دسته‌بندی و ارزیابی الگوریتم‌های کاوش زیرگراف‌های تکراری

محمدرضا کیوان‌پور[1]؛ فرشته عزیزانی[2]

چکیده

در سال‌های اخیر افزایش سرعت ایجاد پایگاه‌داده‌های گراف موجب شده است که توجه فراوانی به داده‌کاوی میان گراف‌ها یا گراف‌کاوی جلب شود. پایگاه‌داده‌ی گراف، نوع خاصی از پایگاه‌داده است که معمولا شامل یک گراف بزرگ و یا چندین گراف کوچک می باشد. برخی از کاربردهای ایجاد پایگاه‌داده‌های گراف عبارتند از: شبکه‌های زیستی، وب معنایی، مدلسازی رفتار و.... در میان الگوهای متفاوتی که در پایگاه‌داده‌ی گراف وجود دارد، کاوش زیرگراف‌های تکراری از اهمیت زیادی برخوردار است. زیر گراف تکراری زیر گرافی است که به صورت مکرر در پایگاه‌داده‌ی گراف دیده می شود. زیرگراف‌های تکراری نه تنها به خودی خود دارای ارزش می‌باشند، بلکه در سایر زمینه‌های تحلیل داده و روش‌های داده‌کاوی نیز قابل استفاده هستند. از جمله‌ی این زمینه‌ها می‌توان تسهیل جستجوی مشابهت در پایگاه‌داده‌ی گراف، خوشه‌بندی، دسته‌بندی و شاخص‌گذاری گراف‌ها را نام برد. تاکنون الگوریتم‌های مختلفی برای کاوش زیرگراف‌های تکراری ارائه شده‌اند. این مقاله می‌کوشد تا با معرفی این الگوریتم‌ها و مقایسه‌ی مشخصات آنها ، یک دید کلی نسبت به الگوریتم‌های کاوش زیرگراف‌های تکراری ایجاد کند. الگوریتم‌های موجود بر اساس نوع پایگاه‌داده‌ی گرافی که روی آن عمل می‌کنند و روشی که برای ایجاد زیرگراف‌های تکراری بکارمی‌گیرند، دسته‌بندی شده‌اند. با توجه به اهمیت روزافزون زمینه‌های کاربردی زیرگراف‌های تکراری، دسته‌بندی پیشنهادی می‌تواند در انتخاب الگوریتم مناسبِ کاربردها و شناسایی روش‌های نوین گراف‌کاوی در این زمینه موثر باشد.

کلمات کلیدی

پایگاه‌داده‌ی گراف، داده‌کاوی، گراف‌کاوی، زیرگراف تکراری

Classification and Analyze of Frequent Subgraph Mining Algorithms

Mohammad Reza Keyvanpour; Fereshteh Azizani

ABSTRACT

In recent years, data mining in graphs or graph mining have attracted much attention due to explosive growth in generating graph databases. The graph database is one type of database that consists of either a single large graph or a number of relatively small graphs. Some applications of graph database are as followings: Biological networks, semantic web and behavioral modeling. Among all patterns occurring in graph database, mining frequent subgraph is of great importance. The frequent subgraph is the one that occur frequently in the graph database. Frequent subgraphs not only are important themselves but are applicable in other aspects of data analyze and data mining tasks, such as similarity search in graph database, graph clustering, classification, indexing, etc. So far, numerous algorithms are proposed for mining frequent subgraph. This study aims to create overall view of the algorithms through the analysis and comparison of their characterizations. To achieve the aim, the existing algorithms are classified on the basis of its graph database and subgraph generation way. The proposed classification can be effective in choosing applications appropriate algorithms and determination of graph mining new methods in this regard.

KEYWORDS

Graph database, Data mining, Graph mining, Frequent subgraph

---

[1] عضو هیأت علمی دانشگاه الزهراء (س)، دانشکده فنی مهندسی ، پست الکترونیک: keyvanm@modares.ac.ir

[2] دانشجوی کارشناسی ارشد دانشگاه آزاد اسلامی واحد قزوین، دانشکده برق، رایانه و فن‌آوری اطلاعات، پست الکترونیک: f.azizani@qazviniau.ac.ir

۱. مقدمه

نمایش داده‌ها به شکل گراف، امکان بیان ارتباطات موجود بین داده‌ها را به صورت طبیعی ممکن می‌سازد. این ویژگی گراف‌ها، موجب بکارگیری روزافزون آنها برای مدلسازی ساختارهای پیچیده مانند تصاویر[۸]، مولفه‌های شیمیایی[۲۰]، ساختارهای پروتئین[۲۹]، شبکه‌های زیستی[۲۸]، شبکه‌های اجتماعی[۴۱]، وب[۳۰] و اسناد XML [۵] شده است. با وجود سرعت ایجاد و افزایش تعداد گراف‌های حاصل از مدل-سازی ساختارهای پیچیده، بکارگیری روشی که بتواند این حجم وسیع اطلاعات را به صورت کارا تحلیل نماید ضروری است. این مساله موجب تبدیل داده‌کاوی میان گراف‌ها یا گراف‌کاوی، به یکی از زمینه‌های پراهمیت داده‌کاوی شده‌است.

از میان الگوهای متفاوتی که در بین گراف‌ها وجود دارد ، کاوش زیرگراف‌های تکراری از اهمیت زیادی برخوردار است. زیر گراف تکراری زیر گرافی است که بصورت مکرر در پایگاه‌داده‌ی گراف دیده می شود. زیرگراف‌های تکراری نه تنها به خودی خود دارای ارزش می‌باشند، بلکه در سایر زمینه‌های تحلیل داده و روش‌های داده‌کاوی نیز قابل استفاده هستند. از جمله‌ی این زمینه‌ها می‌توان دسته‌بندی، جستجوی مشابهت، خوشه‌بندی و شاخص‌گذاری درگراف‌ها را نام برد. در ارتباط با دسته‌بندی و جستجوی مشابهت، بهره‌گیری از زیرگراف‌های تکراری به عنوان ویژگی، منجر به نتایج دقیق و مقیاس‌پذیری بالایی خواهد شد [۳۹،۳۸،۷]. خوشه‌بندی فضاهایی با ابعاد بالا، بسیار چالش‌برانگیز است. از آنجایی که استخراج زیرگراف‌های تکراری در زیرمجموعه‌هایی از ابعاد به سهولت امکان‌پذیر است، می‌توان از زیرگراف‌های تکراری به عنوان راهکاری در خوشه‌بندی زیرفضا و خوشه‌بندی فضاهایی با ابعاد بالا استفاده کرد[۳۱]. جستجوی کارا در پایگاه‌داده‌ی گراف، در کاربردهای زیادی که از جمله‌ی آنها می‌توان کشف ساختارهای سرطانی را نام برد، یک مساله‌ی اجتناب ناپذیر است. حجم بالای داده در این پایگاه‌ها، امکان جستجوی ترتیبی و آزمایش تک تک اشیاء را غیرممکن و ناکارامد می‌سازد. با بهره‌گیری از راهکارهای کاوش زیرگراف‌های تکراری می‌توان تنها با شاخص‌گذاری گراف‌های تکراری، سرعت جستجو را به صورت چشم‌گیری بالا برد[۳۷].

