استخراج گراف

۱. مقدمه

محتوای کتاب تا اینجا بر استخراج داده ها تمرکز کرده در حالی که ساختار اصلی به وسیله انواع ویژه ای از گراف ها که وجود حلقه (مانند گراف ها یا درخت های حلقه ای) در آنها مجاز نیست، مشخص می شود. تمرکز این فصل بر مسئله استخراج الگوهای پر تکرار است در حالی که ساختار اصلی داده ها می تواند به گراف کلی که وجود حلقه نیز در آن مجاز است، تعلق داشته باشد. اینگونه تصاویر به فرد اجازه می دهد ابعاد پیچیده و دشوار این حوزه مانند ترکیب های شیمیایی، شبکه ها، وب، بیوانفورماتیک و ... را طراحی و مدل سازی کند. به طور کلی، گراف ها با توجه به پیچیدگی های الگوریتمی دارای ویژگی های تئوریک (نظری) نامطلوب فراوانی هستند. در مسئله استخراج گراف شمارش و بررسی دقیق زیر گراف های یک گراف خاص است که با عنوان مسئله استخراج زیر گراف های پرتکرار شناخته می شود. برأساس روش تحلیل گراف موجود، تمرکز خود را بر مسئله فوق که لازمه بررسی پیوند های جالب در میان داده های دارای ساختار گراف است و کاربردهای بسیاری دارد، محدود می کنیم . برای بررسی و ارزیابی همه جانبه استخراج گراف در چارچوب کلی از جمله قوانین مختلف، مولد ها و الگوریتم های داده ها و ... لطفا به (Chakrabati & Faloutsos 2006; Washio & Motoda 2003, Han & Kamber 2006) مراجعه کنید. با توجه به وجود حلقه ها در گراف، مسئله استخراج گراف های پرتکرار بسیار پیچیده تر و دشوارتر از مسئله استخراج زیر مجموعه های درختی است. با وجود اینکه به لحاظ مسئله استخراج گراف یک مسئله NP-کامل است، در عمل تعدادی از رویکردها برای تحلیل داده های گراف حقیقی قابل اجرا است. در ادامه تعدادی از این رویکرد ها و نگرش های مختلف بر مسئله استخراج زیر گراف های پر تکرار و بعضی از رویکردها برای تحلیل کلی داده های گراف را بررسی خواهیم کرد.

بقیه فصل نیز بصورت زیر سازماندهی شده است. مفاهیم ضروری مرتبط با مسئله استخراج گراف در بخش ۲ مورد بررسی و بحث قرار می گیرد. مسئله هم ریختی و هم شکلی گراف ها که حوزه بسیار مهمی از فرایند استخراج زیر گراف های پرتکرار است در بخش ۳ مورد بحث قرار می گیرد. در بخش ۴، یک نمای کلی از برخی روش های موجود برای استخراج گراف که بر طبق رویکرد و تلقی اصلی از این مسئله طبقه بندی شده اند، ارائه می شود. نتیجه گیری این فصل در بخش ۵ آورده شده است.

۲. مفاهیم و تعاریف کلی گراف

گراف مجموعه ای از گره هاست که هر یک از این گره ها را میتوان به گره ای دیگر و حتی خود گره متصل کرد. انواع زیادی گراف وجود دارد از جمله: مستقیم / غیر مستقیم، وزنه ای، متناهی / نامتناهی، منظم (متقارن)

و گراف های کامل. این نوع از گراف ها اغلب برای کاربردهای ویژه و تخصصی که در آن رابطه ها یا الزامات خاصی حفظ می شود یا حفظ آنها تقویت و تایید می شود، ایجاد شده اند. با این وجود، اکثر مدارک و اسناد نیمه-سازمان یافته ای که ساختار اصلی آنها نوعی گراف است را می توان بصورت یک گراف با کران های غیر مستقیم و بدون برچسب مدل سازی کرد. از این نظر، گراف را می توان بصورت G = (V, L, E) تعریف کرد، که (۱) V مجموعه رأس ها یا گره ها است; (۲) L یک تابع برچسب زن است که به هر رأس $v \in V$ یک برچسب L (v) نصب می کند; و (۳) $E = \{(v1, v2) \mid v1, v2 \in V\}$ مجموعه ای از کران ها در مجموعه G است. تعداد گره ها در G به عنوان مرتبه G نامیده می شود در حالی که تعداد کران ها (|E|) v به عنوان اندازه G معرفی می شود. هر کران معمولا دو گره به همراه دارد ولی ممکن است کران فقط دارای یک گره باشد در حالتی که یک کران از یکی از گره ها به طرف خودش کشیده شود (مانند یک حلقه). اگر دو رأس v1 , v2 به یک کران متصل شده باشند؛ انگاه گفته می شود که آنها مجانب یکدیگرند، و در غیر اینصورت این دو رأس غیر مجانب یا مستقل خوانده می شوند. دو کران که در یک رأس به هم می رسند مجانب یکدیگر هستند و در غیر اینصورت غیر مجانب اند. به عبارتی دیگر، اگر دو کران (V_{a1}, V_{a2}) و (V_{b1}, V_{b2}) مجانب یکدیگر باشند و $(V_{a1} \neq V_{a2})$ و $(V_{a1} \neq V_{a2})$ ، آنگاه یکی از عبارت های دیگر صدق می کند:

$(V_{a1} = V_{b1})$ یا $(V_{a1} = V_{b2})$ یا $(V_{a2} = V_{b1})$ یا $(V_{a2} = V_{b2})$. مجموعه همه رأس های مجانب رأس V با مجانب های V متناظر است. یک مسیر بصورت توالی متناهی از کران ها بین هر دو گره تعریف می شود و در گراف بر خلاف گراف درختی اگر یک مسیر خاص بین دو گره وجود داشته باشد، ممکن است مسیرهای چندگانه ای وجود داشته باشد، طول مسیر P به تعداد کران ها در p گفته می شود. مرتبه یک گره به تعداد گره های صادر شده از یک گره گفته می شود.

گراف G’ زیر گراف G خواهد بود، اگر هم ریختی زیر گراف بین G’ و G وجود داشته باشد. هم ریختی عبارت است از متناظر یک به یک مجموعه گره های یک گراف با مجموعه گره های گراف دیگر در شرایطی که مجانب یا غیر مجانب بودن محفوظ باشد.

در مورد گراف های درختی، تعدادی از زیر گراف های مختلف وجود دارند و بعضی از متداول ترین نمونه ها در چارچوب استخراج زیر گراف پرتکرار همراه با فرمول مسئله هم ریختی زیر گراف ها به ترتیب مورد بحث قرار گرفته است.

۳. مسئله هم ریختی گراف

دو گراف $G_1(V_1, L_1, E_1)$ و $G_2(V_2, L_2, E_2)$ هم ریخت یکدیگر گفته می شوند اگر رابطه $F : V_1 \rightarrow V_2$ صدق کند به صورتی که $(V_1, V_2) \in E_1$ اگر $\big(f(V_1), f(V_2)\big) \in E_2$. بنابراین، دو گراف برچسب دار

$G_1(V_1, L_1, E_1)$ و $G_2(V_2, L_2, E_2)$ هم ریخت یکدیگرند اگر متناظر یک به یک از V_1 به V_2 وجود داشته باشد به صورتی که رأس ها، برچسب رأس ها و مجانب یا غیر مجانب بودن رأس ها حفظ شود.

