مدیریت و استخراج داده های گراف: ارزیابی الگوریتم ها و برنامه های کاربردی

چکیده

استخراج و مدیریت گراف به خاطر کاربردهای متعددی که در دامنه وسیعی از حوزه های حقیقی از قبیل زیست کامپیوتری، مکان یابی عیب یاب های نرم افزار و ایجاد شبکه های کامپیوتری دارد، به قلمرو تحقیقی متداولی در سال های اخیر تبدیل شده است. برنامه های کاربردی مختلف منجر به پیدایش گراف هایی با اندازه ها و پیچیدگی های گوناگون شده اند. بر همین مقیاس، برنامه های کاربردی دارای الزامات مختلفی برای الگوریتم های استخراج اصلی هستند. در این فصل، ارزیابی از انواع مختلف الگوریتم های استخراج و مدیریت گراف در اختیار شما قرار می گیرد. همچنین، تعدادی از برنامه های کاربردی که وابسته به بازنمایی گراف هستند را مورد بحث قرار خواهیم داد. به این نکته خواهیم پرداخت که چگونه الگوریتم های مختلف استخراج گراف برای کاربردهای مختلف برگزیده و اعمال می شوند. در نهایت، مسیرهای مهم برای تحقیقات آتی در این حوزه را بررسی خواهیم کرد.

واژگان کلیدی: استخراج گراف، مدیریت گراف

۱. مقدمه

استخراج و مدیریت گراف به واسطه کاربردهای متعددی که در دامنه وسیعی از حوزه های حقیقی از قبیل زیست کامپیوتری، مکان یابی عیب یاب های نرم افزار، و ایجاد شبکه های کامپیوتری دارد، در سال های اخیر به قلمرو تحقیقی متداولی تبدیل شده است، علاوه بر این، انواع جدیدی از داده ها مانند داده های نیمه ـ ساختاری و XML [۸] را می توان به شکل گراف ها ارائه کرد. بحث مشروح درباره ی انواع گوناگون الگوریتم های استخراج گراف در [۵۸] قابل مشاهده است.

در حوزه گراف، الزام و نیاز به برنامه های کاربردی مختلف یکنواخت نیست. بنابراین، الگوریتم های استخراج گراف که در یک حوزه عملکرد خوبی دراند ممکن است در حوزه ی دیگر عملکرد خوبی نشان ندهند. به عنوان مثال، اجازه دهید حوزه های داده های زیر را بررسی کنیم.

- **داده های شیمیایی:** داده های شیمیایی اغلب به صورت گراف هایی بیان می شوند که گره ها متناظر با اتم ها و اتصال ها متناظر با پیوندهای بین اتم ها هستند. زیر ساختارهای داده ها نیز ممکن است به عنوان گره های اختصاصی مورد استفاده قرار بگیرند. در این حالت، گراف های اختصاصی کاملاً کوچک هستند، با این همه، کپی های متعددی در میان گره های مختلف وجود دارند. این موضوع منجر به چالش های هم ریختی در برنامه هایی مثل تناظر گراف می شود. چالش هم ریختی این است که گره ها در یک جفت گراف مشخص از جهات گوناگونی باهم تناظر و شباهت داشته باشند. تعداد تناظرهای موجود کاربردی از جمله استخراج الگوی پرتکرار، تناظر گراف و طبقه بندی گراف، موضوع مهمی به شمار می رود.
- **داده های زیستی:** داده های زیستی به شیوه ای مشابه با داده های شیمیایی طراحی می شوند. با این وجود، گراف های اختصاصی به طور معمول اندازه بسیار بزرگتری دارند. به علاوه، گره ها معمولاً بخش های مدل های زیستی را با دقت طراحی می کنند. آمینواسید می تواند نمونه ای بارز از یک گره در برنامه ی کاربردی DNA باشد. هر شبکه ی زیستی مجزا به سادگی می تواند شامل هزاران گره باشد. اندازه ی پایگاه داده های کلی نیز به حد کافی برای گراف های اصلی بزرگ هست که روی دیسک ذخیره شود. ماهیت ذخیره سازی ـ روی دیسک مجموعه داده ها اغلب به موضوعات منحصر بفردی منتهی می شود که در سایر طرح ها با آن مواجه نمی شویم. به عنوان مثال، ترتیب دستیابی کران ها در گراف در این حالت بسیار جدی تر و مهم تر می شود، هر الگوریتمی که برای دسترسی تصادفی به کردن ها طراحی شده باشد کارایی چندانی در این حالت نخواهد داشت.
- **داده های شبکه ای شده کامپیوتری و داده های وب:** در مورد شبکه های کامپیوتری و وب، تعداد گره ها در گراف اصلی ممکن است بسیار انبوه باشد. وقتی تعداد گره ها انبوه باشد موجب پیدایش تعداد بیشماری از کران های متمایز می شود. این پدیده به عنوان مسئله قلمرو انبوه در داده های شبکه ای نیز معرفی می شود. در اینگونه موارد، تعداد کران های متمایز ممکن است آنقدر زیاد باشد که نگهداری

آنها در فضای ذخیره سازی موجود دشوار شود. از این رو، باید تکنیک هایی برای فشرده کردن وکار با بازنمایی های فشرده ی مجموعه داده ها پدیدار شوند. در اینگونه موارد، چالش دیگری از این واقعیت بر می خیزد که ذخیره ی کران های تازه وارد برای تحلیل در آینده امکان پذیر نخواهد بود. با این وجود، تکنیک های فشرده سازی به ویژه برای این حالت الزامی و حیاتی است. فشرده ی زنجیره ممکن است برای پردازش گراف های اصلی در آینده ذخیره شود.

- **داده های XML:** داده های XML فرمی طبیعی از داده های گراف نسبتاً کلی هستند. خطر نشان می کنیم که الگوریتم های استخراج و مدیریت برای داده های XML در مورد گراف ها نیز کاملاً کارآمد و سودمند خواهند بود، زیرا داده های XML را می توان به عنوان گراف های برچسب دار در نظر گرفت. به علاوه، ترکیب های خاصیت ـ مقدار همراه با گره ها می تواند این مسئله را چالشی تر سازد. با این وجود، تحقیق در حوزه داده های XML اغلب کاملاً مستقل از تحقیق در حوزه استخراج گراف بوده است. با این حال، در این فصل تلاش خواهیم کرد الگوریتم های استخراج داده های XML را همراه با الگوریتم های استخراج و مدیریت گراف مورد بحث قرار دهیم. امید است که با این کار بتوانیم ارزیابی منسجم تری از این حوزه داشته باشیم.

بدیهی است که طراحی یک الگوریتم استخراج ویژه به حوزه کاربردی مرتبط با آن بستگی دارد. به عنوان مثال، یک مجموعه داده قابل ذخیره سازی روی دیسک نیازمند طراحی دقیق یک الگوریتم است که در آن کران های هر گراف به طور تصادفی قابل دستیابی نباشند. همچنین، شبکه های دارای حوزه فراگیر و گسترده نیازمند فشرده سازی دقیق گراف های اصلی به منظور تسهیل فرآیند هستند. از سوی دیگر، مولکول های شیمیایی حاوی کپی های فراوان از گره ـ برچسب ها چالش منحصر بفردی در قالب هم ریختی گراف را به انواعی از برنامه های کاربردی تحمیل می کنند.

در این فصل، انواع مختلفی از برنامه های کاربردی استخراج و مدیریت گراف به همراه کاربردهای متناظر با آن را مورد بررسی و بحث قرار خواهیم داد. به این نکته اشاره می کنیم که مرز بین الگوریتم های استخراج و مدیریت داده ها اغلب اوقات روشن نیست، زیرا بسیاری از الگوریتم ها را می توان در هر دو گروه طبقه بندی کرد. موضوعات مطرح شده در این فصل را می توان اساساً به سه گروه تقسیم کرد. این گروه ها به شکل زیر به بحث گذاشته می شوند:

- **الگوریتم های مدیریت گراف:** این عنوان به الگوریتم هایی برای مدیریت و شاخص گذاری حجم زیادی از داده های گراف باز می گردد. در این بخش، الگوریتم هایی برای شاخص گذاری گراف ها و همین طور پردازش پرس و جوهای گراف ارائه خواهیم کرد. انواع دیگری از پرسش ها از قبیل پرس وجوهای مرتبط با دسترسی را نیز مورد تحقیق قرار خواهیم داد. و به تحقیق درباره ی الگوریتم های تناظر و تطبیق گراف ها و کاربردهای آنها خواهیم پرداخت.