فرآیند استخراج زیرگراف‌های تکراری، فرآیندی تکراری است که معمولا از دو مرحله‌ی اصلی تشکیل شده است[۲۴]. مرحله‌ی اول تولیدِ کاندید است. در این مرحله زیرگراف‌هایی که احتمال تکراری بودن آنها وجود دارد، یا به عبارت دیگر کاندید تکراری بودن هستند، تولید می‌شوند. مرحله‌ی بعدی، مرحله‌ی شمارش است. در این مرحله تعداد دفعاتی است که کاندیدهای تولیدشده در پایگاه‌داده دیده می‌شوند، شمارش می‌شود. برای این منظور، باید کاندید مورد نظر در پایگاه‌داده‌ی گراف جستجو شود. رویکرد این مرحله بسته به نوع پایگاه‌داده‌ی گرافی که فرآیند گراف-کاوی روی آن انجام می‌گیرد، می‌تواند متفاوت باشد. چنانچه پایگاه‌داده تنها شامل یک گراف بزرگ مجرد باشد[۲۱]، تعداد رخداد زیرگراف در آن شمارش می‌شود. اما اگر پایگاه‌داده از تعداد زیادی گراف کوچک تشکیل شده باشد، تعداد رخداد زیرگراف در یک گراف خاص اهمیتی ندارد و برابر با تعداد گراف‌هایی که آن زیرگراف در آنها جود دارد، خواهد بود[۱۷]. الگوریتم‌های تولید شده برای کاوش روی یک گراف مجرد را می‌توان برای مجموعه‌ای از گراف‌ها نیز بکاربرد، اما حالت عکس این قضیه امکان پذیر نیست. در هر دو صورت شمارش تعداد رخداد کاندیدها، نیازمند بررسی همریختی زیرگراف‌ها[۱] است که یک مساله‌ی NP-complete می‌باشد[۱۰] و به‌ویژه برای گراف‌های بزرگ هزینه‌ی بسیار زیادی خواهد داشت.

با توجه به اهمیت روزافزون زیرگراف‌های تکراری این مقاله می‌کوشد تا با معرفی این الگوریتم‌ها و ارائه‌ی مشخصات آنها یک دید کلی نسبت به الگوریتم‌های کاوش زیرگراف‌های تکراری ایجاد نماید. ازآنجایی‌که اکثر پژوهشات موجود در این زمینه، بر مرحله‌ی تولید کاندید متمرکز شده‌اند و سعی دارند الگوریتم‌هایی را ایجاد کنند که حداقل تعداد زیرگراف‌ها را در حداقل زمان ممکن کاوش کنند، تمرکز اصلی این مقاله نیز بر انواع روش‌های تولید کاندید خواهد بود. در این راستا، الگوریتم‌های موجود بر اساس نوع روشی که برای تولید کاندید بکارمی‌گیرند، دسته‌بندی می-شوند. کاربردها برای انتخاب الگوریتم مناسبِ خود، نیاز به ارزیابی روش‌های موجود دارند. اگرچه ارزیابیِ مستقیم و همه‌جانبه‌یِ الگوریتم‌های کاوش زیرگراف تکراری ممکن نیست، اما این مقاله سعی می‌کند تا با ارائه‌ی مشخصات بارز آنها در قالب یک جدول، پژوهشگران را در انتخاب الگوریتم مورد نظرشان راهنمایی کند.

ساختار مقاله بدین‌گونه خواهد بود. در بخش دوم، مساله‌ی کاوش زیرگراف تکراری تعریف می‌شود و مفاهیم پایه‌ای و تعاریف کلی که در کاوش زیرگراف‌ها مطرح هستند، ارائه خواهند شد. بخش سوم انواع الگوریتم‌های کاوش زیرگراف تکراری را به تفکیک نوع پایگاه‌داده‌ای که با آن کار می‌کنند و نوع روش تولید کاندیدی که بکار می‌گیرند دسته‌بندی می‌کند. بخش چهارم به ارزیابی قیاسی الگوریتم‌ها می‌پردازد و در بخش پنج نیز مقاله با نتیجه‌گیری پایان می‌یابد.

۲. کاوش زیرگراف تکراری

اگر D پایگاه‌داده‌ی ورودی (اعم از یک گراف مجرد یا مجموعه‌ای از گراف‌ها) باشد، هدف از کاوش زیرگراف تکراری، کاوش زیرگراف‌هایی است که دارای مقدار پشتیبانِ[۲] بیشتری نسبت به حد آستانه‌ی از پیش تعیین شده داشته باشند[۱]. مقدار پشتیبانِ زیرگراف G_S با $sup(G_s)$ نمایش داده می‌شود و با استفاده از رابطه‌ی (۱) بدست می‌آید.

$$sup(G_S) = \frac{\text{تعداد زیرگراف } G_S \text{ موجود در } D}{\text{تعدادِ کل زیرگراف‌هایِ موجود در } D} \qquad (۱)$$

چنانچه G_S زیرگرافی از گراف G باشد، آنگاه رابطه‌ی $sup(G) \leq sup(G_S)$ برقرار است. در واقع می‌توان گفت تعداد تکرار زیرگراف با طول آن نسبت عکس دارد. زیرگراف‌هایی که این ویژگی را ندارند، یعنی با افزایش طولشان تکرارشان نیز افزایش می‌یابد،زیرگراف‌های نامرتبط نامیده می‌شوند. از این ویژگی معیار پشتیبان می‌توان برای هرس زودهنگامِ زیرگراف‌های غیرتکراری استفاده کرد[۱۳]. زیرا اگر زیرگرافی پشتیبانِ کمتری از حدآستانه داشته باشد (تکراری نباشد) زیرگراف‌های بعدی ایجاد شده از آن نیز با توجه به ویژگی معیار پشتیبان، پشتیبانِ کمتری خواهند داشت (تکراری نخواهند بود) و می‌توان آنها را هرس کرد.

کاوش زیرگراف تکراری معمولا از گره‌ها و یال‌های تکراری شروع می‌شود. در واقع در مرحله‌ی اول زیرگراف‌های تکراری با یک گره یا یک یال کاوش می‌شوند. این امر به راحتی با شمارش تعداد تکرارِ گره‌ها و یال‌های موجود در پایگاه‌داده و حذف آنهایی که مقدار پشتیبانشان کمتر ازحد آستانه است، امکان‌پذیر است. در مرحله‌ی بعد زیرگراف جدیدی، با اضافه کردن گره‌ها و یال‌های جدید به زیرگراف نتیجه شده از مرحله‌ی قبل تولید می‌شود. از آنجایی که زیرگراف جدید تولید شده ممکن است دیگر تکراری نباشد، زیرگراف کاندید یا به طور مختصر کاندید نامیده می‌شود[۳۲]. سپس تکراری بودن کاندید یا کاندیدهای ایجاد شده بررسی و آنهایی که پشتیبانشان از حدآستانه بیشتر است یا با آن مساوی است، به عنوان ورودی مرحله‌ی بعد برگردانده می‌شوند. این فرآیند تا زمانی که زمان اجرای الگوریتم به یک حد از پیش تعیین شده برسد، یا تمامی زیرگراف‌های تکراری کاوش شوند ادامه می‌یابد. در ادامه دو مفهوم پراهمیت در این زمینه معرفی خواهند شد.

بررسی همریختی زیرگراف- برای تعیین تعداد تکرار کاندید لازم است تا زیرگراف‌های همریخت با آن در D کاوش شوند. دو زیرگراف همریخت هستند اگر از نظر همبندی کاملا مشابه هم باشند. به عبارت دیگر، دو گراف $G_1=(V_1,E_1)$ و $G_2=(V_2,E_2)$ در صورتی همریختند که بتوان نگاشتی از V_1 به V_2 را چنان یافت که هر یال در E_1 به یک یال منفرد در E_2 نگاشت شود و بالعکس. در مورد گراف‌های برچسب‌دار، برچسب‌ها نیز باید به نگاشت مربوطه اضافه شوند. هر همریختی از یک گراف به خودش، خودریختی[۳] نامیده می‌شود[۲۱]. همریختی را می‌توان به دو صورت دقیق و تقریبی انجام داد. در روش دقیق، دو زیرگراف تنها در صورتی همریخت تشخیص داده می‌شوند که کاملا همبندیِ مشابهی داشته باشند. اما در نوع تقریبی دو گراف حتی در صورت وجود تفاوت‌های جزئی همریخت تشخیص داده می‌شوند[۱۵،۴]. زیرگراف‌های همریخت با گراف G_S در D، تعبیه‌های Gs نامیده می‌شوند[۳۴].