به طوری که پیشتر گفته شد، هر گراف می تواند زیر گراف یک گراف دیگر باشد اگر هم ریختی زیر گراف بین آن دو صدق کند. به طور صریح تر، این مسئله را می توان به صورت زیر بیان کرد:

گراف $G_s(V_s, L_s, E_s)$ زیر گراف $G(V, L, E)$ است اگر $V_s \subseteq V$ و $E_s \subseteq E$.

مسئله استخراج زیر گراف پر تکرار را می توان بطور کلی به این صورت بیان کرد: با ارائه پایگاه داده های G_{DB} و آستانه حفاظت مینیمم (σ)، می توان همه زیر گراف هایی که دست کم σ بار در G_{DB} وجود دارند را استخراج کرد.

علاوه بر تعریف کلی زیر گراف که در بخش های قبلی ارائه شده سایر زیر گراف های متداول عبارتند از:

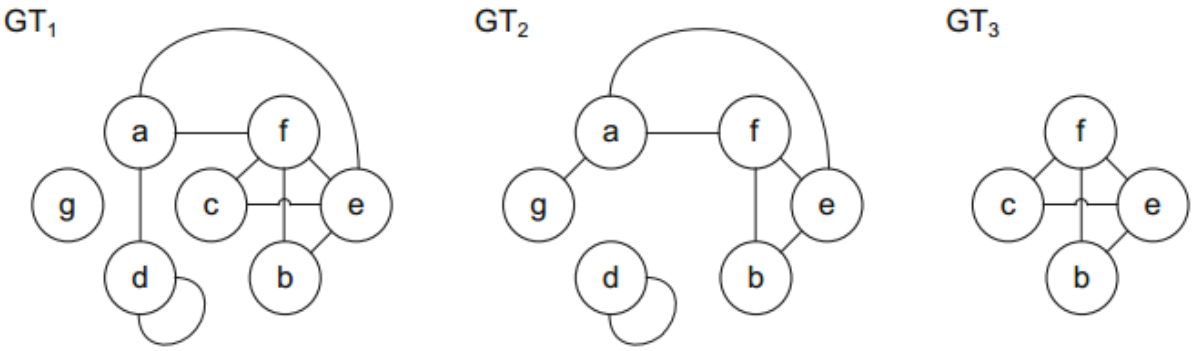
دربرگیرنده، پیوسته و القا شده.

گراف $G_s(V_s, L_s, E_s)$ زیر گراف در بر گیرنده گراف $G(V, L, E)$ است اگر و تنها اگر $V_s = V$ و $E_s \subseteq E$.

گراف $G_s(V_s, L_s, E_s)$ زیر گراف پیوسته گراف $G(V, L, E)$ است اگر $V_s \subseteq V$ و $E_s \subseteq E$ و تمام رأس های V_s به طور متقابل از طریق کران های E_s قابل دستیابی باشند.

گراف $G_s(V_s, L_s, E_s)$ زیر گراف القا شده گراف $G(V, L, E)$ است. اگر $V_s \subseteq V$ و $E_s \subseteq E$ و تناظر $F : V_1 \rightarrow V_2$ صدق کند به طوری که برای هر جفت رأس $V_x, V_y \in V_s$. اگر کران $E \in \big(f(V_x), f(V_y)\big)$ وجود داشته باشد آنگاه $(V_x, V_y) \in E_s$.

برای روشن شدن این ابعاد، لطفا به تصویر ۱ نگاه کنید که یک نمونه پایگاه داده گراف G_{DB} شامل سه فعالیت را نشان می دهد.

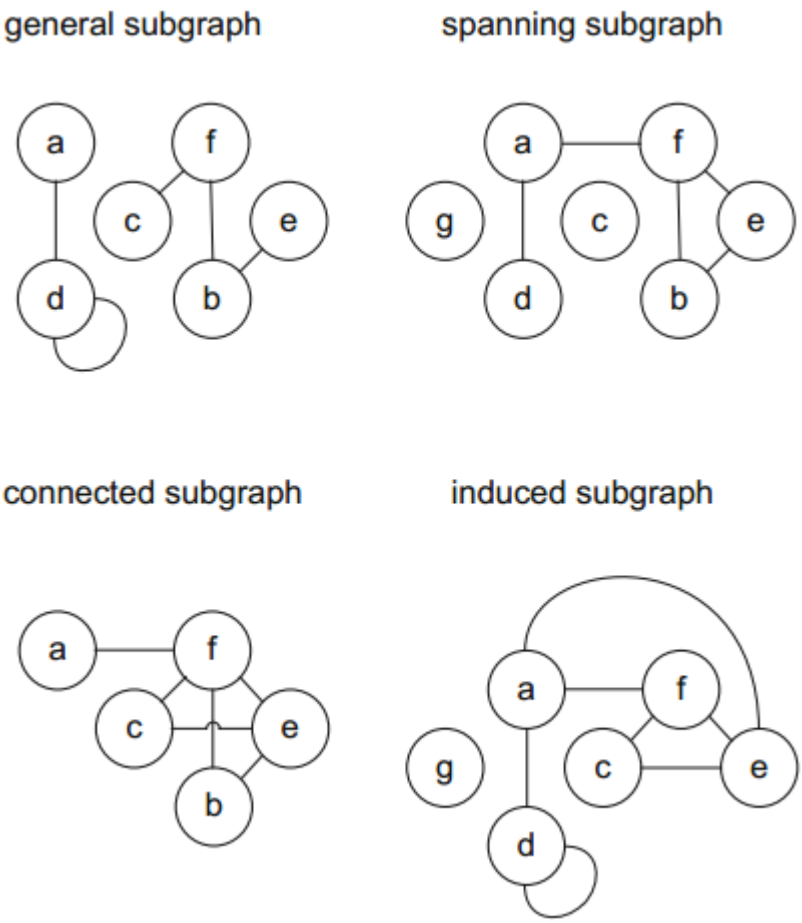


تصویر ۱ نمونه ای از پایگاه داده گراف G_{DB} شامل سه فعالیت GT_1 , GT_2 , GT_3.

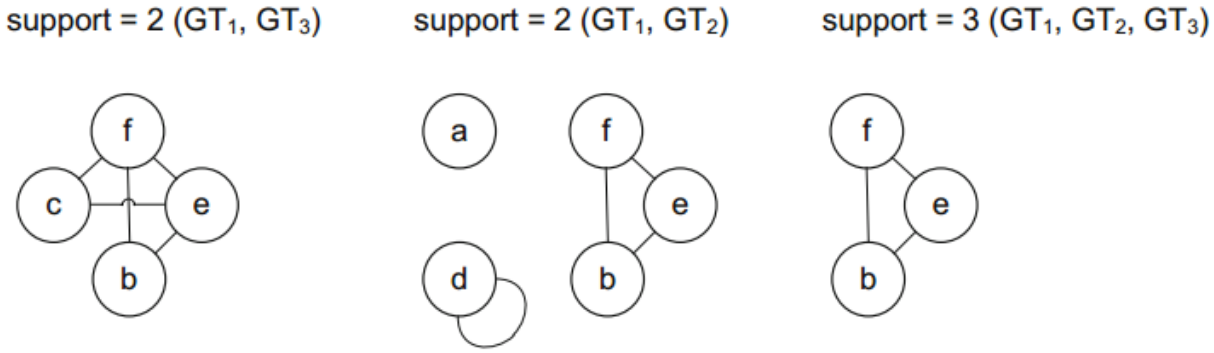
در این بخش، تمرکز خود را بر متداول ترین تعاریف زیر گراف در مسئله استخراج الگوی پر تکرار از دادهای دارای ساختار گراف در حوزه استخراج داده ها محدود می کنیم. تعداد بسیار کمی گونه ها و شکل های مختلف گراف وجود دارد که در مسئله تحقیق گسترده تحلیل داده های گراف قابل بررسی و ملاحظه هستند و همچنین انواع مختلفی از معیارها و الزامات (محدودیت ها) که می توان بر فرایند تحلیل داده ها تحمیل کرد . برای دسترسی به جزئیات بیشتر درباره این جنبه ها و سایر موضوعات مرتبط خواننده علاقه مند و پیگیر را به مشاهده (Chakrabati & Faloutsos 2006, Washio & Motoda 2003) ارجاع می دهیم. در بخش بعدی، روش های گوناگونی را که برای مسئله استخراج گراف پر تکرار ایجاد شده اند مورد بررسی قرار می دهیم. در بعضی از این روش ها الزاماتی و محدودیت هایی وضع شده است تا تحلیل را به سمت هدف کاربردی ویژه ای سوق دهد یا تعداد الگوهای شمارش شده را کاهش دهد تا پیچیدگی و دشواری حاصل از استخراج داده های گراف را کم کند.