- **الگوریتم های استخراج گراف:** این به الگوریتم هایی که برای استخراج الگوها، روندها، مسیرها و خوشه ها از گراف به کار می روند، اشاره دارد. در بعضی از موارد، ممکن است لازم باشد الگوریتم ها را بر مجموعه های بزرگی از گراف های موجود در دیسک اعمال کنیم. روش هایی برای خوشه بندی، طبقه بندی و استخراج الگوی پرتکرار را بررسی خواهیم کرد. همچنین، بررسی مشروحی از این الگوریتم ها در آثار این حوزه ارائه خواهیم کرد.
- **کاربردهای مدیریت و استخراج داده های گراف:** حوزه های کاربردی مختلف که در آن الگوریتم های مدیریت و استخراج داده های گراف مورد نیاز است را مورد مطالعه قرار خواهیم داد. این حوزه عبارتند از: داده های وب، شبکه های اجتماعی و کامپیوتری، داده های زیستی و شیمیایی، و مکان یابی عیب یاب های نرم افزار. این فصل به شکل زیر سازماندهی شده است. در بخش بعدی، انواعی از الگوریتم های مدیریت داده های گراف را بررسی خواهیم کرد. در بخش ۳، الگوریتم های استخراج داده های گراف را مورد بحث قرار خواهیم داد. انواعی از حوزه های کاربردی که در آنها این الگوریتم ها استفاده می شود را در بخش ۴ بررسی خواهیم کرد. بخش ۵ به بررسی نتیجه گیری و خلاصه مطالب می پردازد. راهنمایی های تحقیقی تکمیلی در همین بخش مورد بحث قرار خواهد گرفت.

۲. الگوریتم های مدیریت داده های گراف

مدیریت داده های گراف چالشی تر از داده های چند بعدی شده است. باز نمایی ساختاری گراف دارای توانایی گویایی بیشتری است، اما این گویایی در ازای هزینه ای به دست می آید. این هزینه در قالب دشواری بازنمایی، دسترسی و پردازش ظاهر می شود زیرا عملیات میانی از جمله ارزیابی شباهت، معدل گیری و محاسبه ی فاصله و بعد ذاتاً در داده های ساختاری به شیوه ی داده های چند بعدی تعیین نمی شود. علاوه براین، پایگاه داده های ارتباطی سنتی با استفاده از قطعه خواندن ـ نوشتن به طور کارآمدی قابل دسترسی است؛ این برای داده های ساختاری که در آن کران ها در یک ترتیب اختیاری قابل دسترسی اند، طبیعی به نظر نمی رسد. با این وجود، پیشرفت های اخیر قادر به کاهش دست کم بعضی از نگرانی ها بوده اند. در این بخش، ارزیابی بسیاری از الگوریتم ها و برنامه های اخیر در زمینه ی مدیریت گراف ارائه خواهیم داد.

۲,۱. تکنیک های شاخص گذاری و پردازش پرس و جو

مدل های پایگاه داده و زبان های پرسش موجود، شامل مدل ارتباطی و SQL، فاقد پشتیبان طبیعی برای ساختارهای داده های پیشرفته مثل نمودارهای درختی و گراف ها هستند. اخیراً، با توجه به استفاده گسترده از XML به عنوان فرصتی برای تبادل داده ها، تعدادی از مدل های داده ها و زبان های پرسش جدیدی برای ساختارهای درخت ـ مانند پیشنهاد شده اند. در این اواخر، موج جدیدی از برنامه های کاربردی در حوزه مختلفی

مثل شبکه، مدیریت هستی شناسی، بیوانفورماتیک، غیره... به دنبال مدل های داده ی جدید، زبان ها و سیستم های نوین برای داده های دارای ساختار گراف بوده اند.

به طور کلی، این وظیفه ساده خواهد بود اگر به شکل زیر ارائه شود: برای یک الگوی پرس وجو (نمودار درختی یا گراف) نمودارهای درختی یا گراف هایی را در پایگاه داده ها پیدا کنید که دارای یا مشابه الگوی پرس و جو باشند. برای تحقق مؤثر و دقیق این وظیفه، می بایست به چند موضوع مهم بپردازیم: (i) چگونه داده ها و پرس و جو را طراحی کنیم؛ (ii) چگونه داده ها را ذخیره کنیم؛ و (iii) چگونه داده ها را برای پرس و جوی مؤثر شاخص گذاری کنیم.

پردازش پرس و جوی داده هایی به شکل نمودارهای درختی. تحقیقات زیادی در مورد پردزاش پرس و جوی XML انجام شده است. در سطح بالا، دو رویکرد برای طراحی داده های XML وجود دارد. یکی از این رویکردها ذخیره ی مدل ارتباطی موجود پس از انطباق داده های نمودار درختی با طرح ارتباطی است [۱۶۹]. رویکرد دیگر، ایجاد پایگاه داده طبیعی XML از خط شروع است. [۱۰۶] برای مثال، بعضی اقدامات با ایجاد نمودار درختی جبر و آنالیز برای داده های XML آغاز به کار کرده اند [۱۰۷]. گراف جبری پیشنهادی با تعیین اپراتورهای جدید مانند حذف گره و نصب گره برای داده های دارای ساختار گراف، جبر ارتباطی را توسعه داد.

SQL روش دسترسی استانداری برای داده های ارتباطی است. تلاش های زیادی برای طراحی یک بدیل SQL برای داده های نمودار درختی انجام شده است. معیارها عبارتند از: نخست، توان گویایی که انعطاف پذیری بیشتری برای ارائه ی پرس و جو از طریق داده های نمودار درختی به کاربر می دهد؛ و دوم قابلیت اظهار که به سیستم اجازه می دهد پردازش پرس و جو را بهینه سازی کند. استفاده ی گسترده از XML، گروه های استاندارد را وادار کرده است مشخصات SQL را به گونه ای تعمیم دهد که تابع های پردازش XML را در برگیرد. XQuery [۲۶] با استفاده از ساختار FLwor مسیر Xpath را برای بیان پرسش گسترش می دهد. ساختار Flwor شبیه به ساختارا SELECT – FROM – WHERE است با این تفاوت که پیشتیبان اضافی برای داده های نمودار درختی ایجاد می کند و توسط کنسرسیوم شبکه ی جهان وب (W3C) به عنوان زبان پررسش برای اسناد XML معرفی شده است.

در داده های XML، هسته ی پردازش پرس و جو در تناظر کارآمد و مؤثر الگوی نمودار درختی نهفته است. اکثر تکنیک های شاخص گذاری XML در [۱۱۵، ۵۱، ۵۹، ۱۳۲، ۱۴۱، ۸۵] ارائه شده اند تا از این فعالیت حمایت کند. به عنوان مثال Data Guide [۸۵] یک خلاصه ی کوتاه از ساختار مسیر در پایگاه داده های نمودار درختی ارائه می دهد از سوی دیگر، Tindex [۱۴۱] مجموعه مشخصی از این نظر که تمام مسیرهای برچسب دار که از قسمت ریشه شروع می شوند را حفظ می کند. Index Fabric هر یک از مسیرهای برچسب در هر یک از عامل های XML را با مقادیر داده ها به عنوان یک زنجیره کدگذاری می کند و هر مسیر بر حسب رمزدار و مقدار داده را در یک شاخص برای زنجیره هایی مثل نمودار درختی پاتریشا قرار می دهد. APEX [۵۱] از الگوریتم های

استخراج داده برای پیدا کردن مسیرهایی که مکرراً در پروسه ی پرس و جو ظاهر می شوند، استفاده می کند. در حالیکه اکثر تکنیک ها بر عبارات مسیر ساده تمرکز می کنند، FBInder [۱۱۵] بر انشعاب عبارات مسیر تأکید می کند. با این همه، از آنجایی که محتوای یک نمودار درختی به گره، مسیر یا شاخه های کوچک تجزیه می شود، به هم چسباندن نتایج میانی به فرآیندی زمان بر تبدیل شده است. علامت گذاری XML توالی ـ مبنا [۱۸۶، ۱۵۹، ۱۸۵]، الگوهای نمودار درختی را به شهروند درجه اول در پردازش محتوای XML تبدیل می کند. این عملیات، اسناد XML و متن ها ار به توالی ها و زنجیره هایی تبدیل می کند و پردازش محتوای نمودار درختی را از طریق تطابق و تناظر توالی ها (غیر مجاور) انجام می دهد.

پردازش محتوای داده های گراف. یکی از ویژگی های مشترک در دامنه ی وسیعی از برنامه های کاربردی نوظهور از قبیل شبکه اجتماعی، مدیریت هستی شناسی، شبکه / مسیرهای زیستی و غیره... این است که داده هایی که به آنها مربوط است همگی دارای ساختار گراف هستند. هرچه اندازه و پیچیدگی داده ها افزایش پیدا می کند، مدیریت آن با یک سیستم پایگاه داده ها اهمیت بیشتری پیدا می کند.