برچسب‌گذاری استاندارد- بررسی همریختی زیرگراف‌ها پیچیدگی محاسباتی بالایی دارد. برای حل این مشکل می‌توان با استفاده از برچسب گره‌ها و یال‌ها، به هر زیرگراف یک کد یکتا نسبت داد. این کد برچسب استاندارد گراف نام دارد. بدین ترتیب به جای بررسی یکسان بودن دو گراف، کافی است بررسی شود که دو گراف برچسب استاندارد یکسانی دارند یا خیر[۹] . پیچیدگی محاسباتی برچسب‌گذاری استاندارد نیز در بدترین حالت نمایی خواهد بود. الگوریتم‌های مختلف با تعریف برچسب‌های استاندارد گوناگون سعی در تخفیف این مشکل دارند.

۳. دسته‌بندي راهکارهاي توليد کانديد

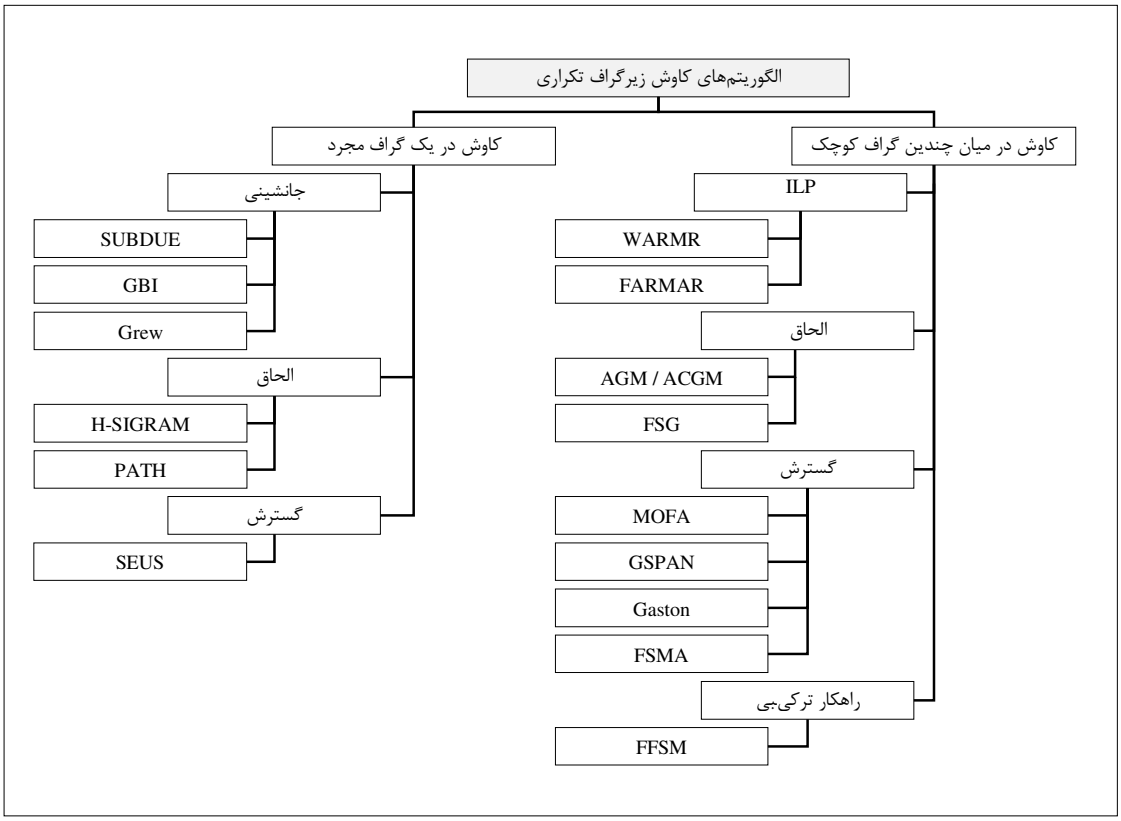
تولید کاندیدهای افزونه یا همریخت، کاندیدهایی که تکراری نیستند و در پایگاه‌داده وجود ندارند سه چالش اصلی فراروی فرآیند تولید کاندید است. الگوریتم‌های کاوش زیرگراف‌های تکراری از راهکارهای متفاوتی برای رفع یا تخفیف این چالش‌ها استفاده می‌کنند. در شکل (۱) الگوریتم‌های مختلف بر اساس نوع پایگاه‌داده‌ای که روی آن عمل می‌کنند و نوع روشی که برای تولید کاندید بکار می‌گیرند، دسته‌بندی شده‌اند. در ادامه‌ی این بخش این الگوریتم‌ها بر اساس روشی که برای تولید کاندید بکارمی‌گیرند، معرفی می‌شوند.

۱.۳. برنامه‌نويسي منطق استقرايي

در این راهکار زیرگراف‌ها به جای گراف برچسبدار، با استفاده از برنامه‌نویسی منطقی استقرایی (ILP)[۴] [۲۵] به صورت گزاره‌های مرتبه‌ی اول نمایش داده می‌شوند. برای مثال در یک پایگاه‌داده‌ی مولکولی، گزاره‌های مرتبه‌ی اولی مانند atomel(C,A_1,c) ، atomel(C,A_2,c) ، bond(C,A_1, A_2, BT) به ترتیب یعنی؛اتم کربن در مولفه‌ی شیمیایی C با پارامتر A_1 نمایش داده می‌شود، اتم گوگرد در مولفه‌ی شیمیایی C با پارامتر A_2 نمایش داده می‌شود و این دو اتم با پیوندی از نوع BT به هم متصل شده‌اند. این نوع نمایش گراف، دارای دو مزیت می‌باشد. اول، بدون

در نظر گرفتن نوع همبندی پایگاه‌داده، می‌توان هر نوع پایگاه‌داده‌ای را با گزاره‌های مرتبه‌ی اول به سادگی نمایش داد. دوم، می‌توان دو گره در زیرگراف را به یک پارامتر در پایگاه داده نگاشت کرد، درحالیکه در سایر الگوریتم‌های کاوش زیرگراف این کار ممکن نیست[۳]. بدین ترتیب مساله-ی کاوش زیرگراف‌های کاندید به مساله‌ی کاوش گزاره‌های مرتبه‌ی اول کاندید تبدیل می‌شود.

WARMAR [۶] اولین سیستمی است که برای تولید کاندید از راهکار برنامه‌نویسی منطق استقرایی استفاده می‌کند. کاوش گزاره‌های تکراری نیازمند بررسی هم‌ارزی گزاره‌هاست. اما از آنجایی که هیچ رده‌بندی خاصی برای بررسی هم‌ارزی گزاره‌ها در این سیستم وجود ندارد، پیچیدگی محاسباتی آن بسیار بالا است. FARMAR [۲۷] برای حل این مشکل از شرط هم‌ارزی ضعیف‌تری استفاده می‌کند. یعنی حتی در صورت وجود اختلافات جزئی دو گزاره، آنها را هم‌ارز تلقی می‌کند. این موضوع باعث می‌شود در خروجی تعداد زیادی گزاره‌ی هم‌ارز با اشکال متفاوت وجود داشته باشند. چالش‌های موجود در مرحله‌ی شمارش این راهکار، آن را به راهکار نامناسبی تبدیل می‌کند. مزیت اصلی این نوع روش تولید کاندید، قدرت نمایش بالای الگوهای کشف شده می‌باشد.