۴. روش های موجود برای استخراج گراف

تعدادی روش های استخراج داده های گراف – محور طراحی و ایجاد شده اند. در این بخش، عمدتا بر روش هایی که برای انجام عملیات استخراج زیر گراف های پرتکرار (که در بخش ۳ گفته شد) طراحی شده اند تمرکز می کنیم. در انتهای این بخش تعدادی از روش های ایجاد شده برای رفع مسائل مرتبط با استخراج زیر گراف های استخراج و به طور کلی تحلیل داده های گراف را ارزیابی می کنیم.



تصویر ۲ نمونه هایی از زیر گراف های متعلق به گراف فعالیت GT از پایگاه داده های G_{DB}



تصویر ۳ تعدادی از زیر گراف های عمومی پر تکرار از پایگاه داده های G_{DB}

۴,۱ روش های آپریوری – مانند

این بخش بر روش های استخراج گراف که با استفاده از اصول و قواعد استخراج مجموعه آیتم / توالی / درختچه های گراف پرتکرار بر مبنای روش آپریوری تمرکز کرده است. فرایند شمارش و بررسی به شیوه سروته انجام می شود و با زیر گراف هایی که شامل (مثلا زیر گراف –۱) آغاز می شود. در هر چرخه (تکرار)، زیر گراف داوطلب K از نظر پر تکرار بودن مورد بررسی قرار می گیرند و فقط الگوهای پرتکرار برای ایجاد زیر گراف های (K+1) مورد استفاده قرار می گیرند. تعدادی از واریانس های مختلف در میان الگوریتم های آپریوری-مبنا در خصوص روش ایجاد داوطلب ها (گراف های داوطلب) وجود دارد. به طوری که پیشتر در کتاب مورد بحث قرار گرفت، رویکرد اتصال که برای استخراج مجموعه آیتم های پرتکرار به خوبی کار می کند ممکن است برای برنامه های کاربردی که ویژگی های ساختاری الگوهای داده در نظر گرفته می شوند، مناسب و کارآمد نباشد. تعداد زیادی داوطلب نامعتبر بی جهت ایجاد و فرض می شوند. در استخراج زیر گراف پرتکرار با توجه به وجود روش های بسیاری برای عملیات اتصال دو زیر ساختار پدیده فوق بیشتر مشاهده می شود.

الگوریتم AGM (ایکوچی، واشیو و موتودا ۲۰۰۰) رویکرد آپریوری سطح – محور را بر می گزینند که گراف ها با استفاده از یک ماتریکس مجانب ارائه می شوند. فرایند مورد نظر از کوچک ترین زیر گراف ها شروع می شود و داوطلب ها با اجرای عملیات اتصال برای ماتریکس های مجانب نشان دهنده زیر گراف های داوطلب بررسی و شمارش می شوند. بر اساس دسته بندی ماتریکس های مجانب که در تعیین دقیق فراوانی نقش دارند، شکل مجاز برای زیر گراف القا شده تعریف می شود. الگوریتم FSG (کوراموشی و کاریپیس ۲۰۰۱)، برای استخراج زیر گراف های مستقیم / غیر مستقیم پر تکرار (فراوان) طراحی شد. برای به حداقل رساندن فرایند ذخیره سازی و محاسبه الگوریتم های FSG برای ذخیره موثر فعالیت های ورودی، زیر گراف های داوطلب و زیر گراف های پر تکرار از نمایش گراف کم تراکم (غیر متراکم) استفاده می کند. این الگوریتم ها شکل مجاز ماتریکس مجانب را اجرا می کنند و سپس آن را به نمایش فهرست – مجانب ها تبدیل می کند. برای ایجاد زیر گراف های داوطلب (K+1) زیر گراف پر تکرار K، عملیات اتصال برای زیر گراف های پر تکرار K که شامل زیر گراف (K-1) مشابهی است اجرا می شود. فراوانی گراف (K+1) جدید به صورت اندازه مقطع فهرست های شناساگر اجرایی (فهرست TID) زیر گراف های متصل K تعریف می شود. استفاده از فهرست های TID می تواند فرایند بررسی و شمارش زیر گراف های داوطلب در FSG را ساده تر کند اما لزوم ذخیره تمام فهرست های TID برای داده های گراف بزرگ موجب ایجاد مشکلات حافظه می شود. همین نویسندگان (کوراموشی و کاریپیس ۲۰۰۲) الگوریتم gFSG را پیشنهاد داده اند که دنباله مسئله یافتن الگوهای هندسی پرتکرار در گراف های هندسی است. این گراف های هندسی در واقع گراف هایی هستند که رأس های آنها دارای مختصات دو یا سه بعدی هستند. gFSG از یک رویکرد سطح-محور استفاده می کند که در هر نوبت به وسیله یکی از کران ها، زیر گراف های پر تکرار را امتداد می دهد و تعدادی از الگوریتم ها برای ارزیابی هم ریختی زیر گراف های هندسی تلفیق (ادغام) می شوند (که عبارتند از دوران، مرتبه، واریانس برگردان).

یکی دیگر از الگوریتم های آپریوری-محور در (وانتیک، گودس و شیمونی، ۲۰۰۲) ارائه شد. این الگوریتم از نمایش های مجاز مسیرها و توالی مسیرها استفاده می کند و نظام واژگان نگاری روی جفت مسیرها بر اساس برچسب های گره و مرتبه گره های درون مسیر تعیین می شود. گراف به عنوان گروهی از مسیرهایی با کران های قطعه قطعه شده بیان می شود که دو مسیر در صورتی دارای کران های قطعه قطعه شده هستند که دارای هیچ کران مشترکی نباشند. رویکرد سروته جایی مورد استفاده قرار می گیرد که در ابتدا، تمام زیر گراف های پر تکراری که دارای مسیر مجزا هستند پیدا شوند و همه آنها ترکیب شوند و هرجا که امکان داشته باشد زیر گرافهایی دارای دو مسیر ساخته شود. الگوریتم به تدریج زیر گراف های دارای K مسیر را با اتصال زیر گراف های پرتکرار با K-1 مسیر معین می کند

۴,۲. روش های توسعه الگو

اشتراک بین الگوریتم هایی که از رویکردهای شبیه به آپریوری استفاده می کنند این است که زیر گراف ها به طور منظم به روش Bottom-Up معین می شوند در حالی که وقتی داده ها بسیار بزرگ باشند یا رابطه ساختاری نسبتا پیچیده باشد مسائل مختلفی به وجود خواهد آمد. این مورد به ویژه در شرایطی روی خواهد داد که زیر گراف های داوطلب با استفاده از رویکرد اتصال ایجاد می شوند. در شرایطی که دو زیر گراف قابل اتصال باشند احتمالات زیادی وجود خواهد داشت و شکل های ساختاری یک زیر گراف داوطلب همواره در پایگاه داده وجود نخواهد داشت. هم ریختی زیر گراف یک آزمایش پر هزینه است و به همین خاطر، ایجاد زیر گراف های داوطلبی که می بایست در ادامه از بین بروند، بیهوده است. این مشکلات، انگیزه ها برای طراحی تعدادی از الگوریتم ها که برای به حداقل رساندن زیر گراف های داوطلب غیر ضروری از رویکرد ساختار گراف هدایت شونده استفاده می کنند، را تقویت می کند.