چندین روش برای مدیریت گراف ها در پایگاه داده ها وجود دارد. یک احتمال اینست که برای پشتیبانی داده های گراف یک موتور RDBMS تجاری را عرضه کنیم. احتمال دیگر، استفاده از جدول های ارتباطی هدف عمومی برای ذخیره گراف هاست. وقتی این روش ها قادر به ارائه عملکرد مورد انتظار نباشند، تحقیقات اخیر با چالش طراحی یک پایگاه داده گراف با هدف مشخص نیز مواجه خواهد شد. در حال حاضر، اوراکل (oracle) تنها DBMS تجاری است که پشتیبانی داخلی برای داده های گراف ایجاد می کند. پایگاه داده های ده گیگا بایتی جدید آن شامل مدل داده های شبکه ی فضایی اوراکل است (۳)، که به کاربران امکان می دهد داده های گراف را طراحی کنند و به کار بگیرند. مدل شبکه شامل اطلاعات منطقی مانند قابلیت اتصال بین گره ها و لینک ها، دستورالعمل لینک ها، برآوردهای گره ها و لینک ها و غیره است. مدل منطقی عمدتاً از طریق دو جدول قابل فهم است: جدول گره و جدول لینک، که اطلاعات اتصال گراف را ذخیره می کنند. هنوز هم، خیلی ها نگرانند که مدل ارتباطی اساساً برای پشتیبانی داده های گراف مناسب نیست، زیرا حتی بنیادی ترین عملیات مانند پیمودن گراف برای اجرا در DMBS های ارتباطی پر هزینه هستند. به ویژه وقتی که گراف های بزرگی داشته باشیم. اخیراً علاقه به شبکه ی معنای، توجه فزاینده ای را به Resource Description Framework (RDF) (چارچوب تشریح منابع) معطوف کرده است [۱۳۹]. ذخیره ی سه بعدی (Triple store) پایگاه داده ای با هدف ویژه برای ذخیره و بازیابی داده های RDF است. حافظه سه بعدی، بر خلاف پایگاه داده های ارتباطی، برای ذخیره و بازیابی تعداد زیادی از عبارت های کوتاه در قالب فاعل ـ گزاره، مفعول که سه بعد خوانده می شوند بهینه سازی شده است. تلاش های زیادی برای پشتیبانی دسترسی مؤثر به داده های ذخیره شده روی حافظه سه بعدی انجام گرفته است [۱۴، ۱۵، ۱۹، ۳۳، ۹۱، ۱۵۲، ۱۸۲، ۱۹۵، ۳۸، ۹۲، ۱۹۴، ۱۹۳]. اخیراً، گروه شبکه معنایی چالش سه بعدی [۴] را اعلام کرد، که بیش از پیش ضرورت و فوریت برای پشتیبانی استنباط و استنتاج در داده های RDF وسیع و حجیم را آشکار می کند.

تعدادی از زبان های پرس و جوی گراف از اوایل ۱۹۹۰ معرفی شده اند. به عنوان مثال، GraphLog [۵۶]. که ریشه در DataLog دارد، استنتاج قوانین و اصول درباره مسیرهای گرافی که از طریق عبارات متعارف و منظم بیان شده اند را اجرا می کند. GOOD [۸۹] که ریشه هایش در پایگاه داده های مفعول ـ محور قرار دارد، یک زبان تغییر شکل تعریف می کند که حاوی پنج عملیات بنیادی روی گراف است. GraphDB [۸۸] مدل دیگری از داده های مفعول ـ محور در زبان پرس و جوی دیگری برای گراف هست که پرس و جوی گراف را در چهار مرحله پیاده می کند، که هر یک از این مراحل عملیاتی را بر گراف های فرعی مشخص شده بوسیله عبارات منظم اجرا می کند. Graph QL [۹۷] ، بر خلاف سایر زبان های پرس و جوی قبلی که روی گره ها، کران ها یا مسیرها عمل می کنند، به طور مستقیم روی گراف ها عمل می کند. به عبرات دیگر، گراف ها به عنوان نوع بازگشتی و بازده تمام فعالیت ها مورد استفاده قرار می گیرد. Graph QL عمل کننده های جبری ارتباطی از قبیل مجموعه، حاصل ضرب دکارتی، و عملیات مجموعه برای ساختار های گراف هستند. به عنوان مثال، عمل کننده مجموعه برای تناظر الگویی گراف جمع بندی شده است. Graph QL به لحاظ ارتباطی کامل است و نسخه غیر متناوب آن معادل جبر ارتباطی است. شرح مفصل Graph QL و مقایسه آن با سایر زبان های پرس و جوی گراف را می توانید در [۹۶] بیابید.

با ظهور برنامه های کاربردی شبکه معنایی، نیاز به داده های RDF کار آمد مورد توجه قرار گرفته است. زبان پرس و جوی SPARQL [۱۵۴] برای این هدف طراحی شده است. به طوری که پیش تر گفتیم، یک گراف در فرمت RDF به وسیله مجموعه ای از سه بعد توصیف می شود که هر یک از آنها متناظر با یک کران بین دو گره است. پرس و جوی SPARQL، که SQL ـ مانند نیز هستند احتمالاً مشکل از الگوهای سه بعدی، پیوستگی ها، گسستگی ها، و الگوهای مطلوب است. الگوی سه بعدی از نظر نحوی به سه بعدی RDF نزدیک است با این تفاوت که هر یک از فاعل، گزاره یا مغول ها ممکن است متغیر باشند. پردازشگر پرس و جوی SPARQL، مجموعه ای از سه بعدی ها را جستجو می کند که با الگوهای سه بعدی متناظر باشند، و متغیرهای موجود در پرس و جوی گراف را به بخش های متناظر هر سه بعدی متصل می کند [۱۵۴].

روند کاری دیگری در شاخص گذاری گراف از ویژگی های ساختاری مهم گراف اصلی به منظور تسهیل شاخص گذاری و پردازش پرس و جود استفاده می کند. این ویژگی های ساختاری می تواند به شکل مسیرها یا الگوهای پرتکرار در گراف های اصلی باشند. از اینها می توان به عنوان فیلترهای پیش ـ پردازش که گراف های نامتناسب داده های اصلی را در مرحله ابتدایی حذف می کنند، استفاده کرد. به عنوان مثال، تکنیک GraphGrey [۱۸۳] از مسیرهای تعیین شده به عنوان ویژگی های شاخص استفاده می کند که به منظور فیلتر کردن گراف های نامتناظر می توان از آنها استفاده کرد. همچنین، تکنیک GIndex [۲۰۱] از قطعه های پرتکرار قابل تشخیص به عنوان ویژگی های شاخص استفاده می کند. یک تکنیک کاملاً مرتبط [۲۰۲]، به منظور تسهیل شاخص گذاری به ذخیره زیر ساختارها در گراف های اصلی می پردازد. روش دیگر شاخص گذاری گراف ها، استفاده از ساختارهای نمودار درختی [۲۰۸] در گراف های اصلی به منظور تسهیل شاخص گذاری و جستجو است.

موضوع پردازش پرس و جو در داده های گراف سالیان متمادی مورد مطالعه قرار گرفته است، ولی بسیاری از چالش های مرتبط با آن به جای خود باقی است. از یک سو، داده ها به طور فراینده ای در حال افزایش هستند؛ یک احتمال برای کنترل و استفاده از چنین داده های عظیمی از طریق پردزاش موازی با استفاده از، مثلاً چارچوب Map/Reduce است. با این وجود، به خوبی می دانیم که بسیاری از الگوریتم های گراف نیز وجود دارند که متناظر کردن آنها بسیار دشوار است. از سوی دیگر، پرس و جوی گراف ها به طور فراینده ای در حال پیچیده شده است. به عنوان مثال، پرس و جوهایی که در برابر هستی شناسی پیچیده قرار دارند اغلب طولانی هستند، فارغ از اینکه کدام زبان پرس و جوی گراف برای ارئه پرس و جو مورد استفاده قرار می گیرد. علاوه بر این، در صورت وجود یک گراف پیچیده (مانند هستی شناسی پیچیده)، کاربرها اغلب به جای درک و تعریف درست و روشنی از آنچه که درباره آن به پرس و جو پرداخته اند. فقط ایده ای مبهم وگنگ از این موضوع دارند. این ها موجب فراخوانی روش های جایگزین برای پردازش و بیان پرس و جوهای گراف می شود. به عبارت دیگر، به جای بیان شفاف پرس و جو ها در دقیق ترین چارچوب، ممکن است بخواهیم از جستجوی واژگان کلیدی برای تسهیل پرس و جوها [۱۸۳]، یا بهره گیری از روش های استخراج داده برای تشکیل پرس و جوی نیمه خودکار استفاده کنیم [۱۳۴].

۲,۲. پرس و جوهای دسترسی

پرس و جوهای دسترسی گراف بررسی می کند که آیا مسیری از گره V به گره u در یک گراف مستقیم بزرگ وجود دارد یا نه. پرس و جو برای دسترسی گراف یک عملیات بنیادی است که برای بسیاری از برنامه های کاربردی از جمله برنامه های حوزه ی شبکه معنایی (سمنتیک وب)، شبکه های زیستی، پردازش پرس و جوهای XML و غیره اهمیت زیادی دارد.

پرس و جو های دسترسی را می توان به دو روش روشن و مشخص پاسخ داد. در روش اول، با استفاده از جستجوی عرض ـ محور یا عمق ـ محور و با شروع از گره V، از گراف عبور می کنیم تا مشاهده کنیم که آیا قادر به دسترسی به گره u هستیم یا نه. زمان پرس و جو برابر با O(n+m) است، که n تعداد گره ها و m تعداد کران های موجود در گراف است. در آن سو، بن بست گذرای کران در گراف را محاسبه و ذخیره می کنیم. با بن بست گذرا، که مستلزم ذخیره ی $O(n^2)$ است، پرس و جوی دسترسی را می تواند در زمان O (۱) و از طریق بررسی وجود (u,V) در بن بست گذرا پاسخ داد. با این وجود، در نمودارهای گسترده، هیچ یک از دو روش شدنی نیست: روش اول از نظر زمان پرس و جو بسیار پر هزینه است، و روش دوم نیازمند فضای بسیار زیادی است.