شکل (۱) دسته‌بندی الگوریتم‌های کاوش زیرگراف تکراری

۲.۳. الحاق

ایده‌ی اصلی تولید کاندید با استفاده از الحاق، در الگوریتم apriori مطرح شد[۱]. این الگوریتم برای کاوش مجموعه‌اقلام تکراری ایجادشده‌است و مجموعه اقلامِ تکراری با اندازه‌ی (k+۱) را با الحاق دو مجموعه اقلام تکراری با اندازه‌ی k بوجود می‌آورد. با این روش apriori از خاصیت معیار پشتیبان که در بخش دوم توضیح داده شد، برای حذف مجموعه‌اقلام غیرتکراری با اندازه‌ی k، استفاده می‌کند. تعمیم این ایده و مناسب کردن آن برای کاوش گراف‌ها، الگوریتمِ شکل (۲) را نتیجه می‌دهد[۱۱]. الگوریتم‌های زیادی از این روش، استفاده می‌کنند. تفاوت این الگوریتم‌ها در نوع قطعه‌ی ساخت و شرطی است که برای الحاق بکارمی‌برند. قطعه‌ی مورد نظر بنابه نوع الگوریتم می‌تواند گره، یال یا مسیر باشد.

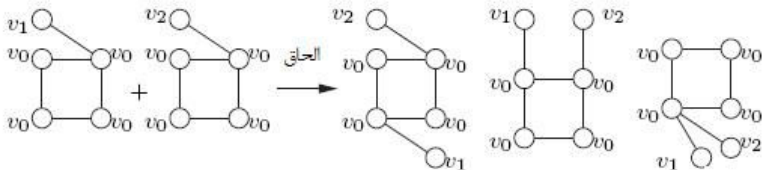
تولید کاندید با این روش سه نوع هزینه در برخواهد داشت. اولین هزینه مربوط به تشخیص هسته، دوم هزینه‌ی الحاق زیرگراف‌ها و سوم شمارش تعداد تکرار زیرگراف است[۴]. از آنجایی که در این مقاله راهکارهای تولید کاندید مدنظر هستند، تنها راهکارهایی کاهش دو هزینه‌ی اول معرفی می‌شوند. یکی از راهکارهای کاهش این هزینه‌ها، کاهش تعداد کاندیدها است. AGM [۱۹،۱۸،۱۷] با بهره‌گیری از این ایده تنها زیرگراف‌های القایی پایگاه‌داده را که دارای پشتیبان معین هستند، می‌یابد. زیر گراف $Gs(Vs,Es)$ از $G(V,S)$ القایی است اگر و تنها اگر، تمام یال‌های موجود

بین گره‌های Vs، بین همین رئوس در G نیز موجود باشند. این الگوریتم برای تولید کاندید از گره استفاده می‌کند، در واقع دو کاندید تکراری با اندازه‌ی k را زمانی پیوند می‌زند که بخش مشترکی با k-۱ گره داشته باشند. گرچه کاوش زیرگراف‌های القایی فضای جستجوی کاندیدها را کاهش می‌دهد، اما کاندیدهایی که القایی نیستند توسط الگوریتم قابل تشخیص نمی‌باشند.

۱. یافتن تمام زیرگراف‌هایی که شامل یک قطعه هستند و حذف غیرتکراری‌ها.
۲. یافتن تمام کاندیدهای که شامل دو قطعه هستند و حذف غیرتکراری‌ها.
۳. در مرحله‌ی n :
   أ ایجاد کاندیدهایی با n+۱ قطعه بوسیله‌ی الحاق کردن زیرگراف‌های n قطعه‌ای که دارای هسته‌ی مشترک هستند. هسته‌ی مشترک، بخش مشترک دو زیرگراف است که دارای n-۱ قطعه می‌باشد.
   ب حذف کاندیدهای غیرتکراری.
   ت توقف الگوریتم در صورت عدم وجود کاندید تکراری.

شکل (۲) تولید زیرگراف تکراری با استفاده از الحاق

یک زیرگراف القایی می‌تواند گراف ناهمبندی باشد که از بخش‌های مجزا تشکیل شده است. از آنجایی که در بیشتر کاربردها گراف‌های همبند مورد نظر هستند، محدود کردن جستجو به زیرگراف‌های همبند تاثیر زیادی بر کاربردی بودن گراف‌گاوی نمی‌گذارد. الگوریتم FSG [۲۲] با بهره-گیری از این ایده، تنها زیرگراف‌های همبند تکراری در پایگاه‌داده را می‌یابد. FSG از قطعه‌ی ساخت یال استفاده می‌کند و راهکارهای متفاوتی را برای کاهش هزینه‌های تولید کاندید بکار می‌گیرد. FSG برای کاهش هزینه‌ی تشخیص هسته، برچسب‌های استاندارد تمامی (k-۱)-زیرگراف‌هایِ (زیرگراف‌هایی با اندازه‌ی k-۱) یک k-گراف را ذخیره می‌کند. بدین‌ترتیب می‌تواند هسته‌ی مشترک بین دو k-گراف را تنها با اشتراک گرفتن بین مجموعه‌ی برچسب استاندارد زیرگراف‌هایشان محاسبه کند. الحاق دو زیرگراف منجر به تولید گراف‌های زیادی خواهد شد. همانطور که در شکل (۳) نشان داده شده‌است، یکی از دلایل این امر می‌تواند وجود چندین خودریختی در هسته باشد. FSG در مرحله‌ی تشخیص هسته خودریختی-های موجود در هر هسته را ذخیره می‌کند. بنابراین در مرحله‌ی الحاق، به جای محاسبه‌ی دوباره‌ی خودریختی‌ها از خودریختی‌های ذخیره شده استفاده می‌کند و هزینه‌ی الحاق را کاهش می‌دهد.



شکل (۳) ایجاد چندین زیرگراف از الحاق دو ماتریس به دلیل وجود

FSG برای تولید کاندید، تمامی زیرگراف‌هایی که دارای هسته‌ی مشترک هستند، را الحاق می‌کند. این مساله موجب ایجاد کاندیدهای بسیار زیادی می‌شود، که آن را با وجود تکنیک‌های تولید کاندیدی که بکار می‌گیرد، در پایگاه‌داده‌های بزرگ ناکارامد می‌سازد. HSIGRAM [۲۱] برای کاهش تعداد کاندیدها، دو زیرگراف را تنها در صورتی که زیرگراف‌های اصلی مشترک نداشته باشند الحاق می‌کند. دو (k-۱)-زیرگراف از گراف G، در صورتی که در میان تمام برچسب‌های کانونی (k-۱)-زیرگراف‌ها، کوچکترین برچسب‌ها را داشته باشند، زیرگراف‌های اصلیِ G نامیده می‌شوند. این روش تولید کاندید به میزان زیادی تعداد زیرگراف‌های افزونه و غیرمرتبط را کاهش می‌دهد. یکی دیگر از راهکارهای کاهش کاندید، افزایش اندازه‌ی قطعه‌ی ساخت در الگوریتم ارائه شده در شکل (۲) است. [۱۱] و [۳۳] با بهره‌گیری از این ایده، از قطعه‌ی مسیر استفاده می‌کنند و بدین ترتیب با تعداد تکرارهای کمتری زیرگراف‌های تکراری را می‌یابند (این الگوریتم‌ها در شکل (۱) و جدول (۱) با نام Path مشخص شده‌اند) .