الگوریتم gSpan (یان و هان ۲۰۰۰) نخستین رویکردی بود که از جستجوی عمقی برای زیر گراف های پر تکرار استفاده کرد. این جستجو توسط سیستم برچسب گذاری مجاز نوینی پشتیبانی می شود. هر گراف با یک کد زنجیره ای منطبق می شود و تمام کدها بر حسب ترتیب واژگان نگاری افزایشی دسته بندی می شوند. سپس جستجوی عمقی برای درخت هایی که با اولین گره های کدها جفت می شوند اعمال می شود، و به تدریج داده های پرتکرار بیشتری اضافه می شوند. وقتی پشتیبانی که یک گراف به پایین تر از حداقل پشتیبانی می رسد یا در شرایطی که زیر گراف قبلا شناسایی شده باشد آنگاه این زیر گراف نقاط توقف را افزایش می دهد. الگوریتم gSpan از نظر زمان و حافظه مورد نیاز کارایی زیادی دارد.

الگوریتم ارائه شده در (بورگلت و برتولد ۲۰۰۲)، در شناسایی زیر ساختارهای پرتکرار در ترکیب های مولکولی تمرکز می کند و همچنین از استراتژی جستجوی عمقی برای پیدا کردن زیر گراف های پرتکرار استفاده می کند. در پی فرایند افزایش الگو، وقوع گسستگی درون تمام مولکول ها در تعادل نگه داشته می شود. ترتیب اتم

ها و پیوندهای محلی برای از بین بردن گسست های غیر ضروری که سرعت جستجو را پایین می آورند مورد استفاده قرار می گیرند و از ایجاد الگوهای زاید جلوگیری می کنند. الگوریتم FFSM (هوآن، وانگ و پرینس ۲۰۰۳) برای تعیین صریح زیر گراف های پرتکرار از چارچوب گراف های جبری استفاده می کند. هر گراف با استفاده از ماتریکس تجانب بیان می شود و با بهره گیری از تثبیت کننده های هر زیر گراف پر تکرار از هم ریختی زیر گراف جلوگیری می شود. الگوریتم GASTON (گراف، توالی، استخراج نمودارهای درختی) (نیجسن و کوک ۲۰۰۴) با استفاده از ایده آغاز سریع جستجوی ساختارهای گراف پرتکرار با تلفیق جستجوی مسیرها، درخت ها و گراف های پرتکرار در یک رویکرد خاص طراحی شد. رویکرد سطح-محور مورد استفاده قرار می گیرد که مسیرهای ساده تعیین شود و به دنبال آن ساختارهای درختی و اکثر ساختارهای گراف پیچیده در انتها تعیین شوند. این نوع رویکرد با این تفکر تقویت می شود که در عمل اکثر گراف ها واقعا پیچیده و دشوار نیستند و تعدادی از حلقه ها خیلی بزرگ نیستند. بنابراین، در مرحله تعیین و شمارش، ابتدا توالی ها / مسیرها تعیین می شود پس از آن ساختارهای درختی با اضافه شدن مداوم گره و کران همراه آن به گره های مسیر تعیین می شوند. سپس، ساختارهای گراف با اضافه کردن کران ها در میان گره های ساختار درختی تعیین می شوند.

۴,۳. روش های برنامه ریزی منطقی استقرایی (IPL)

رویکردهایی که در این دسته قرار می گیرند با استفاده از IPL برای بیان داده های گراف با استفاده از جملات برجسته ای معرفی می شوند. آنها از حوزه استخراج داده های چند رابطه ای می آیند، که در اصل استخراج داده هایی است که در جدول داده های در هم آمیخته چند گانه پایگاه داده های رابطه ای سازماندهی شده اند.

الگوریتم WARMR (دهاسپ و توی وونن ۱۹۹۹) برای استخراج پرسش های پرتکرار از پایگاه داده های DATALOG طراحی شده است. این الگوریتم در مقایسه با روش های استاندارد استخراج الگوهای پرتکرار از یک پارامتر (کلید) اضافی استفاده می کند که این پارامتر در اصل خاصیتی است که می بایست در تمام الگوهای استخراج شده گنجانده شود. این الگوریتم به کاربر اجازه می دهد فضای جستجو را به الگوهایی که دارای آیتم مطلوب هستند، محدود کند. الگوریتم مذکور از نظر نوع الگوهای قابل استخراج انعطاف پذیر است (محدودیت ندارد). زبان تسریع کننده ای (WRMODE) طراحی شده است که فضای جستجو را به پرسش های قابل قبول و بالقوه جالب محدود می کند. فرایندهای اصلی الگوریتم عبارتند از ایجاد داوطلب و ارزیابی داوطلب. مرحله ارزیابی داوطلب برای تعیین فراوانی داوطلب های پرتکرار مورد استفاده قرار می گیرد. در مرحله ایجاد داوطلب، الگوهایی که شامل کلیدهای معین هستند پیدا می شوند و به طور فزاینده ای گسترده می شوند، این کار از کوچکترین الگوها آغاز می شود. در هر مرحله، الگوهای پرتکرار، کم تکرار (نادر) تعیین می شوند و الگوهای پرتکرار با اضافه کردن یک گره در هر نوبت گسترده می شوند. الگوهایی که قبلا در مجموعه الگوهای پرتکرار

قرار گرفته اند حذف می شوند. یا اگر شکل تخصصی شده الگویی باشند که قبلا در مجموعه الگوهای پرتکرار وجود داشته، حذف می شوند. این روش در مورد تحلیل هشدارهای مخابراتی و سم شناسی شیمیایی مورد استفاده قرار گرفت. نقطه ضعف عمده الگوریتم به کارایی آن بر می گردد چرا که آزمایش تعادل بین بند های ردیف اول بسیار دشوار است. این مسئله نیحسن و کوک (۲۰۰۱) را واداشت تا یک راه حل جایگزین برای این مرحله پرهزینه پیشنهاد دهند. الگوریتم FARMER که توسط آن ها پیشنهاد شد هنوز هم از نشانه گذاری منطقی ردیف اول استفاده می کند و از بسیاری جهات شبیه به WARMR است، و از لحاظ زمان انجام عملیات پیشرفت قابل ملاحظه ای داشته است. تفاوت اصلی این دو الگوریتم این است که به جای آزمایش های پر هزینه تعادل WARMR برای محاسبه و ایجاد پرسش های کاندیدا به شیوه ای موثر، از ساختار داده های درختی استفاده می کند.دی رات و کرامر (۲۰۰۱) روشی را برای یافتن ذرات مولکولی پر تکرار از پایگاه داده های ترکیب های مولکولی طراحی کردند. یک ذره مولکولی به صورت توالی اتم هایی که به صورت خطی به یکدیگرند متصل شده اند تعریف می شود، و پایگاه داده ها شامل اطلاعاتی در باره ویژگی هایی از اتم های یک مولکول و ترتیب پیوند هاست . این رویکرد، تعیین ضابطه کلی را امکان پذیر می کند به گونه ای که تمام ذرات مولکولی اجتماع ضابطه های اولیه را عملی می کنند. علاوه بر پارامتر فراوانی مینیمم سنتی، رویکرد آنها استفاده از ضابطه فراوانی ماکزیمم برای الگو ها را امکان پذیر می کند. این ضوابط قابلیت انعطاف پذیری بیشتری به آن دسته از پرسش ها می دهد که از طریق الگو ها می توان به آنها پاسخ داد، و در حوزه پیدا کردن ذره های مولکولی، می توان الگو های جالب تری پیدا کرد. رویکرد IPL-محور دیگر در (لیسی و مالربا، ۲۰۰۴) ارائه شده است و یک زبان آمیخته و ترکیبی برای حفظ اطلاعات مربوط به ویژگی های ارتباطی و ساختاری در قوانین چند سطحی استخراج شده از روابط چندگانه پیشنهاد شد . برای تعیین ترتیب کلی و اپراتور اصلاح نزولی از رابطه های زیر گروهی پرسش ها استفاده می شود .