تحقیق در این حوزه بر یافتن بهترین میانگین بین زمان پرس و جو O(n+m) و هزینه ی ذخیره $O(n^2)$ تمرکز می کند. این تحقیق از راه شهودی تلاش می کند اطلاعات دسترسی را در بن بست کران فشرده کند و با استفاده از داده های فشرده به پرس و جو ها پاسخ دهد.

رویکردهای مبتنی بر نمودار درختی دربرگیرنده. رویکردهای بسیار، مانند [۱۸۴، ۱۷۶، ۴۷]، گراف را به دو بخش تقسیم می کنند: i) نمودار درختی در بر گیرنده، و ii) کران هایی که روی نمودار درختی قرار ندارند (کران های غیر درختی). اگر در نمودار درختی در برگیرنده، مسیری بین u و V وجود داشته باشد، دسترسی بین u و V قطعاً به سادگی می تواند باشد. با اختصاص یک کد فاصله ی (u_{start} و u_{end}) به هر گره u این کار امکان پذیر است، به طوری که V از u قابل دسترس باشد اگر و تنها اگر $u_{start} \leq V_{start} \leq u_{end}$ کل نمودار درختی را می توان با اجرای یک عبور عمق ـ محور ساده از نمودار درختی کد گذاری کرد. با انجام فرآیند کد گذاری، بررسی دسترسی در زمان O (۱) قابل اجرا خواهد بود.

اگر هیچ مسیری روی نمودار درختی در برگیرنده، دو گره را به هم متصل نکند، باید بررسی کنیم که آیا مسیری که کران های غیر درختی را در بر می گیرد، دو گره را به هم متصل می کند یا نه. به این منظور، باید ساختارهای شاخص به علاوه کد فاصله را ایجاد کنیم تا بررسی دسترسی را شتاب بیشتری ببخشیم. چن و دیگران [۴۷] و تیرمبل و دیگران [۱۷۶] ساختارهای شاخص برای این منظور معرفی کردن و هر دو رویکرد آنها به زمان پرس و جویی معادل O (m-n) دست یافت. به عنوان مثال. SSPI (شاخص جایگزین و مازاد پیشین) که توسط چن و دیگران ارائه شده فهرست پیشین PL (u) را برای هر گره u حفظ می کند، که همراه با کد فاصله موجب بررسی دسترسی کارآمد و مؤثر می شود. وانگ و دیگران [۱۸۴] این عقیده را ابراز کردند که بسیاری از گراف های گسترده در برنامه های کاربردی واقعی نا متراکم هستند، که این بدان معناست که تعداد کران های غیر درختی اندک است. الگوریتم پیشنهادی براساس این فرض با استفاده از ساختار شاخص اندازه ی O ($n + t^2$) به پرس و جوی دسترسی در زمان O (۱) پاسخ دادند، که t معادل تعداد کران های غیر درختی است، و $t \ll n$.

رویکردهای مبتنی بر پوشش مجموعه. چندین روش برای استفاده از ساختارهای ساده تر داده ها (مثلاً نمودارهای درختی، مسیرها و غیره) برای پوشش اطلاعات دسترسی که در قالب ساختار گراف بیان شده اند، معرفی می شود. به عنوان مثال، اگر V به تواند به u دسترسی پیدا کند، آنگاه V می تواند به هر یک از گره های نمودار درختی که از u منشأ گرفته، دسترسی پیدا کند. بنابر این، اگر نمودار درختی را در شاخص بگنجانیم آنگاه مجموعه بزرگی از دسترسی در گراف را تحت پوشش قرار داده ایم. سپس، از نمودار درختی چندگانه برای پوشش یک گراف کامل استفاده می کنیم. پوشش نمودار درختی مطلوب آگراوان و دیگران [۱۰] به زمان پرس و جوی O (logn) دست پیدا می کند، که n برابر با تعداد گره های موجود در گراف است. چاگاریش و دیگران [۱۰۵] به جای استفاده از نمودار های درختی پیشنهاد کردند که گراف ها به زنجیره های اتصال جفت محور تقسیم شوند، و سپس از زنجیره ها برای پوشش گراف استفاده شود. ایده استفاده از زنجیره شبیه به استفاده از نمودار درختی است: اگر V بتواند روی یک زنجیره به u دسترسی پیدا کند، آنگاه V می تواند به هر یک از گره هایی که پس از u روی زنجیره قرار می گیرند دسترسی پیدا کند. روش پوشش ـ زنجیره ای به زمان پرس و جوی O (nk) دست می یابد، که k برابر با تعداد زنجیره های موجود در گراف است. کوهن و دیگران [۵۴] یک پوشش ۲ـ مرحله ای برای پرس و جوهای دسترسی پیشنهاد کردند. گره u به وسیله ی دو مجموعه از گره ها

به نام $L_{in}(u)$ و $L_{out}(u)$ بر چسب زده می شوند، که $L_{in}(u)$ معادل گره هایی است که می تواند به u دسترسی داشته باشد و $L_{out}(u)$ معادل گره هایی است که u می تواند به آنها دسترسی پیدا کند. روش ۲ـ مرحله ای، برچسب های L_{in} و L_{out} را به هر یک از گره ها نسبت می دهد به گونه ای که u بتواند به V دسترسی پیدا کند اگر و تنها اگر $L_{out}(u) \cap L_{in}(v) \neq \emptyset$ مسئله ی پوشش ۲ـ مرحله ای مطلوب برای یافتن اندازه مینیمم پوشش ۲ـ مرحله NP دشوار است. یک الگوریتم سخت گیر، پوشش دو مرحله ای را متناوباً پیدا می کند. در هر تناوب، گره ω را انتخاب می کند که مقدار $\frac{S(A_\omega,\omega,D_\omega)\cap TC'}{|A_\omega|+|D_\omega|}$ را به حداکثر می رساند، در حالیکه $S(A_\omega,\omega,D_\omega) \cap TC'$ بیانگر دسترسی (غیر پوششی) جدیدی است که خوشه ی ۲ـ مرحله ای متمرکز در ω قابل پوشش است، و $|A_\omega| + |D_\omega|$ برابر با اندازه خوشه ۲ـ مرحله ای متمرکز در ω است. چندین الگوریتم برای ارزیابی کارآمد پوشش های ۲ـ مرحله ای با کیفیت [۴۸،۴۹،۱۶۸،۵۴] ارائه شده اند. تعمیم های بسیاری برای روش های مبتنی بر پوشش مجموعه پیشنهاد شده اند. به عنوان مثال، جین و دیگران [۱۱۲] یک روش پوشش ۳ـ مرحله ای معرفی کردند که پوشش زنجیره را با پوشش در هم ادغام می کند.

تعمیم هایی به مسئله ی دسترسی. پرس و جو های دسترسی یکی از اساسی ترین بخش های سازنده بسیاری از فرآیندهای عملیاتی پیشرفته گراف هستند، و بعضی از آنها مستقیماً با پرس و جو های دسترسی ارتباط دارند. حوزه گراف های برچسب دار یکی از مسائل جالب است. در بسیاری از برنامه های کاربردی، به کران ها برچسب زده می شود تا بیانگر رابطه ی بین دو گره باشد که به وسیله ی کران به هم متصل شده اند. گونه جدیدی از پرس و جو های دسترسی این پرسش را مطرح می کند که آیا دو گره به وسیله ی مسیری که کران های آن با مجموعه ی معینی از برچسب ها محدود شده، به هم پیوسته اند [۱۱۱]. در بعضی از برنامه های کاربردی، می خواهیم کوتاه ترین مسیر بین دو گره را پیدا کنیم. مسئله ی کوتاه ترین مسیر نیز همانند مسئله ی دسترسی آسان می تواند از طریق روش های نیروی اجباری، مثل الگوریتم دیجکسترا، مرتفع شود اما اینگونه روش ها برای پرس و جو های آن لاین در گراف های گسترده و بزرگ مناسب به نظر نمی رسند. کوهن و دیگران یک روش پوشش ۲ـ مرحله ای برا این مسئله ارائه کردند [۵۴].

شرح مفصل نقاط قوت و نقاط قوت و نقاط ضعف روش های دسترسی مختلف مقایسه ی زمان پرس و جوی آنها، اندازه ی شاخص، و زمان ایجاد شاخص را می توانید در [۲۰۴] مشاهده کنید.

۲,۳. تناظر گراف

مسئله تناظر گراف، پیدا کردن تناظر تقریبی یا یک به یک بین گره ها دو گراف است. این تناظر براساس یک یا چند ویژگی ساختاری زیر در گراف ها استوار است: (۱) بر چسب های روی گره ها در دو گراف می بایست مشابه باشند. (۲) حضور کران ها بین گره های متناظر در دو گراف می بایست متناظر باشند. (۳) بر چسب های روی کران ها در دو گراف باید متناظر باشند.