۳.۳. جایگزینی

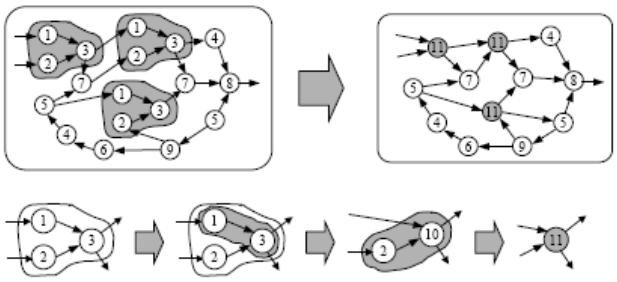
در این راهکار بعد از یافتن زیرگراف تکراری در هر مرحله، زیرگراف تکراری یافته شده با یک گره در گراف اصلی جایگزین می‌شود و در مرحله‌ی بعد فرآیند کاوش بر روی گراف جدیدی که از جایگزینی حاصل شده‌است ادامه می‌یابد. Subdue [۱۵] اولین الگوریتم از این گروه است که برای تولید کاندید از این ایده استفاده می‌کند که هرچه تعداد تکرار زیرگراف بالاتر باشد، در صورت جایگزینی آن با یک گره گراف فشرده‌تری حاصل خواهد شد. این الگوریتم با گره‌های تکراری شروع می‌شود و در هر مرحله آن‌ها را با گره‌های بیشتر گسترش می‌دهد. برای تعیین تعداد

تکرار کاندیدها از طول توصیف کلی کاندید استفاده می‌کند. طول توصیف کلی کاندید S در گراف G، قدرت فشرده‌سازی S را نشان می‌دهد و از رابطه‌ی زیر بدست می‌آید:

$$I(S) + I(G|S) \qquad (۲)$$

$I(S)$ تعداد بیت‌های مورد نیاز برای توصیفِ S است و $I(G|S)$ تعداد بیت‌های مورد نیاز برای توصیف گراف G در صورت جایگزینی S با یک گره است.کاندیدی که این مقدار را کمینه نماید یعنی قدرت فشرده‌سازی بالاتری دارد و به عنوان زیرگراف تکراری شناخته می شود. در این مرحله فرآیند کاوش متوقف می‌شود، گراف ورودی با جایگزینی زیرگراف تکراری یافته شده با یک گره، بازنویسی می‌شود و تکرار بعدی با گراف ورودی جدید آغاز می‌شود. بدین ترتیب Subdue در هر مرحله، گراف ورودی را فشرده‌تر می‌سازد و حتی ممکن است گراف ورودی را تا یک گره فشرده کند.

GBI [۴۰] نیز روش تولید کاندید مشابهی با Subdue دارد. برای جلوگیری از فشرده شدن بیش از حد گراف (تا حدود یک گره) اندازه‌ی زیرگراف‌های استخراج شده و میزان فشرده شدن گراف، با استفاده از روش‌های تجربی تعیین می‌شوند. در این الگوریتم در هر مرحله هدف یافتن بهترین جفت گره‌ای است که با جایگزینی آن، گراف فشرده‌تری حاصل شود. روند کار در شکل (۴) نشان داده شده است. این فرآیند تا زمانی که اندازه‌ی گراف به میزان تعیین شده برسد ادامه می‌یابد. GBI برخلاف Subdue اطلاعات موجود در هر مرحله را ذخیره می‌کند و می‌تواند در هر مرحله از جستجو گراف اصلی را بازیابی نماید.



شکل (۴) روش کار الگوریتم GBI [۴۰]

الگوریتم Subdue در هر مرحله یک ساختار تکراری را می‌یابد. GBI نیز در هر مرحله تنها یک یال تکراری را مورد بررسی قرار می‌دهد این ویژگی آنها را در پایگاه‌داده‌های متراکم ناکارامد می‌سازد. زیرا در پایگاه‌داده‌های متراکم تعداد یال‌های ورودی به گره‌ها بسیار زیاد است[۱۱] و در هر مرحله رسیدگی تنها به یک کاندید، بسیار نامناسب خواهد بود. علاوه‌براین، اگر پایگاه داده شامل زیرگراف کوچک با تکرار بالا باشد، این دو الگوریتم با یافتن آن متوقف می‌شوند و تکراری بودن زیرگراف‌های بزرگتر را مورد رسیدگی قرار نمی‌دهند. GREW [۲۳] از دیگر الگوریتم‌های این گروه است که بهبودهایی را برای مشکلات دو الگوریتمِ قبل بکار می‌گیرد. این الگوریتم در هر لحظه می‌تواند چندین یال تکراری را جستجو و جایگزین کند. برای افزایش تنوع زیرگراف‌ها و محدود نشدن به زیرگراف‌های تکراری با تعداد بالا نیز از فرآیندهای تصادفی برای انتخاب جفت گره-ها استفاده می‌کند.

۴.۳. گسترش

اگر فضای جستجوی زیرگراف‌ها به صورت درختی که شامل تمامی زیرگراف‌های پایگاه داده است، در نظر گرفته شود. بطوریکه سطح اول شامل هیچ زیرگرافی نباشد، سطح دوم شامل زیرگراف‌هایی با یک یال و به همین ترتیب سطح k شامل تمام زیرگراف‌ها با k+۱ یال باشد، راهکار های مبتنی بر پیوند برای کاوش زیرگراف‌های تکراری در این درخت، مجبور به استفاده از جستجوی اول سطح[۵] هستند. زیرا این الگوریتم ها، قبل از کاوش زیرگراف‌هایی با اندازه (k+۱) زیرگراف‌هایی با اندازه k را به طور کامل کاوش می نمایند. اما راهکارهای گسترش در استفاده از روش جستجو انعطاف پذیر هستند و علاوه بر جستجوی اول سطح می توانند از جستجوی اول عمق[۶] نیز استفاده کنند. راهکارهایی که برای تولید کاندید از جستجوی اول عمق استفاده می‌کنند، در برخی از پژوهشات گراف‌کاوی به عنوان راهکارهای بدون تولید کاندید معرفی می‌شوند[۱۲،۱۴]. زیرا این راهکارها به جای تولید تمامی کاندیدهایی با اندازه‌ی k و سپس بررسی تکرار آنها، در هر مرحله یک کاندید تولید و بطور همزمان تکرار آن را بررسی می‌کنند.

در این راهکار در هر مرحله زیر گراف تکراری با اضافه کردن یال در هر موقعیت ممکن از زیر گراف گسترش می یابد. این مساله باعث ایجاد کاندیدهای مشابه زیادی خواهد شد. الگوریتم های این گروه، روش‌های متفاوتی را برای جلوگیری از ایجاد کاندیدهای افزونه به کار می گیرند. از

آنجایی که الگوریتم‌های موجود در این گروه، به مراتب از الگوریتم‌های گروه‌های پیش بیشتر هستند[۲،۱۱،۲۶،۳۵،۳۶] ، تنها تعدادی از آنها در این بخش بررسی شده‌اند.