۴,۴. روش های جستجوی سختگیرانه

ویژگی های روش های جستجو – محور سختگیرانه این است که آنها از محاسبات مفرط و اضافی مورد نیاز برای جستجوی تمام زیرگراف های پرتکرار اجتناب می کنند و به همین خاطر از رویکرد سختگیرانه برای کاهش تعداد زیرگراف های بررسی شده استفاده می کنند. در ازای از دست دادن تعدادی از زیرگراف های پرتکرار، از پیچیدگی بیش از حد مسئله هم ریختی گراف جلوگیری به عمل می آید.

یکی از نخستین رویکردهای استخراج گراف تحت عنوان سیستم SUBDUE ساخته می شود (کوک و هولدر ۱۹۹۳) و بسیاری از اصلاحات و بهینه سازی های این سیستم انجام شده است (کوک و هولدر ۲۰۰۰ ، کوک و دیگران ۲۰۰۱ ، جانیر و هولدر و کوک ۲۰۰۲ ، نوبل و کوک ۲۰۰۳ ، هولدر و دیگران ۲۰۰۳ ، کتکار و هولدر و کوک ۲۰۰۵) سیستم SUBDUE بر اصل اندازه مینیمم توصیف (MDL) استوار است، که به صورت تعداد بیت

های لازم برای توصیف گراف اندازه گیری می شود. تعاریف مفهومی جایگزین نمونه های زیر گراف شناسایی شده می شود. این رویداد باعث فشرده شدن مجموعه داده های اصلی می شود و مبنایی برای شناسایی ساختارهایی که به صورت طبقاتی تعریف شده اند، فراهم می آورد. انگیزه این گونه روش ها، تعیین زیر ساختارهای به لحاظ ذهنی جالبی است که تفسیر داده ها را افزایش می دهند. سیستم SUBDUE در کنار استفاده از اصل MDL، امکان تلفیق سایر اطلاعات و یافته های پیشین برای تمرکز بر جستجوی زیرگراف های مناسب تر را به وجود می آورد. رویکرد جستجوی سختگیرانه (حریص) برای شناسایی زیر گراف های داوطلب از داده های موجود است. این رویکرد از گره های مجزا (تکی) شروع می شود و متناوبا زیر ساختارهایی با یک کران مجاور را تعمیم می دهند تا جایی که تمام گسترده های ممکن را پوشش بدهد. هر یک از زیر گراف های داوطلب جدید تعمیم (گسترش) داده می شود و الگوریتم برای شناسایی بهترین زیر گراف ها بر طبق اندازه مینیمم توصیف، تمام زیر گراف های ممکن را مورد بررسی قرار می دهد. هنگامی که تمام زیر گراف های ممکن مورد بررسی قرار بگیرند یا فرایند جستجو به محدودیت محاسبه برسد آنگاه الگوریتم متوقف خواهد شد. برای جلوگیری از تعمیم از زیر گراف هایی که اندازه توصیفی آنها افزایش خواهد یافت از تکنیک حذف انتخابی استفاده می شود. سیستم SUBDUE در جستجوی خود برای زیر گراف های نظیر، از الگوریتم جفت گراف های غیر دقیق استفاده می کند تا تغییرات ناچیز را امکان پذیر کند زیرا زیر ساختارهای جالب ممکن است اغلب اوقات به شکلی تقریبا متفاوت در داده ها دیده شوند.

همچنین، روش القا گراف (GBI) (یوشیدا و موتودا و ایندورکیا ۱۹۹۴) به منظور دستیابی به زیرگراف های جالب و کوچک، داده های گراف را فشرده کردند. این روش در ابتدا برای پیدا کردن مفاهیم جالب از الگوهای پرتکرار یافته شده در نتیجه گیری طراحی شد. تکنیک فشرده کردن در اصطلاح طبقه بندی جفت-مجور خوانده می شود، که دو گره درون یک گره با هم جفت می شوند. ارتباط بین گره های جفت شده از بین می رود، و در صورت لزوم رابط های (تکنیک های) بین سایر گره های گراف به خاطر سپرده می شوند به طوری که در هر زمانی در خلال فرایند جستجو بازسازی گراف اصلی امکان پذیر باشد. قطعه بندی جفت-محور را می توان به صورت مکانی مناسب در نظر گرفت و می توان گراف به طور متناوب فشرده کرد. اندازه کوچک انتخاب می شود تا مقدار فشردگی که بیانگر اندازه الگوهای استخراج شده و گراف فشرده است، محدود شود. جستجوی داوطلب های زیر گراف های محلی با استفاده از جستجوی فرصت طلبانه انجام می گیرد.

۴,۵. سایر روش ها

همه روش های بررسی شده در بخش های قبلی به خانواده روش های استخراج زیر ساختار های (زیر گراف های) پرتکرار تعلق دارند. اندازه جستجو که به سمت تحویل اتوماتیک داده های گراف رفته است به طور کلی بزرگ است و برای دستیابی به بررسی همه جانبه قوانین، محدودیت ها، برنامه های تخصصی و الگوریتم های

مختلف در حوزه استخراج داده ها به (Chakrabarti & Faloutsos, 2006; Washio & Motoda, 2003, Han & Kamber 2006) مراجعه کنید. این بخش بعضی از رویکردهای جایگزین برای استخراج داده های گراف را ارائه خواهد کرد که اهداف مختلف تحلیل داده ها یا نیازهای نرم افزاری مشخص یا محدودیت های خاص عامل اصلی برای روی آوردن به این رویکرد هاست.

