湖南郴州人,博士生,主要研究领域为时空数据管理,城市计算,云计算.

洪亮(1982－),男,博士,副教授,主要研究领域为图数据库,社会网络,时空数据管理.