این سه ویژگی ممکن است برای تعیین تناظر بین دو گراف مورد استفاده قرار بگیرند به طوری که تناظر یک به یک بین ساختارهای دو گراف وجود داشته باشد. اینگونه مسائل، اغلب در چارچوب تعداد برنامه های مختلف پایگاه داده ها از قبیل تناظر چارچوب کلی، تناظر پرس و جو و جاسازی فضای بردار ظاهر شود. شرح مفصل این برنامه های کاربردی مختلف در [۱۶۱] قابل مشاهده است. در تناظر گراف دقیق تلاش می کنیم تناظر یک به یک بین دو گراف را مشخص کنیم. بنابراین، اگر یک کران بین یک جفت گره در یک گراف وجود داشته باشد، آنگاه آن کران باید بین جفت متناظر در گراف دیگر نیز وجود داشته باشد. این پدیده در برنامه های کاربردی واقعی که در آنها تناظر تقریبی وجود دارد ولی تناظر دقیق شدنی به نظر نمی رسد، چندان عملی نخواهد بود. از این رو، در بسیاری از برنامه های کاربردی، تعریف یک تابع واقعی که تشابه در تناظر بین دو گراف را تعیین کند، امکان پذیر است. تناظر خطای مجاز یک برنامه کاربدی مهم در حوزه ی گراف است، زیرا باز نمایی عمومی گراف ها ممکن است دارای گره ها و کران های گمشده ی بسیاری باشد. این مسئله به عنوان تناظر گراف غیر دقیق نیز شناخته می شود. اکثر متغیرهای مسئله تناظر گراف به عنوان هارد تفکیک نشده[۱] در نظر گرفته می شوند. متداول ترین روش برای تناظر گراف مربوط به تکنیک ها ی جستجوی مبتنی بر نمودار درختی است. در این تکنیک، از مجموعه ی مناسب کشت گره های متناظر شروع می کنیم و به طور متناوب مجاور تعیین شده توسط همان مجموعه را تعمیم می دهیم. تعمیم متناوب، از طریق افزون گره ها به مجموعه گره کنونی قابل اجرا است تا زمانی که هیچ یک ازمحدودیت های کران نقض نشود. اگر مشخص شود که مجموعه گره کنونی قابل تعمیم نیست، آنگاه یک پروسه عقب نشینی را آغاز می کنیم که آخرین مجموعه تناظر را لغو کنیم. تعدادی از الگوریتم هایی که بر مبنای این ایده فراگیر طراحی شده اند در [۱۸۰،۱۲۵،۶۰] مورد بررسی قرار گرفته اند. ارزیابی بسیاری از الگوریتم های سنتی برای تناظر گراف در [۵۷] قابل مشاهده است.

مسئله تناظر دقیق گراف ارتباط نزدیکی با مسئله هم ریختی گراف دارد. در مورد مسئله هم ریختی گراف، تلاش می کنیم تناظر یک به یک دقیقی بین گره ها و کران های دو گراف پیدا کنیم. جمع بندی این مسئله مربوط به یافتن زیر گراف مشترک بیشینه است که در آن سعی می کنیم تعداد بیشینه گره ها بین دو گراف را با هم تطبیق دهیم. توجه داشته باشید که راه حل مسئله زیرگراف (گراف فرعی) مشترک بیشینه می تواند راه حلی نیز برای مسئله تناظر دقیق بین دو زیر گراف ارائه کند، البته در صورتی که اصولاً چنین راه حلی وجود داشته باشد. تعدادی از معیارهای تشابه را می توان بر اساس رفتار تناظر بین دو گراف استخراج کرد. اگر دو گراف مشترکاً در تعداد زیادی از گره ها سهیم باشند، آنگاه تشابه چشمگیرتر است. تعدادی از الگوریتم ها و مدل ها برای تعیین و تشخیص زیرگراف های مشترک بین دو گراف در [۳۷ ـ۳۴] قابل مشاهده اند. ایده اصلی و فراگیر در بسیاری از این روش ها تعیین یک معیار فاصله براساس ماهیت تناظر بین دو گراف و استفاده از این معیار فاصله به منظور هدایت الگوریتم ها به سمت راه حلی مؤثر است.

[۱] NP-Hard

تناظر غیر دقیق گراف عملی تر و اجرایی تر است زیرا خطاهای طبیعی که ممکن است در طی فرآیند تناظر اتفاق بیفتد را توجیه می کند. بدیهی است که روشی برای تعیین این خطاها و نزدیکی بین گراف های مختلف ضروری است. تکنیک مشترکی که برای تعیین این خطا مورد استفاده می گیرد، استفاده از تابعی مثل فاصله ویرایش گراف است. این تابع فاصله بین دو گراف را با اندازه گیری میزان ویرایش های لازم برای تبدیل یک گراف به گرافی دیگر تعیین می کند. این ویرایش ها (اصلاحات) ممکن است در قالب افزودن، حذف یا جایگزینی گره ها یا کران ها صورت بگیرد. تناظر غیر دقیق گراف، امکان تناظر بین دوگراف پس از یک سلسله از این ویرایش ها را به وجود می آورد. کیفیت و حالت تناظر با توجه به اندازه ویرایش های متناظر تعیین می شود. لازم به ذکر است که مفهوم فاصله ویرایش گراف کاملاً وابسته به یافتن زیر گراف مشترک بیشینه است [۳۴]؛ زیرا این امکان وجود دارد که یک الگوریتم مبتنی بر ویرایش، فاصله را به سمت پیدا کردن زیر گراف مشترک بیشینه از طریق تعیین فاصله ویرایش مناسب هدایت کنیم.

گونه خاص مسئله زمانی است که مقادیر بر چسب های روی گره ها و کران ها را در طی فرآیند تناظر پیدا می کنیم. در این حالت، لازم است فاصله بین برچسب های گره ها و کران ها را به منظور تعیین میزان جایگزینی برچسب محاسبه کنیم. بدیهی است که میزان جایگزینی برچسب وابسته به برنامه کاربردی مورد نظر است. در صورت وجود برچسب های عددی، ممکن است تعیین فاصله بر مبنای توابع فاصله ای عددی بین دو گراف طبیعی به نظر برسد. به طور کلی، میزان ویرایش ها نیز به برنامه کاربردی بستگی دارد، چرا که برنامه های مختلف ممکن است از مفاهیم مختلف شباهت استفاده کنند. بنابراین، از تکنیک های مشخص شده بر اساس قلمرویی خاص به منظور تعیین میزان ویرایش ها استفاده می شود. (بعضی از موارد، میزان ویرایش حتی ممکن است با استفاده از گراف های نمونه تعیین شود [۱۴۴ و ۱۴۳] وقتی با مواردی مواجه می شویم که در آنها گراف های نمونه به طور طبیعی فاصله بین آنها را مشخص می کنند، اندازه و میزان ویرایش به عنوان مقادیری تعیین می شوند که فاصله های تناظر تا جایی که امکان دارد به مقادیر نمونه نزدیکند.

الگوریتم های متعارف برای تناظر غیر دقیق گراف از جستجوی ترکیبی و تلفیقی در فضای ویرایش های ممکن استفاده می کنند به این منظور که تناظر مطلوب را مشخص کند [۱۴۵، ۳۵]. الگوریتم معرفی شده در [۳۵] در رویکرد خود نسبتاً فراگیر و جامع است، و به همین دلیل می تواند در عمل به لحاظ محاسباتی کامل و متمرکز باشد. به منظور رفع این مسئله، الگوریتم های بررسی شده در [۱۴۵]، مناطق محلی گراف ها را شناسایی می کنند تا ویرایش های متمرکز تر را تعیین کنند. به ویژه، کار ارائه شده در [۱۴۵] یک گروه مهم از روش ها که به عنوان توابع اصلی شناخته می شوند را پیشنهاد می کند. اینگونه روش ها به شدت در مقابل خطاهای ساختاری مقاومند و به همین دلیل طرحی مفید برای حل مسائل تناظر گراف هستند. ایده کلی ادغام ایده های کلیدی و مهم فاصله ویرایش گراف در توابع اصلی است. از آنجایی که توابع اصلی (کرنل) به عنوان تکنیک هایی به شدت قدرتمند برای شناسایی الگو هستند، اینطور نتیجه گیری می شود که این تکنیک ها را می توان به مسئله تناظر گراف الحاق کرد. انواع مختلفی از سایر تکنیک های اصلی برای تناظر گراف در [۱۱۹، ۸۱، ۹۴]

قابل مشاهده است. روش های اصلی کلیدی عبارتند از: توابع اصلی دشواری و پیچیدگی [۹۴]، توابع اصلی عبور تصادفی [۸۱] و توابع اصلی انتشار [۱۱۹] در توابع اصلی عبور تصادفی [۸۱]، تلاش می کنیم تعداد عبورهای تصادفی بین دو گراف که دارای چند برچسب مشترک هستند را تعیین کنیم. توابع اصلی انتشار [۱۱۹] را می توان به عنوان جمع بندی تابع Gaussim در فضای اقلیدسی در نظر گرفت.

تکنیک بر چسب گذاری تحقیقی (کاهشی) یک گروه بسیار گسترده از روش هایی است که اغلب برای تناظر نمودار مورد استفاده قرار می گیرد. توجه داشته باشید که در مورد مسئله تناظر، ما واقعاً تلاش می کنیم برچسب ها را به گره های هر گراف نسبت دهیم. بر چسب ویژه برای یک گره از مجموعه ناپیوسته احتمالات استخراج می شود. این مجموعه ناپیوسته از احتمالات با گره های تطبیق دهنده در سایر گراف ها متناظر است. احتمال تناظر با استفاده از پراکندگی های احتمال Gaussian تعیین می شود. با برچسب گذاری اولیه بر مبنای ویژگی های ساختاری گراف اصلی شروع می کنیم و پس از آن به طور پی در پی پاسخ را بر اساس شناسایی تکمیلی اطلاعات ساختاری اصلاح می کنیم. توضیح مفصل تکنیک های بر چسب گذاری تخفیفی در [۷۶] قابل مشاهده است.