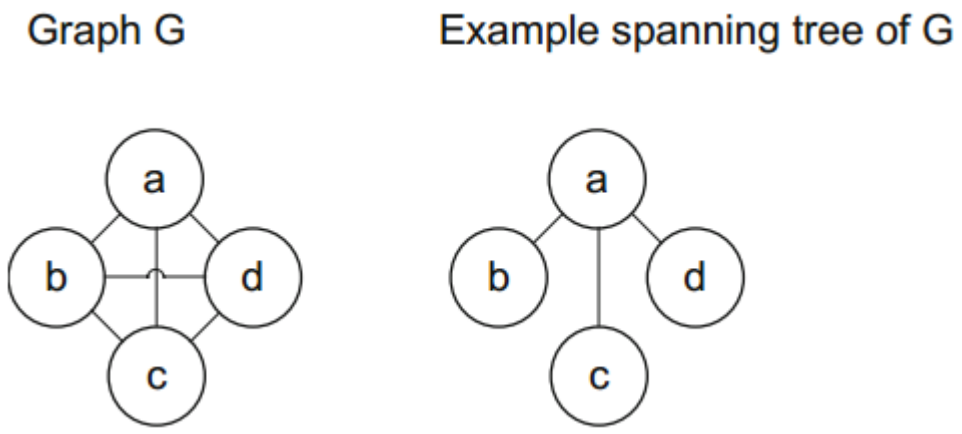
خوشه بندی داده های دارای ساختار گراف عموما در بسیاری از برنامه های کاربردی اهمیت ویژه ای دارند زیرا اغلب خوشه های شناسایی شده اغلب مبنایی برای تحلیل شباهت ها و تفاوت ها به وجود می آورند، و می توان از این خوشه ها برای طبقه بندی داده های گراف بر مبنای ویژگی های خوشه های تشکیل شده استفاده کرد. به عنوان مثال، تکنیک خوشه بندی گراف بر مبنای نظریه جریان شبکه برای تقسیم بندی سلسله تصاویر رزونانس (پژواک) مغناطیسی مغز انسان در (وو و لی هی ۱۹۹۳) اعمال شد. داده های مربوط به صورت کران تجانب غیر مستقیم بیان می شود در حالی که به هر کران یک ظرفیت جریان نسبت داده می شود که نشان دهنده شباهت گره های متصل به کران است. خوشه ها با حذف مداوم کران ها از گراف تشکیل می شوند تا وقتی که زیر گراف های متقابلا اختصاصی و منحصر به فرد تشکیل شود. کران ها براساس ظرفیت جریان شان حذف می شوند با این هدف که بزرگترین جریان ماکزیمم در میان زیر گراف های (خوشه های) تشکیل شده به نقطه مینیمم برسد. مانکوریوس و دیگران (۱۹۹۸) تعدادی از تکنیک های خوشه بندی را برای شناسایی اتوماتیک ساختار قطعه ای یک سیستم نرم افزاری از کد منبع آن طراحی کردند. کد منبع به عنوان یک گراف تابع پیمانه ای بیان می شود و برای شناسایی ساختار سطح بالای سازماندهی سیستم ها از ترکیب خوشه بندی، تپه نوردی و الگوریتم های تکمیلی و ریشه ای استفاده می شود.

تعدادی از الگوریتم های خوشه بندی گراف و بهینه سازی ظابطه خوشه بندی خاصی تمرکز می کنند که در چارچوب خوشه بندی گراف اتفاق می افتد و اغلب اوقات از روش های مربوط به مسائل بهینه سازی گراف های عمومی تر اقتباس می کنند. به عنوان مثال، یک رویکرد تحلیل خوشه تئوری – محور در (هارتو و شمیر ۲۰۰۰) ارائه شده است. گراف تشابه تعیین شده و خوشه ها به صورت زیر گراف هایی با قابلیت اتصال بیش از نیمی رأس ها تعریف می شوند و الگوریتم دارای پیچیدگی کمی است. الگوریتم خوشه بندی گراف ارائه شده در (فلیک و تارجان و تسیولیکلیس ۲۰۰۴) بر مبنای ایده کلی تکنیک های جریان ماکزیمم برای به حداکثر رساندن شباهت درون خوشه ای و به حداقل رساندن تشابه میان خوشه ای استوار است. یک سینک مصنوعی در گراف قرار داده می شود که به تمام گره ها متصل است و جریان های ماکزیمم بین تمام گره ها و سینک محاسبه می شوند. برای محدود کردن تعداد کران های مورد استفاده در ایجاد اتصال بین سینک مصنوعی و سایر گره های گراف، یک پارامتر ویژه انتخاب می شود. خوشه بندی بر مبنای گراف های درختی کوتاه شده مینیمم انجام می گیرد. گراف درختی کوتاه شده مینیمم یکی از زیر گراف های گراف اصلی است در شرایطی که تظمین می شود مسیر بین دو گره داده شده در گراف درختی کوتاه شده مینیمم کوتاه ترین مسیر بین این دو گره در گراف اصلی باشد. به همین دلیل، با انتخاب گراف های درختی کوتاه شده مینیمم از گراف های

گسترده، الگوریتم حلقه های خاصیت و اجزا به شدت پیوسته (متصل) را شناسایی می کنند. یک تابع اصلی بین دو گراف در (کاشیما و ستودا و اینوکوشی ۲۰۰۳) پیشنهاد شد. با محاسبه مسیرهای برچسب که در گراف ظاهر می شوند گراف ها به صورت یک بردار اصلی بیان می شوند. مسیرهای برچسب به وسیله تابع های تصادفی روی گراف ایجاد می شوند و هسته اصلی به صورت محصول داخلی بردارهای اصلی بیان می شود که میانگینی از تمام مسیرهای برچسب ممکن است. محاسبه هسته اصلی به مسئله مهمی در یافتن وضعیت ثابت و پایداری از یک سیستم خطی ناپیوسته تبدیل می شود و راه حل به صورت حل معادله های خطی متقارن با یک ماتریکس دارای ضریب پراکنده ظاهر می شود.

اکثر رویکرد های خوشه بندی شرح داده شده قادر به رفع کامل مسائل استخراج زیر گراف های پر تکرار به شکلی که در بخش ۳ مشخص شد، نمی باشند زیرا تظمینی وجود ندارد که زیر گراف های شناسایی شده با استفاده از روش های خوشه بندی تمام زیر گراف های پرتکرار برای پشتیبانی فرض شده را در بر بگیرد. به همین خاطر، این روش ها یک راه حل تقریبی ارائه می دهند و اغلب قادرند بر بعضی از دشواری هایی غلبه کنند که در هنگام نیاز به بررسی کامل تمام زیر گراف های پرتکرار ظاهر شوند.

روش تقریبی دیگری برای استخراج زیر گراف های پرتکرار بر مبنای گرافهای درختی در بر گیرنده ی (۲۰۰۷) ارائه شد. همانگونه که در بخش ۳ گفته شد، زیر گراف دربرگیرنده یک گراف باید شامل تمام رأس های گراف باشد. گراف درختی دربرگیرنده یک زیر گراف دربرگیرنده است که به شکل درخت است (بدون حلقه). به عبارت دیگر زیر گراف درختی دربرگیرنده گراف غیر مستقیم و پیوسته G، منتخبی از کران های G است که همه رأس های G را بر می گیرد و فاقد حلقه است. نمونه ای از زیر گراف های درختی دربرگیرنده در تصویر ۴ نشان داده شده است. الگوریتم مانکی (ژانگ و یانگ و چیولا ۲۰۰۷) بر مبنای این تفکر طراحی شد که در صورت تغییر شکل کران، انعطاف پذیری در بررسی هم ریختی گراف لازم به نظر می رسد به این منظور که الگوهای پر تکرار علیرغم وجود تغییرات در کران ها مورد بررسی قرار بگیرند. انعطاف پذیری در بررسی هم ریختی گراف بر اساس جستجوی گراف های درختی دربرگیرنده پر تکرار مطرح شد زیرا، در یک زیر گراف درختی دربرگیرنده، تمام رأس ها می بایست نمایش داده شوند در حالی که تا زمانی که زیر گراف مذکور هنوز به شکل گراف درختی است تعدادی از کران ها می توانند غایب باشند. مانکی (Monkey) بر جستجوی عمقی استوار است و در محدوده با هم تلفیق می شوند تا کنترل کران های نامرتب که بر فراوانی الگوهای متناظر اثر می گذارند، امکان پذیر شود.