Mofa [۲] اولین الگوریتم از این سری است که برای پایگاه‌داده‌های مولکولی بوجود آمده است ، اما می توان آن را برای گراف های دلخواه نیز بکاربرد. این الگوریتم برای کاهش کاندیدهای افزونه از سه نوع هرس مختلف استفاده می‌کند. این هرس‌ها عبارتند از: هرس مبتنی بر اندازه، هرس مبتنی بر پشتیبان و هرس ساختاری. در هرس مبتنی بر اندازه، شاخه‌هایی از درخت که به زیرگراف‌هایی منتهی می‌شوند که تعداد یال‌ها و گره-هایشان از حدآستانه‌ی از پیش تعیین شده بیشتر است هرس می‌شوند. هرس مبتنی بر پشتیبان نیز از خاصیت معیار پشتیبان برای هرس استفاده می‌کند. آخرین و مهمترین نوع هرس در این الگوریتم، هرس ساختاری است که با استفاده از شماره‌گذاری محلی امکان پذیر می‌شود. بدین‌ترتیب که گره‌های موجود در زیرگراف به ترتیب اضافه شدنشان به زیرگراف، شماره‌گذاری می‌شوند. هنگامی که یالی به گره‌ی n در زیرگراف G_S اضافه شد، یال‌های بعدی تنها می‌توانند به گره‌ی n یا گره‌های بزرگتر از آن اضافه شوند. اگر به گره‌ی n چندین یال به طور همزمان اضافه شوند، یال‌های موجود بر اساس برچسبی که دارند و برچسب گره‌های موجود در انتهایشان به صورت صعودی مرتب می‌شوند. اگرچه این روش شماره گذاری تا حد زیادی از تولید کاندیدهای افزونه جلوگیری می‌نماید، اما الگوریتم هنوز کاندیدهای افزونه زیادی را تولید می کند و سپس از آزمایش همریختی برای هرس افزونگی استفاده می کند.

gSpan [۳۶] از یک برچسب استاندارد برای گراف به نام کد DFS استفاده می کند. پیمایش DFS گراف، ترتیبی است که در آن گره ها دیده می شوند. با الحاق ارائه یال ها در این ترتیب، کد DFS نتیجه خواهد شد. تولید کاندید در این الگوریتم به دو روش محدود می شود. در ابتدا، زیر گراف ها می توانند تنها در گره هایی تمدید پیدا کنند که در راست‌ترین مسیر درخت DFS واقع شده اند. دوم gSpan برای هر زیرگراف لیستی از گراف‌های پایگاه‌داده را که زیرگراف مربوط در آن وجود دارد ذخیره می کند. بنابراین هنگام گسترش زیرگراف، به جای کاوش کل پایگاه تنها گراف های موجود در این لیست کاوش می شوند. از آنجایی که این دو قانون هرس نمی توانند به طور کامل از تولید کاندیدهای افزونه جلوگیری نمایند gSpan برای هر زیرگراف یک کد DFS استاندارد محاسبه می کند و تنها بخش هایی با کد استاندارد کمینه را گسترش می دهد و باقی را حذف می نماید.
SEUS [۱۱] برای هرس سریع کاندیدهای غیرتکراری، از ساختار داده‌ای با نام مختصر استفاده می‌کند. این ساختار با قراردادن تمام گره‌های گراف ورودی که برچسب مشابهی دارند در یک گره بدست می‌آید و نمایش فشرده‌ای از گراف ورودی را ایجاد می‌کند. این ساختار داده تنها زمانی مناسب است که گراف ورودی شامل تعداد نسبتا کمی از زیرگراف‌های تکراری با تکرار بالا باشد و در مواردی که پایگاه‌داده شامل تعداد زیادی زیرگراف تکراری با تکرار کم است مناسب نیست.

۵.۳. راهکارهای ترکیبی

آخرین نوع الگوریتم‌ها، الگوریتم‌هایی هستند که از ایده‌های راهکارهای قبلی که تا پیش از این معرفی شدند، به صورت ترکیبی استفاده می-کنند. درواقع می‌توان آنها را راهکارهای هوشمندانه‌ای دانست که از مزایای روش‌های تولید کاندید مختلف، در جهت رفع چالش‌های موجود بهره می‌برند. FFSM [۱۶] اولین الگوریتم از این گروه می‌باشد که برای تولید کاندید از روشی مرکب از الحاق و گسترش استفاده می‌کند. الحاق دو زیرگراف با هسته‌ی مشترک، می‌تواند منجر به تولید چندین کاندید مشابه شود. علاوه بر این یک کاندید ممکن است توسط چند عمل الحاق ایجاد شود. این مساله راهکار الحاق را در پایگاه‌داده‌های بزرگ ناکارامد می‌سازد. در راهکار گسترش نیز گسترش تنها به گره‌هایی که یال جدید به آنها ختم می‌شود، محدود می‌شود. بنابراین الگوریتم به زمان زیادی برای رسیدگی به تمامی گره‌ها نیازمند است. FFSM با استفاده از ساختار متفاوتی برای نمایش گراف‌ها و بکارگیری روشی مرکب از الحاق و گسترش در جهت رفع این چالش‌ها عمل می‌کند.

۴. ارزیابی الگوریتم‌های کاوش زیرگراف تکراری

با توجه به تعداد زیاد الگوریتم‌های موجود برای کاوش زیرگراف تکراری، انتخاب الگوریتم مناسب بسیار چالش‌برانگیز خواهد بود. از آنجایی که ادعاهای الگوریتم‌های پیشنهادی تنها بر پایگاه‌داده‌های خاصی که روی آن پیاده‌سازی شده‌اند است، نمی‌توان موفقیت آنها را در پایگاه‌داده‌های دیگر تضمین کرد و ارزیابی مطلق الگوریتم‌ها امکان‌پذیر نیست. اما هنوز راهکارهایی برای ارزیابی الگوریتم‌ها و انتخاب الگوریتم مناسب وجود دارد که از جمله‌ی آنها می‌توان سوالات زیر را نام برد :

- چه نوع گرافی مورد کاوش قرار می‌گیرد ؟
- آیا استفاده از دانش پیش‌زمینه در حین کاوش اهمیت دارد؟
- آیا تقریبی یا دقیق بودن نتیجه اهمیت دارد؟
- میزان حافظه‌ی در دسترس چقدر است ؟

- آیا کاربر در حین کاوش لازم است فرآیند کاوش را مدیریت کند؟
- گم شدن زیرگراف‌ها تا چه اندازه اهمیت دارد ؟

سوالات بالا نمونه سوالاتی هستند که می‌توانند در انتخاب الگوریتم موثر مناسب باشند. برای مثال کاربردهایی که با کمبود حافظه مواجه هستند بهتر است از الگوریتم‌های تولید کاندیدی با جستجوی اول عمق استفاده کنند. یا در مواردی که سرعت بیشتر از دقت اهمیت دارد، بهتر است برای بالابردن سرعت از روش تعیین همریختی تقریبی استفاده شود. با پاسخ به این سوالات و سوالات مشابه، به راحتی می‌توان با استفاده از مشخصات الگوریتم‌ها که در جدول (۱) نشان داده شده اند الگوریتم مناسب را انتخاب کرد. علاوه بر آن، جدول (۱) مشخصات پایگاه‌داده‌هایی که الگوریتم‌های مربوطه در آنها بدون تردید نتایج خوبی خواهند داشت را نیز نشان می‌دهد.