۲,۴. جستجوی واژگان کلیدی

در مسئله جستجوی واژگان کلیدی، ممکن است بخواهیم گروه های کوچکی از گره های دارای لینک های پیوسته را تعیین کنیم که به واژگان کلیدی خاص وابسته اند. به عنوان مثال، یک گراف وب یا شبکه اجتماعی ممکن است به عنوان یک گراف گسترده در نظر گرفته شود که در آن هر گره ممکن است حاوی مقادیر زیادی از داده های قنی باشد. به رغم اینکه جستجوی واژگان کلیدی با توجه به متن درون گره ها تعیین می شود، لازم به ذکر است که ساختار اتصال نیز نقش مهمی در تعیین مجموعه مناسبی از گره ها بازی می کند. به خوبی می دانیم که متن در واحدهای متصل، مثل وب، به هم مرتبط هستند در هنگامی که موضوعات متناظر به هم متصل اند. بنابراین، با پیدا کردن گروه هایی از گره های کاملاً پیوسته که دارای واژگان کلیدی مشترکی هستند، به طور کلی تعیین گره هایی که به لحاظ کیفی مؤثر و کارآمد باشند امکان پذیر است. جستجوی واژگان کلیدی یک سطح مشترک ساده اما کاربر پسند برای بازیابی اطلاعات روی وب ارائه می دهد. همچنین، اثبات شده است که روشی کار آمد برای دسترسی به داده های ساختاری است. از آنجایی که بسیاری از مجموعه داده های واقعی به شکل جدول ها، نمودارهای درختی و گراف ها سازماندهی شده اند. جستجوی واژگان کلیدی در چنین داده هایی دارای اهمیت فزاینده ای است و توجه زیادی به تحقیق در زمینه ی پایگاه داده ها و جوامع IR معطوف کرده است.

گراف یک ساختار عمومی است و می توان آن را برای طراحی انواعی از داده های پیچیده و دشوار مانند داده های ارتباطی و داده های XML مورد استفاده قرار داد. از آنجایی که داده های اصلی به عنوان یک ساختار گراف

تلقی میشوند، جستجوی واژگان کلیدی دشوارتر و پیچیده تر از جستجوی سنتی واژگان کلیدی در اسناد و مدارک است. چالش مذکور از سه منظر زیر قابل بررسی است:

- **معنای پرس و جو:** جستجوی واژگان کلیدی در یک مجموعه از اسناد متنی معنای روشنی دارد: یک سند در صورتی پاسخگوی پرس و جوی واژگان کلیدی است که تمام واژگان کلیدی موجود در پرس و جو را در بر بگیرد. در موردی که ما به آن پرداخته ایم، کل مجموعه داده ها اغلب به عنوان یک گراف مجزا در نظر گرفته می شود بنابراین الگوریتم ها می بایست در یک زمینه ی بهتر کار کنند و زیر گراف ها را به عنوان پاسخ باز گردانند. باید تصمیم بگیریم که کدام زیر گراف ها حائز شرایطی هستند که به عنوان پاسخ در نظر گرفته شوند.
- **استراتژی رتبه بندی:** در مورد یک پرس و جوی واژگان کلیدی مشخص، این احتمال وجود دارد که بسیاری از گراف فرعی بر مبنای معنای پرس و جوی مورد استفاده خود پاسخگوی پرس و جو باشند. با این وجود، هر گراف فرعی دارای ساختار گراف اصلی مختص به خود است با معنای نامشهودی که آن را از سایر گراف های فرعی متمایز می کند و پرس و جو را پاسخ می گوید از این رو، باید ساختار گراف را در نظر بگیریم و استراتژی های رتبه بندی طراحی کنیم که معنادارترین و به جا ترین پاسخ را پیدا کند.
- **کارایی پرس و جو:** اکثر گراف های واقعی به شدت گسترده و بزرگند. کارایی پرس و جو، چالش عمده برای جستجوی واژگان کلیدی در داده های گراف است که تا حد زیادی منوط به معنای پرس و جو و استراتژی رتبه بندی است.

رویکردهای کنونی برای جستجوی واژگان کلیدی بر اساس ساختار اصلی داده ها به سه گروه تقسیم می شوند. در هر گروه معنای پرس و جو، استراتژی های رتبه بندی و الگوریتم های باز نما (نمونه) را به طور خلاصه مورد بررسی قرار می دهیم.

جستجوی واژگان کلیدی در داده های XML. داده های XML عمدتاً دارای ساختار گراف هستند، که هر گروه فقط یک مسیر جدید مجزا دارد. این ویژگی تأثیر شگرفی بر معنای پرس و جو و رتبه بندی پاسخ دارد، و فرصت بهینه سازی فوق العاده ای در طراحی الگوریتم ارائه می دهد. [۱۹۷].

با تعیین یک پرس و جو که شامل مجموعه ای از واژگان کلیدی است، الگوریتم جستجو قطعه هایی از اسناد XML را می فرستد که مرتبط ترین قطعه ها به واژگان کلیدی هستند. تفسیر "مرتبط و متناسب" متغیر است اما متداول ترین روال، پیدا کردن کوچکترین نمودارهای درختی فرعی است که شامل واژگان کلیدی مورد نظر باشند.

پیدا کردن نمودارهای درختی فرعی که شامل تمام واژگان کلیدی باشند آسان است. فرض کنید L_i مجموعه از گره ها در سند XML باشد که شامل واژگان کلیدی K_i است. اگر یکی از گره های n_i را از هر L_i برداریم و نمودار

درختی فرعی از این گره ها تشکیل دهیم، آنگاه نمودار درختی فرعی شامل تمام واژگان کلیدی خواهد بود. بنابراین، پاسخ به پرس و جو را می توان از طریق Lca (n_i, ... , n_n)، غیر متداول ترین شکل های قبلی گره ها n_i, ... , n_n در نمودار درختی بیان کرد، که در آن $n_i \epsilon L_i$.

اکثر معنی های پرس و جو فقط به کوچکترین پاسخ ها توجه دارند. شیوه های گوناگونی برای تفسیر مفهوم "کوچکترین" وجود دارد. چندین الگوریتم [۱۹۶، ۱۰۲، ۱۹۷] بر مبنای معنای SLCA (کوچکترین شکل های قبلی غیر متداول) طراحی شده اند که نیازمند آن است که یک پاسخ (دست کم شکل قبلی متداول گره هایی که شامل تمام واژگان کلیدی هستند) دارای هیچ زاده ای نباشد که خود به صورت پاسخ ظاهر شود. XRank [۸۶] معنای پرس و جوی متفاوتی برای جستجوی واژگان کلیدی انتخاب می کند. در XRank، پاسخ متشکل از نمودارهایی درختی فرعی است که شامل حداقل یک پیشامد از تمام واژگان کلیدی پرس و جو باشد پس از خارج کردن زیر گره هایی که از قبل شامل تمام واژگان کلیدی پرس و جو بوده اند. بنابراین، مجموعه ی پاسخ ها بر اساس معنای SLCA، زیر مجموعه ای پاسخ های واجد شرایط بریا XRank است.

پرس و جوی واژه کلیدی ممکن است تعداد زیادی پاسخ پیدا کند، اما با توجه به تفاوت در شیوه جاسازی آنها در ساختار XML همه پاسخ ها مساوی و هم ارز نیستند. بسیاری از رویکردها برای جستجوی واژه کلیدی در داده های XML، مثل XRank [۸۶] و XSEarch [۵۵]، روش رتبه بندی ارائه می دهند. مکانیسم های رتبه بندی چندین فاکتور را در نظر می گیرد. به عنوان مثال، پاسخ های صریح تر در مقایسه با پاسخ هایی که صراحت و دقت کمتری دارند باید در رتبه بالاتری قرار بگیرند. SLCA و معنی هایی که توسط XRank برگزیده شده اند بر این تفکر دلالت دارند. علاوه بر این، واژگان کلیدی در یک پاسخ باید کاملاً نزدیک به یکدیگر ظاهر شوند و نزدیکی به صورت فاصله معنایی تعیین شده در ساختار جاسازی شده XML تفسیر می شود.

جستجوی کلیدی در داده های ارتباطی. SQL یک زبان پرس و جوی بالفعل برای دسترسی به داده های ارتباطی است. با این وجود، برای استفاده از SQL می بایست اطلاعاتی درباره ی چارچوب کلی داده های ارتباطی داشته باشیم. این به مانعی بر سر راه کاربران بالقوه برای دسترسی به مقادیر قابل توجهی از داده های ارتباطی تبدیل شده است.

جستجوی واژگان کلیدی با توجه به سهولت استفاده می تواند جایگزین خوبی باشد. چالش های استفاده از جستجوی واژگان کلیدی در داده های ارتباطی ناشی از این واقعیت است که در پایگاه داده های ارتباطی، اطلاعات درباره یک واحد مجزا معمولاً میان چندین برچسب مختلف تقسیم می شود. این امر ناشی از اصل عادی سازی است که روش طراحی چارچوب کلی پایگاه داده های ارتباطی است.