تصویر ۴ نمونه ای از گراف درختی دربرگیرنده از یک گراف G

مسئله استخراج پایگاه داده های گراف دیسکت–محور در (وانگ و دیگران) مورد توجه قرار گرفته است. اصل مطلب به ساختار ADI (شاخص تجانب) مربوط است که در شرایطی که نمی توان پایگاه داده های گراف را در حافظه اصلی قرار داد از کارهای اصلی در فرایند استخراج گراف پشتیبانی می شود. الگوریتم gSpan (یان و هان ۲۰۰۲) که پیش تر مورد بررسی قرار گرفت برای استفاده از ساختار ADI انتخاب شد، و الگوریتم حاصل یعنی ADI-Mine نشان داده شد تا هنگامی که پایگاه داده های دیسکت-محور بزرگ مورد تردید قرار می گیرند الگوریتم gSpan عملکرد بهتری داشته باشد. الگوریتم ADI-Mine و بعضی از تکنیک های استخراج محدودیت-محور در هم ادغام شدند. تا یک سیستم Graph Miner (استخراج کننده گراف) تشکیل شود (وانگ و دیگران ۲۰۰۵).

این سیستم یک فاصل گرافیکی کاربر ایجاد می کند به طوری که می توان بعضی از محدودیت ها از جمله محدودیت اندازه الگو، کران / رأس هایی که باید تا نباید در الگو ظاهر شوند، گراف هایی که الگوهایشان باید ابر الگو یا زیر الگو باشند، و محدودیت های ترکیبی را انتخاب کرد . علاوه بر این، سیستم امکان تماشا و تحلیل آسان الگو ها و قابلیت های پرسشی به منظور تمرکز بر الگوهای دارای کاربرد ویژه یا جالب را به وجود می آورند.

سایگو و ستودا (۲۰۰۸) روش استخراج گراف متناوبی برای تحلیل اجزا بنیادی پیشنهاد کردند، که در آن الگوهای برجسته و مهم به طور فزاینده ای از طریق درخواست های جداگانه برای استخراج گراف جمع آوری می شوند. در هر درخواست استخراج گراف، مقادیر حقیقی به فعالیت های گراف نسبت داده می شوند و الگوهایی که آستانه پشتیبانی مقرر را برآورده می کنند مشخص می شوند. استراتژی جستجو بر مبنای یک الگوریتم شاخه -و- محدوده استوار است که از این گراف درختی کد جستجوی عمقی به عنوان فضای جستجوی مجاز استفاده می کند. الگوها به شکل گراف های درختی سازماندهی می شوند به گونه ای که یک گره کوچک دارای یک زیر گراف الگو از گره مادر (اصلی) است.

با تولید منظم الگوها از ریشه تا برگ های گراف به شیوه ای چرخشی و با استفاده از تعمیم اصلی ترین گره، الگوها مشخص می شوند.

۴,۶. استخراج الگوهای زیر گراف مسدود / بیشینه

استخراج زیر گراف های مسدود / بیشینه پرتکرار نیز مانند سایر مسائل استخراج الگوهای پرتکرار، حوزه مهم در تحقیقات مربوطه است زیرا یکی از روش های کاهش دشواری و پیچیدگی ناشی از تعیین تمام زیر گراف های پرتکرار است.

الگوریتم Close Graph (یان و هان ۲۰۰۳)، زیر گراف های پرتکرار مسدود را استخراج می کند و چارچوب کلی آن شبیه به الگوریتم gSpan (یان و هان ۲۰۰۲) است با این تفاوت که از تکنیک های انتخاب گراف های مناسب بهره می گیرد تا با کارآمدی بیشتری فضای جستجو را کوچک کند و تنها زیر گراف های مسدود را مشخص کند. این الگوریتم با حذف تمام گره ها و کران های اتفاقی (کم تکرار) آغاز به کار می کند و سپس تمام زیر گراف های پرتکرار شامل یک گره مجزا را تولید می کند. سپس، مجموعه گراف های پرتکرار را با استفاده از جستجوی عمقی و تعمیم گراف اصلی (بیشینه) گسترده می کند.

جستجوی عمق – محور (عمقی) بر مبنای ترکیب واژگان نگاری DFS عمل می کند، یعنی سیستم بر چسب گذاری مجاز نوین ابتدا در الگوریتم gSpan ارائه می شود. نویسندگان براساس هم ارزی وجود زیر گراف های پرتکرار در گراف اصلی یک حالت توقف اولیه تعریف می کنند، به طوری که نیازی به تعمیم یا بررسی تمام گرافها در راستای دستیابی به ویژگی زیر گراف مسدود نیست. این حالت (شرط) توقف اولیه برای تمام زیر گراف ها معتبر نیست و حالت دیگری برای بررسی این موارد اعمال می شود به این منظور که از مفقود شدن اطلاعات مربوط به زیر گراف های مسدود پرتکرار به طور بالقوه جلوگیری شود. این حالت ها به الگوریتم Close Graph اجازه می دهد حالت هایی را که امکان مسدود شدن تمام زیر گراف های حاصل از گراف پر تکرار وجود ندارد و نیازی به محاسبه وجود ندارد، مورد بررسی قرار دهد.

مسئله استخراج زیر گراف های پرتکرار با محدودیت های اتصال در گراف های ارتباطی در (یان و ژو و هان ۲۰۰۵) مورد بررسی قرار گرفته است و دو الگوریتم Close Cut و Splat پیشنهاد شده اند. الگوریتم Close Cut یک رویکرد افزایش (گسترش) الگو است، یعنی ابتدا زیر گراف های پر تکرار مشخص می شوند و با افزودن کران های جدید تعمیم داده می شوند. زیر گراف های داوطلب که بر اساس این فرایند ایجاد شده اند باید محدودیت های اتصال تعیین شده و آستانه فراوانی مینیمم را رفع کنند. گراف های داوطلب با اضافه کردن کران های جدید تعمیم داده می شوند و این کار تا زمانی ادامه پیدا می کند که گراف حاصل از افزودن کران ها دیگر پر تکرار نباشد. از سوی دیگر، الگوریتم Splat رویکردی بر مبنای کاهش الگو است که از آن طریق گراف های ارتباطی قطع و تجزیه می شوند تا گراف های دارای پیوستگی (ارتباط) زیاد به دست آید. در هر مرحله، الگوریتم

بررسی می کند که آیا گراف جدید تولید شده در مجموعه نتایج وجود دارد، و در این حالت نیازی به ادامه پردازش نیست زیرا نمی توان هیچ یک از گراف های پیوسته مسدود را از گراف داوطلب مشخص کرد.

الگوریتم MARGIN (توماس و والری و کارلا پالم ۲۰۰۶) زیر گراف های پرتکرار بیشینه را به شیوه بالا به پایین استخراج می کند. به منظور رکورد گراف از پایگاه داده های گراف، الگوریتم به طور مکرر یکی از کران ها را در هر نوبت حذف می کند بدون اینکه گراف های ناپیوسته (غیر مرتبط) ایجاد کند؛ این کار تا زمانی ادامه پیدا می کند که نمونه پرتکرار گراف مورد نظر پیدا شود. این نمونه پرتکرار احتمالا یک الگوی بیشینه پرتکرار خواهد بود زمانی که تعدادی از کران ها وگره ها اتفاقی حذف شده باشند. به همین خاطر، پروسه ای متناوب بر زیر گراف های پیوسته پرتکرار اعمال می شود که به طور فزاینده ای کران ها و گره های اتفاقی را از زیر گراف تعیین شده حذف می کند و این روند ادامه پیدا می کند تا زمانی که ویژگی های زیر گراف بیشینه پرتکرار برآورده شود.