جدول (۱) مشخصات بارز الگوریتم‌های کاوش زیرگرافِ تکراری

الگوریتم‌های کاوش زیرگراف تکرارشونده	معیارهای مقایسه									
	تاریخ ظهور	محدودیتِ بر نوع همبندی پایگاه‌داده‌ی ورودی	نوع زیرگراف-های استخراج-شده	استفاده از دانش پیش-زمینه	تعیین همریختی	روش جستجو	تعامل با کاربر	کامل بودن جستجو	نحوه‌ی عملکرد در پایگاه متراکم	زمینه‌ی کاربردی موثر
SUBDUE	۱۹۹۴	برچسب‌دار	همبند	بله	تقریبی	حریصانه	خیر	خیر	بد	پایگاه‌داده بدون زیرگراف-های کوچک با تکرار بالا
GBI	۱۹۹۴	محدودیت ندارد	همبند	خیر	دقیق	حریصانه	خیر	خیر	بد	پایگاه‌داده بدون زیرگراف-های کوچک با تکرار بالا / پایگاه‌داده با تعداد زیادی گره با برچسب یکسان
Grew	۲۰۰۴	برچسب‌دار، بدون جهت	همبند	خیر	دقیق	حریصانه	خیر	خیر	خوب	-
WARMR	۱۹۹۸	محدودیت ندارد	همبند	بله	دقیق	اول سطح	خیر	وابسته به دانش پیش-زمینه	بد	اثبات مفهومی یک چهارچوب
FARMAR	۱۹۹۸	محدودیت ندارد	همبند	بله	تقریبی	اول سطح	خیر	وابسته به دانش پیش-زمینه	بد	اثبات مفهومی یک چهارچوب
AGM	۲۰۰۰	محدودیت ندارد	القایی	خیر	دقیق	اول سطح	خیر	بله	بد	-
FSG	۲۰۰۱	بدون جهت	همبند	خیر	دقیق	اول سطح	خیر	بله	بد	پایگاه‌داده با برچسب یال‌ها و گره‌های متنوع
HSIGRAM	۲۰۰۴	برچسب‌دار، بدون جهت	همبند	خیر	قابل تنظیم	اول سطح	خیر	بله	بد	گراف کم‌پشت با مقیاس بزرگ
PATH	۲۰۰۶	محدودیت ندارد	همبند	خیر	دقیق	اول سطح	خیر	بله	بد	داده‌کاوی روی بخش قابل رشد اسناد Web
Mofa	۲۰۰۲	برچسب‌دار، بدون جهت	همبند	خیر	دقیق	اول عمق	خیر	بله	بد	پایگاه‌های مولکولی
gSpan	۲۰۰۲	برچسب‌دار، بدون جهت	همبند	خیر	دقیق	اول عمق	خیر	بله	بد	-
Gaston	۲۰۰۴	برچسب‌دار، بدون جهت	همبند	خیر	دقیق	اول عمق	خیر	بله	بد	-
FASM	۲۰۰۸	برچسب‌دار	همبند	خیر	دقیق	اول سطح	خیر	بله	خوب	-
FFSM	۲۰۰۳	برچسب‌دار، بدون جهت	همبند	خیر	دقیق	اول عمق	خیر	بله	بد	-
SEUS	۲۰۰۶	برچسب‌دار، جهت‌دار	همبند	خیر	دقیق	اول عمق	بله	قابل تنظیم	بد	کاوش زیرگراف تکراری بدون تعیین تکرار

۵.نتیجه‌گیری

در این مقاله الگوریتم‌های کاوش زیرگراف تکراری مورد بررسی قرار گرفتند. در ابتدا این الگوریتم‌ها بر اساس نوع پایگاه‌داده‌ای که روی آن کار می-کنند و روشی که برای تولید کاندید بکار می‌گیرند دسته‌بندی شدند و سپس به منظورکمک به کاربردها در انتخاب الگوریتم مناسبشان، مشخصات بارز آنها در قالب جدولی ارائه شد. با توجه به نتایج، از جمله چالش‌های موجود در این زمینه می‌توان بد عمل کردن اکثر الگوریتم‌ها در پایگاه‌های متراکم، عدم تعامل با کاربر، عدم ارائه‌ی نتایج میانی و پیچیدگی‌های زمانی و حافظه‌ای را نام برد. از ارزیابی ارائه شده می‌توان برای ایجاد راهکارهای ترکیبی ای که مزایای حاصل از راهکارهای مختلف تولید کاندید را در جهت رفع هر چه بیشتر چالش‌ها ترکیب نمایند، استفاده کرد. اضافه کردن الگوریتم‌ها و راهکارهای بیشتر به دسته‌بندی فوق ازدیگرجهت‌های پژوهشی آینده است. مقایسه و دسته‌بندی الگوریتم‌های کاوش زیرگراف تکراری بر اساس نوع روشی که برای تعیین همریختی بکار می‌برند نیز از جمله فضاهای پژوهشی است که کمتر مورد توجه قرار گرفته است و می‌تواند در تدوین الگوریتم‌های کارا در این زمینه موثر باشد.

۶.مراجع

[۱] Agrawal.R and Srikant.R, "*Fast Algorithms For Mining Association Rules*", In Proceedings of the ۲۰th Very Large Dada Base Conference(VLDB'۹٤), pp.٤٨٧-٤٩٩, Santiago, ۱۹۹٤.

[۲] Borgelt.C, Berthold.MR, "*Mining Molecular Fragments: Finding Relevant Substructures of Molecules*",In Proceeding of the international conference on data mining (ICDM'۰۲), Japan, pp. ۲۱۱–۲۱۸, ۲۰۰۲.

[۳] Chakrabarti.D and Faloutsos.C, "*Graph Mining: Laws, Generators, and Algorithms*", ACM Computing Surveys, New York, pp.۲-٦۹, ۲۰۰٦.

[٤] Cook.J and Holder.L, "*Substructure Discovery Using Minimum Description Length and Background Knowledge*", Journal of Artificial Intelligence Research, pp. ۲۳۱-۲٥٥, ۱۹۹٤.

[٥] Damiani.E , Oliboni.B, Quintarelli.E and Tanca.L, "*Modeling Semistructured Data by Using Graph-Based Constraints*", In Proceedings of OTM Workshops, pp.۲۲-۲۳, ۲۰۰۳.

[٦] Dehaspe.L and Toivonen.H, "*Discovery of Frequent Datalog Patterns*", Data Mining and Knowledge Discovery, pp.۷-۳٦, ۱۹۹۹.

[۷] Deshpande.M , Kuramochi.M and Karypis.G , "*Frequent sub-structure-based approaches for classifying chemical compounds*",In Proceedings of the international conference on data mining (ICDM'۰۳), pp. ۳٥–٤۲,۲۰۰۳.

[۸] Doulamis.AD, Doulamis.ND and Kollias.ND, "*A Pyramidal Graph Representation for Efficient Image Content Description*", IEEE International Workshop on Multimedia Signal Processing (MMSP), Denmark, pp.۱۰۹-۱۱٤,۱۹۹۹.

[۹] Fortin.S," *The graph isomorphism problem*" ,Technical Report TR۹٦-۲۰, Department of Computing Science, University of Alberta, ۱۹۹٦.

[۱۰] Garey, M. R and Johnson D. S, " *Computers and Intractability:A Guide to the Theory of NP-Completeness*" , W. H.Freemanand Company, New York, ۱۹۷۹.

[۱۱] Gudes.E, Shimony.E and Vanetik.N, "*Discovering Frequent Graph Patterns Using Disjoint Paths*", IEEE Transactions on Knowledge and Data Engineering , Los Angeles, pp.۱٤٤۱–۱٤٥٦, ۲۰۰٦.

[۱۲] Han.J, Cheng.H, Xin.D and Yan.X, "*Frequent Pattern Mining: Current Status and Future Directions*", Data Mining and Knowledge Discovery (DMKD'۰۷), ۱۰th Anniversary Issue, pp.٥٥–۸٦, ۲۰۰۷.

[۱۳] Han.J and Kamber.M, *Data Mining: Concepts and Techniques*, Second edition:Morgan Kaufmann, ۲۰۰٥.