بنابر این، برای پیدا کردن واحدهایی که با پرس و جوی واژه کلیدی مرتبط باشند. الگوریتم جستجو باید داده ها را از جدول های چندگانه به هم متصل کند. اگر هر جدول را به شکل یک گره بیان کنیم، و هر ارتباط کلید خارجی به صورت یک کران بین دو گره تعریف شود، آنگاه گرافی به دست می آوریم که به ما اجازه می دهد

مسئله فعلی را به مسئله جستجوی واژگان کلیدی در گراف ها تبدیل کنیم. با این وجود، احتمال خود اتصالی نیز وجود دارد: یعنی، ممکن است یک جدول شامل یک کلید خارجی باشد که خودش را به عنوان مرجع تلقی کند. به نحو گسترده تری، ممکن است حلقه هایی در گراف وجود داشته باشد که به این معناست که فقط اندازه داده می تواند اندازه اتصال را محدود کند. برای پرهیز از این مسئله، الگوریتم جستجو ممکن است از یک محدوده بالایی برای محدود کردن تعداد اتصال ها استفاده کند [۱۰۳].

شناخته شده ترین الگوریتم های جستجوی واژگان کلیدی برای داده های ارتباطی عبارتند از DBX-Plover [۱۲] و DISCOVER [۱۰۳]. این دو الگوریتم از پایگاه داده های فیزیکی جدیدی در پایگاه داده استفاده می کنند. کین و دیگران [۱۵۵]، در عوض، روشی که مزیت قدرت ADBMS را دارد و از SQL برای اجرای جستجوی واژگان کلیدی در داده های ارتباطی بهره می گیرد، پیشنهاد کردند.

جستجوی واژگان کلیدی در داده های گرافی. جستجوی واژگان کلیدی در گراف های بزرگ فاقد چارچوب با این چالش مواجه است که چگونه در ساختار گراف گردش کند و گراف های فرعی را پیدا کند که حاوی تمام واژگان کلیدی موجود در پرس و جو باشند. برای ارزیابی "کیفیت خوب" یک پاسخ، اکثر رویکردها به همه کران ها و گره ها امتیاز می دهند، و سپس امتیازات را به عنوان مقیاس کیفیت خوب در گراف فرعی جمع بندی می کنند [۹۹، ۱۱۳، ۲۴]. به طور معمول، هر کران با توجه به قدرت پیوستگی و هر گره با توجه به اهمیت آن بر اساس مکانیسم Page Rank امتیاز می گیرد.

الگوریتم های جستجوی واژگان کلیدی گراف را می توان به دو گروه طبقه بندی کرد. الگوریتم های گروه اول، گراف های فرعی تناظر را با بررسی لینک به لینک گراف پیدا می کند، بدون اینکه از هیچ یک از شاخص های گراف استفاده کند. الگوریتم های نمونه در این گروه عبارتند از: BANKS [۲۴] و الگوریتم جستجوی دو سویه [۱۱۳] یکی از نقطه ضعف های این رویکردها این است که آنها الگوریتم ها را کورکورانه بررسی می کنند زیرا فاقد یک تصویر کلی از ساختار گراف هستند و از پراکندگی واژگان کلیدی در گراف اطلاعی ندارند. الگوریتم های گروه دوم شاخص ـ محور [۹۹] هستند، و از شاخص برای جهت دادن به بررسی گراف و پشتیبانی از جهش های رو به جلو در جستجو استفاده می کنند.

۲,۵. خلاصه سازی گراف های بزرگ

یک چالش کلیدی که در بسیاری از برنامه های کاربردی که در ادامه مورد بررسی قرار می گیرند، پدید می آید این است که گراف هایی که با آنها سرو کار داریم بسیار بزرگند. در نتیجه، دسترسی به این گراف ها فقط روی دیسک ممکن خواهد بود. اکثر برنامه های استخراج گراف سنتی فرض را بر این می گذارند که داده ها روی حافظه اصلی قرار دارند. با این وجود، وقتی گراف روی دیسک ذخیره شده باشد، برنامه هایی که دسترسی تصادفی به کران ها دارند احتمالاً به شدت گران خواهند بود. به عنوان مثال، مسئله پیدا کردن قطعه ـ مینیمم

در الگوریتم های روی حافظه اصلی بسیار کار آمد است، اما در صورتی که گراف های اصلی روی دیسک ذخیره شده باشند به طور غیر منتظره ای پرهزینه خواهد بود [۷]. در نتیجه باید الگوریتم ها را به دقت طراحی کرد تا هزینه های دسترسی به دیسک کاهش یابد. یک تکنیک معمول که اغلب مورد استفاده قرار می گیرد طراحی یک تکنیک خلاصه سازی است [۱۴۲، ۴۶، ۷]، که گراف را در یک فضای بسیار کوچکتر فشرده می کند، اما اطلاعات کافی را به منظور پاسخ مؤثر به پرس و جو ها حفظ می کند.

خلاصه سازی به طور معمول از طریق تضادهای گره یا کران تعیین می شود. نکته کلیدی، تعیین خلاصه ای است که تمام ویژگی های ساختاری مرتبط و متناسب گراف اصلی را حفظ کند. در [۷]، الگوریتم معرفی شده در [۱۷۷] به منظور تجزیه نواحی متراکم گراف و ارائه گراف خلاصه شده در قالب نواحی نا متراکم مورد استفاده قرار می گیرد. گراف فشرده حاصل تمام ویژگی های ساختاری مهم از جمله قابلیت اتصال گراف را حفظ می کند. در [۴۶]، یک تکنیک خلاصه سازی تصادفی به منظور تعیین الگوهای پرتکرار در گراف اصلی مورد استفاده قرار می گیرد. در [۴۶]، محدوده ای برای تعیین مثبت های نادرست و منفی های نادرست حاصل از این رویکرد پیشنهاد شده است. در نهایت، تکنیک ارائه شده در [۱۴۲] نیز با بیان مجموعه گره ها به عنوان سوپرگره ها و ذخیره جداگانه "اصلاحات کران" به منظور بازسازی کل گراف، اقدام به فشرده سازی گراف می کند. یک محدوده خطای مجاز برای استفاده از این رویکرد در [۱۴۲] ارائه شده است.

یک مسئله کاملاً مرتبط در این زمینه، استخراج زنجیرهای گراف است. در این حالت، کرانهای گراف به طور مداوم در طول زمان به دست می آیند. این گونه موارد به طور مکرر در برنامه هایی از قبیل شبکه های اجتماعی، شبکه های ارتباطی، و تحلیل لوگ شبکه به چشم می خورد. استخراج زنجیره های گراف به شدت دشوار و چالشی است، زیرا ساختار گراف باید بلادرنگ استخراج شود. از این رو، یک رویکرد معمول ایجاد یک خلاصه از زنجیره گراف و ذخیره آن به منظور تحلیل ساختاری است. در [۷۳] نشان داده شده است که چگونه گراف را به شیوه ای خلاصه سازی کنیم که فاصله های اصلی حفظ شوند. بنابراین، از این خلاصه سازی می توان برای برنامه های کاربردی فاصله ـ محور مثل مسئله کوتاه ترین مسیر استفاده کرد. برنامه کاربردی دوم که در چارچوب زنجیره های گراف مورد مطالعه قرار گرفته است مربوط به تناظر گراف است [۱۴۰]. لازم به ذکر است که این نسخه متفاوتی از مسئله ای است که در بخش های پیشین مورد بحث قرار گرفت. در این شرایط، تلاش می کنیم مجموعه ای از کران ها در یک گراف مجزا را پیدا کنیم به طوری که هیچ دو کرانی دارای نقطه پایانی مشترک نباشند. مطلوب است مقدار پیشینه یا تناظر عددی بیشینه را پیدا کنیم. ایده اصلی در [۱۴۰] حفظ همیشگی یک تناظر داوطلب و به روز کردن آن همزمان با ورود کران های جدید است. وقتی یک کران جدید وارد می شود. فرآیند اضافه کردن آن ممکن است به اندازه کران در نقاط پایانی آن جابه جایی به وجود آورد. اجازه می دهیم یک کران تازه وارد در نقاط پایانی خود کران ها را جابه جا کند، در صورتی که مقدار کران تازه وارد یک فاکتور (γ +۱) از کران های رفتنی باشد. در [۱۴۰] نشان داده شده است که این تناظر درون فاکتور(+ 3 2 0√2) از تناظر مطلوب قرار دارد.