۵. نتیجه گیری

این بخش مقدمه ای مختصر در مورد مسئله استخراج گراف ارائه داده است که به گراف های حلقه ای که تمرکز اصلی کتاب بر آنهاست، محدود نمی شود. برخی تعاریف رسمی با مثال های تصویری ارائه شده تا به خواننده امکان درک پیچیدگی و دشواری مسئله و بعضی از ویژگی های نامطلوب که عامل این دشواری هستند، داده شود. این دشواری و پیچیدگی دلیل اصلی طراحی تعداد زیادی رویکردهای مختلف برای حل مسئله استخراج گراف را شرح می دهد (که به این نکته در بخش ۴ اشاره شده است). می توانیم مشاهده کنیم که تجربه های تازه اغلب زمانی به وجود می آیند که مسئله تا حدی تخفیف پیدا می کند. به طور مثال در مورد الگوریتم GASTON (نیحسن و کوک ۲۰۰۴) که با شناسایی الگوهای ساده تر در ابتدای فرایند و افزایش پیچیدگی با حرکت از توالی ها، درخت ها و در نهایت گراف های حلقه ای به یک شروع سریع و چشم گیر دست می یابد. با این وجود، لازم به ذکر است علیرغم اینکه مسئله استخراج گراف از نظر تئوری علمی نیست ولی در شرایط علمی الگوریتم های موجود می توانند این عملیات را در چارچوب زمانی معقول برای داده های دارای ساختار گراف به اتمام برسانند. کاربرد های استخراج گراف گوناگون و متنوع است و به طور کلی در امر تحلیل داده ها در هر حوزه ای مفید و سودمند هستند با فرض اینکه داده ها در یک ساختار گراف سازماندهی شده باشند. وقتی الگوریتم ها را مورد بحث و بررسی قرار می دادیم به برخی از حوزه های کاربردی اشاره کردیم و می توانیم به طور کلی ادعا کنیم که برنامه های کاربردی تا حدودی به بسیاری از نرم افزارهای گراف های درختی که در فصل ۹ بررسی شد، شباهت دارند با این تفاوت عمده که در استخراج گراف، داده هایی کاربردی شامل حلقه هایی در میان موضوعات داده هاست. از دیگر نمونه های برنامه های کاربردی می توان به تحلیل ترکیب های شیمیایی، تحلیل شبکه های اجتماعی، استخراج ساختار وب، استخراج مباحث مربوط به هستی

شناسی، و به طور کلی استخراج تکنیک های سودمند و موثر برای بسیاری از نرم افزار های وب و رسانه های اجتماعی اشاره کرد.

مراجع

[1] C. Aggarwal, N. Ta, J. Feng, J. Wang, M. J. Zaki. XProj: A Framework for Projected Structural Clustering of XML Documents,KDD Conference, 2007.

[2] R. Agrawal, A. Borgida, H.V. Jagadish. Efficient Maintenance of transitive relationships in large data and knowledge bases, ACM SIGMOD Conference, 1989.

[3] D. Chakrabarti, Y. Zhan, C. Faloutsos R-MAT: A Recursive Model for Graph Mining.SDM Conference, 2004.

[4] J. Cheng, J. Xu Yu, X. Lin, H. Wang, and P. S. Yu, Fast Computing Reachability Labelings for Large Graphs with High Compression Rate,EDBT Conference, 2008.

[5] J. Cheng, J. Xu Yu, X. Lin, H. Wang, and P. S. Yu, Fast Computation of Reachability Labelings in Large Graphs,EDBT Conference, 2006.

[6] E. Cohen. Size-estimation framework with applications to transitive closure and reachability,Journal of Computer and System Sciences, v.55 n.3, p.441-453, Dec. 1997.

[7] E. Cohen, E. Halperin, H. Kaplan, and U. Zwick, Reachability and distance queries via 2-hop labels,ACM Symposium on Discrete Algorithms, 2002.

[8] D. Cook, L. Holder, Mining Graph Data,John Wiley & Sons Inc, 2007.

[9] D. Conte, P. Foggia, C. Sansone, and M. Vento. Thirty years of graph matching in pattern recognition.Int. Journal of Pattern Recognition and Artificial Intelligence, 18(3):265–298, 2004.

[10] M. Faloutsos, P. Faloutsos, C. Faloutsos, On Power Law Relationships of the Internet Topology.SIGCOMM Conference, 1999.

[11] G. Flake, R. Tarjan, M. Tsioutsiouliklis. Graph Clustering and Minimum Cut Trees,Internet Mathematics, 1(4), 385–408, 2003.

[12] D. Gibson, R. Kumar, A. Tomkins, Discovering Large Dense Subgraphs in Massive Graphs,VLDB Conference, 2005.

[13] M. Hay, G. Miklau, D. Jensen, D. Towsley, P. Weis. Resisting Structural Re-identification in Social Networks,VLDB Conference, 2008.

[14] H. He, A. K. Singh. Graphs-at-a-time: Query Language and Access Methods for Graph Databases. InProc. of SIGMOD '08, pages 405–418, Vancouver, Canada, 2008.

[15] H. He, H. Wang, J. Yang, P. S. Yu. BLINKS: Ranked keyword searches on graphs. InSIGMOD, 2007.

[16] H. Kashima, K. Tsuda, A. Inokuchi. Marginalized Kernels between Labeled Graphs,ICML, 2003.

[17] L. Backstrom, C. Dwork, J. Kleinberg. Wherefore Art Thou R3579X? Anonymized Social Networks, Hidden Patterns, and Structural Steganography.WWW Conference, 2007.

[18] T. Kudo, E. Maeda, Y. Matsumoto. An Application of Boosting to Graph Classification,NIPS Conf.2004.

[19] J. Leskovec, J. Kleinberg, C. Faloutsos. Graph Evolution: Densification and Shrinking Diameters. ACM Transactions on Knowledge Discovery from Data (ACM TKDD), 1(1), 2007.

[20] K. Liu and E. Terzi. Towards identity anonymization on graphs.ACM SIGMOD Conference 2008.

[21] R. Kumar, P Raghavan, S. Rajagopalan, D. Sivakumar, A. Tomkins, E. Upfal. The Web as a Graph.ACM PODS Conference, 2000.

[22] S. Raghavan, H. Garcia-Molina. Representing web graphs. ICDE Conference, pages 405-416, 2003.

[23] M. Rattigan, M. Maier, D. Jensen: Graph Clustering with Network Sructure Indices. ICML, 2007.

[24] H. Wang, H. He, J. Yang, J. Xu-Yu, P. Yu. Dual Labeling: Answering Graph Reachability Queries in Constant Time.ICDE Conference, 2006.

[25] X. Yan, J. Han. CloseGraph: Mining Closed Frequent Graph Patterns, ACM KDD Conference, 2003.

[26] X. Yan, H. Cheng, J. Han, and P. S. Yu, Mining Significant Graph Patterns by Scalable Leap Search,SIGMOD Conference, 2008.

[27] X. Yan, P. S. Yu, and J. Han, Graph Indexing: A Frequent Structure-based Approach,SIGMOD Conference, 2004.

[28] M. J. Zaki, C. C. Aggarwal. XRules: An Effective Structural Classifier for XML Data,KDD Conference, 2003.

[29] B. Zhou, J. Pei. Preserving Privacy in Social Networks Against Neighborhood Attacks.ICDE Conference, pp. 506-515, 2008.