[۱٤] Han.J, Pei.J and Yin,Y, " *Mining Frequent Patterns without Candidate Generation*", In Proceedings of the ACM-SIGMOD International Conference on Management of Data, Texas, pp.۱–۱۲, ۲۰۰۰.

[۱٥] Holder.LB, Cook.DJ and Djoko.S, "*Substructure Discovery in the Subdue System*", In Proceeding of the AAAI'۹٤ workshop knowledge discovery in databases (KDD'۹٤), WA, pp ۱٦۹–۱۸۰, ۱۹۹٤.

[۱٦] Huan.J, Wang.W, Prins.J,"*Efficient mining of frequent subgraph in the presence of isomorphism*", In Proceeding of the international conference on data mining (ICDM'۰۳),Melbourne, pp.٥٤۹–٥٥۲, ۲۰۰۳.

[۱۷] Inokuchi.A, Washio.T and Motoda.H,"*An Apriori-Based Algorithm for Mining Frequent Substructures from Graph Data*", In Proceedings of the ٤th European Conference on Principles and Practice of Data Mining and Knowledge Discovery*(PKDD)*, pp.۱۳–۲۳, ۲۰۰۰.

[۱۸] Inokuchi.A, Washio.T and Motoda.H ,"*Complete Mining of Frequent Patterns From Graphs: Mining graph data*", *MachineLearning*, pp.۳۲۱–۳٥٤, ۲۰۰۳.

[۱۹] Inokuchi.A, Washio.T, Nishimura.Y and Motoda.H, "*General Framework For Mining Frequent Patterns in Structures*", In Proceedings of the ICDM workshop on Active Mining (AM), Netherlands, pp.۲۳–۳۰, ۲۰۰۲.

[۲۰] Kerber.A, Laue.R, Meringer,M and Rücker.C, *"Molecules in Silico: A Graph Description of Chemical Reactions"*, Journal of Chemical Information and Modeling, pp.۸۰٥-۸۱۷, ۲۰۰۷.

[۲۱] Kuramochi.M and Karypis.G, "*Finding Frequent Patterns In a Large Sparse Graph*", in Proceedings of the ٤th SIAM International Conference on Data Mining (SDM ۲۰۰٤), USA, ۲۰۰٤.

[۲۲] Kuramochi.M, Karypis.G, "*Frequent Subgraph Discovery*", In Proceedings of the international conference on data mining (ICDM'۰۱), California, pp.۳۱۳–۳۲۰, ۲۰۰۱.

[۲۳] Kuramochi.M and Karypis.G ," *GREW: A Scalable Frequent Subgraph Discovery Algorithm*", In Proceeding of the international conference on data mining (ICDM'۰٤), Brighton, pp.٤۳۹–٤٤۲,۲۰۰٤.

Meinl.T, Borgelt.C and Berthold.MR," *Discriminative Closed Fragment Mining and Perfect Extensions in MoFa*", In Proceedings of the ٢th Starting AI Researchers' Symposium (STAIRS), pp.٣-١٤, Spain, ٢٠٠٤. [٢٤]

Muggleton.S and De Raedt.L, "*Inductive Logic Programming: Theory and Methods*", Journal of Logic Programming, pp.٦٢٩-٦٧٩, ١٩٩٤. [٢٥]

Nijssen.S, Kok.J, "*A Quickstart in Frequent Structure Mining Can Make a Difference*", In Proceeding of the ACM SIGKDD international conference on kowledge discovery in databases (KDD'٠٤), Washington, pp.٦٤٧–٦٥٢, ٢٠٠٤. [٢٦]

Nijssen.S and Kok.J, "*Faster Association Rules For Multiple Relations*", In Proceeding of the ١٧th International Joint Conference on Artificial Intelligence, pp.٨٩١-٨٩٦, ٢٠٠١. [٢٧]

Peng.JA, Yang LM, Wang.JX and Liu.Z; Li Ming, "*An Efficient Algorithm for Detecting Closed Frequent Subgraphs in Biological Networks*", in Proceedings of the International Conference on BioMedical Engineering and Informatics, USA, pp.٦٧٧-٦٨١, ٢٠٠٨. [٢٨]

Samudrala.R and Moult.J," *A graph-theoretic algorithm for comparative modeling of protein structure* ", Journal of Molecular Biology, USA, pp.٢٨٧-٣٠٢,١٩٩٨. [٢٩]

Schenker.A, Last.M, Bunke.H and Kandel.A, "*Classification of Web Documents Using a Graph Model*", ". In Proceedings of the ٧th International Conference on Document Analysis and Recognition (ICDAR'٠٣), pp.٢٣٣-٢٥١, ٢٠٠٣. [٣٠]

Shahriar Hossain.M and Angryk. R. A, "*GDClust: A Graph-Based Document Clustering Technique*", In Proceedings of the ٧th IEEE International Conference on Data Mining, pp.٤١٧-٤٢٢, ٢٠٠٧. [٣١]

Washio.T and Motoda.H, "*State of the Art of Graph-Based Data Mining*", ACM SIGKDD Explorations Newsletter, NewYork , pp. ٥٩–٦٨, ٢٠٠٣. [٣٢]

Vanetik.N, Gudes.E, Shimony.SE, "*Computing Frequent Graph Patterns From Semistructured Data*", In Proceedings of the international conference on data mining (ICDM'٠٢), Japan, pp.٤٥٨–٤٦٥,٢٠٠٢. [٣٣]

Worlein.M, Meinl.T, Fischer.I and Philippsen.M, "*A Quantitative Comparison of the Subgraph Miners MoFa, gSpan, FFSM and Gaston*", In Proceedings of the ٩th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD'٠٥), Portugal, pp.٣٩٢–٤٠٣, ٢٠٠٥. [٣٤]

Wu.J and Chen.L, "*Mining Frequent Subgraph by Incidence Matrix Normalization*", Journal of Computers, pp.١٠٩–١١٥, ٢٠٠٨. [٣٥]

Yan.X, Han.J, " *GSpan: Graph-Based Substructure Pattern Mining*", In Proceeding of the international conference on data mining (ICDM'٠٢), Japan, pp.٧٢١–٧٢٤, ٢٠٠٢. [٣٦]

Yan.X, Yu.PS and Han.J, "*Graph Indexing: a Frequent Structure-Based Approach*", In Proceedings of the international conference on management of data (SIGMOD'٠٤), Chicago, pp.٣٣٥-٣٤٦, ٢٠٠٤. [٣٧]

Yan.X, Yu.PS and Han.J, "*Searching Substructures With Superimposed Distance*", In Proceedings of the International conference on data engineering (ICDE'٠٦), PP.٨٨٨-٩٨٩, ٢٠٠٦. [٣٨]

Yan.X , Yu.PS and Han.J, "*Substructure Similarity Search in Graph Databases*", In Proceedings of the international conference on management of data (SIGMOD'٠٥), pp.٧٦٦-٧٧٧, ٢٠٠٥. [٣٩]

Yoshida.K, Motoda.H and Indurkhya,N, "*Graph-based Induction as a Unified Learning Framework*", Journal of Applied Intelligence, pp.٢٩٧–٣٢٨. [٤٠]

Zhdanova, A.V., Predoiu, L., Pellegrini, T., Fensel, D. "*A Social Networking Model of a Web Community*". In Proceedings of the ١٠th International Symposium on Social Communication , Cuba, pp. ٥٣٧-٥٤١, ٢٠٠٧. [٤١]

---

١ Subgraph isomorphism
٢ Support
٣ Automorphism
٤ Inductive logic programming
٥ Breadth first search
٦ Depth first search