اخیراً، بعضی از تکنیک ها نیز برای ایجاد خلاصه هایی که بتوان برای ارزیابی ویژگی های ساختاری تلفیقی گراف های اصلی از آنها استفاده کرد، طراحی شده اند. تکنیکی برای ارزیابی آمارهای رتبه های موجود در زنجیره گراف اصلی در [۶۱] ارائه شده است. تکنیک های پیشنهاد شده در [۶۱] از انواعی از تکنیک ها مثل ترسیم نمای کلی، نمونه برداری، شماره گذاری و محاسبه متمایز استفاده می کنند. روش هایی برای تعیین زمان هایی رتبه ها، تعیین رتبه های پر برخورد، و تعیین دامنه مجموع رتبه ها پیشنهاد شده است. به علاوه، تکنیک هایی در [۱۸] معرفی شده اند برای اینکه خلاصه های مکانی کارآمدی را در زنجیره های داده ها اجرا کنند. این خلاصه برای محاسبه مثلث ها در زنجیره داده ها مورد استفاده قرار گرفته است. یک برنامه کاربردی مفید در زنجیره گراف مربوطه به مسئله PageRank است. در این مسئله، تلاش می کنیم صفحه هایی مهم در مجموعه را با استفاده از ساختار اتصال اسناد اصلی تعیین کنیم. بدیهی است که اسنادی که به وسیله تعداد زیادی از اسناد به هم متصل شده اند اهمیت بیشتری دراند [۱۵۱]. در واقع، مفهوم رتبه صفحه را می توان به صورت احتمال مشاهده یک گره از طریق موج سوار تصادفی در شبکه جهانی وب طراحی کرد. الگوریتم های طراحی شده در [۱۵۱] برای گراف های استاتیک (ثابت) به کار می روند. وقتی گراف ها دینامیک باشند، مانند شبکه های اجتماعی، مسئله چالشی تر نیز خواهد شد. یک تکنیک خلاصه سازی طبیعی که از آن می توان برای چنین مواردی استفاده کرد روش نمونه گیری است. در [۱۶۶]، چگونگی استفاده از تکنیک نمونه برداری به منظور ارزیابی رتبه صفحه برای زنجیره های گراف نشان داده شده است. ایده اصلی این است که گره ها را در گراف به طور جداگانه نمونه برداری کنیم و فرآیند گردش تصادفی را با حرکت از این گره ها اجرا کنیم. این گردش های تصادفی را می توان به منظور برآورد احتمال حضور یک موج سوار تصادفی در یک گره مشخص مورد استفاده قرار داد. این پروسه لزوماً هم ارز با رتبه بندی صفحه است.

۳. الگوریتم استخراج گراف

اکثر برنامه های استخراج سنتی برای گراف ها نیز مورد استفاده قرار می گیرند. برنامه های کاربردی استخراج نیز مانند برنامه های کاربردی مدیریت به واسطه محدودیت های سنتی که از ماهیت ساختاری گراف اصلی نشأت می گیرد، برای اجرا شدن با چالش های فراوانی مواجهند، علیرغم این چالش ها، برخی تکنیک ها برای مسائل استخراج سنتی از قبیل استخراج الگوی پرتکرار، خوشه بندی و طبقه بندی الگوی پرتکرار طراحی شده اند. در این بخش، یک نمای کلی از بسیاری از الگوریتم های ساختاری استخراج گراف ارائه می دهیم.

۳,۱. استخراج الگو در گراف ها

مسئله استخراج الگوی پرتکرار در چارچوب استخراج داده های اجرایی به طور گسترده ای مورد مطالعه قرار گرفته است [۹۰و۱۱]. اخیراً تکنیک هایی برای استخراج الگوی پرتکرار برای داده های گراف نیز تعمیم داده

شده اند. تفاوت عمده در مورد گراف ها این است که فرآیند تعیین پشتیبان کاملاً متفاوت است. مسئله برحسب حوزه کاربردی آن به شیوه های مختلفی قابل تعریف است:

- در مورد اول، گروهی از گراف ها را در اختیار داریم و مایلیم تمام الگوهایی که بخشی از گراف های متناظر را تأیید می کنند تعیین کنیم [۱۸۱،۱۲۳،۱۰۴].
- در مورد دوم، یک گراف بزرگ مجزا داریم و مایلیم تمام الگوهایی که حداقل در چند نوبت معین در این گراف بزرگ مورد تأیید قرار می گیرند را تعیین کنیم [۱۲۳،۷۵،۳۱].

در هر دو مورد، لازم است در تعیین تأیید یک گراف توسط دیگری، موضوع هم ریختی را نیز در نظر بگیریم. با این وجود، مسئله تعیین پشتیبان چالشی تر است اگر همپوشی بین تثبیت کننده های مختلف امکان پذیر باشد. این موضوع به این خاطر است که اگر چنین همپوشی هایی را مجاز کنیم آنگاه ویژگی یکنواختی اکثر الگوریتم های استخراج الگوی پرتکرار نقض می شود.

برای مورد اول، جایی که مجموعه ای از داده ها شامل گراف های چند گانه داریم می توان اکثر داده ها را به سادگی تعمیم داد. به عنوان مثال، الگوریتم هایی به سبک آپریوری را می توان با استفاده از استراتژی طبقه ـ محور مشابهی برای ایجاد داوطلب های (K+۱) از الگوهای K به سادگی تعمیم داد. تفاوت عمده در این است که باید فرآیند اتصال را به شیوه ای متفاوت تعریف کنیم. اندازه این ساختار بر حسب گره ها یا کران قابل تعریف است. در مورد الگوریتم AGM [۱۰۴]، این ساختار عمومی بر حسب تعداد رأس های مشترک قابل تعریف خواهد بود. از این رو دو گراف با K رأس به هم متصل می شوند تنها اگر دارای یک گراف فرعی مشترک با دست کم (K_۱) رأس باشند.

راه دوم برای اجرای فرآیند استخراج، اتصال دو گراف است که دارای یک گراف فرعی هستند که دست کم شامل (K_۱) کران مشترک است. از الگوریتم FSG پیشنهاد شده در [۱۲۳] می توان به منظور اجرای اتصال کران ـ محور بهره گرفت. همچنین، تعیین اتصالات برحسب ساختارهای اختیاری نیز امکان پذیر است. به عنوان مثال، بیان گراف ها برحسب مسیرهای تجزیه ـ کران امکان پذیر است. در اینگونه موارد، گراف های فرعی با مسیرهای تجزیه کران (K+۱) را می توان از دو گراف که دارای K مسیر تجزیه کران هستند ایجاد کرد که از میان آنها (K_۱) مسیر باید مشترک باشند. یک الگوریتم در راستای این خطوط در [۱۸۱] پیشنهاد شده است. استراتژی دیگری از اغلب مورد استفاده قرار می گیرد مربوط به تکنیک های افزایش الگو است. که در آن الگوهای گراف پرتکرار با استفاده از کران های اضافی تعمیم داده می شوند [۱۰۰و ۲۰۰و۲۸]. مانند مسئله استخراج الگوی پرتکرار، از ترتیب واژگان نمایی میان کران ها استفاده می شود تا فرآیند جستجو سازماندهی شود به طوری که یک الگوی مشخص تنها یک بار رؤیت شود.

در مورد دوم که یک گراف بزرگ مجزا داریم، ممکن است از چند تکنیک مختلف برای تعیین پشتیبان در حضور همپوشی ها استفاده شود. یک استراتژی عمومی استفاده از اندازه مجموعه مستقل بیشینه ای از گراف های

همپوشان برای تعیین پشتیبان است. به این استراتژی پشتیبان مجموعه مستقل بیشینه نیز گفته می شود. در [۱۲۴]، دو الگوریتم HSIGRAM و VSIGRAM برای تعیین گراف های فرعی پرتکرار درون یک گراف بزرگ مجزا معرفی شدند. در مورد پیشین، از یک رویکرد جستجوی وسعت ـ محور به منظور تعیین گراف های فرعی پرتکرار استفاده می شود، در حالی که از یک رویکرد عمق ـ محور برای مورد دوم استفاده می شود. در [۷۵]، نشان داده شده است که مقیاس مجموعه مستقل بیشینه همچنان به تأمین ویژگی ضد یکنواختی ادامه می دهد. مسئله اصلی در مورد این مقیاس این است که محاسبه آن به شدت پرهزینه است. بنابراین، تکنیک معرفی شده در [۳۱] یک مقیاس متفاوت برای محاسبه پشتیبان الگو تعریف می کند. محاسبه پشتیبان تصویر ـ محور بیشینه یک الگوی مشخص مد نظر است. در این حالت، تعداد گره های منحصر بفرد گراف که یک گره از الگوی مشخص با آن متناظر می شود را محاسبه می کنیم. این مقیاس همچنان به تأمین ویژگی ضد یکنواختی ادامه می دهد و به همین دلیل می توان از آن به منظور تعیین الگوهای پرتکرار اصلی استفاده کرد. یک الگوریتم کارآمد که از این مقیاس استفاده می کند در [۳۱] پیشنهاد شده است.

درمورد استخراج الگوی پرتکرار استاندارد، تعدادی از متغیرها برای یافتن الگوهای گراف محتمل است که عبارتند از تعیین الگوهای بیشینه [۱۰۰]، الگوهای مسدود [۱۹۸]، یا الگوهای مهم [۱۹۸،۱۵۷،۹۸]. لازم به ذکر است که الگوهای گراف مهم را می توان بسته به کاربرد آن به شیوه های مختلفی تعیین کرد. در [۱۵۷]، گراف های مهم با تبدیل نواحی گراف ها به خاصیت ها و ارزیابی ارزش متناظر بر حسب مقادیر ـ p تعیین می شوند. در [۱۹۸]، الگوهای مهم برحسب توابع واقعی اختیاری تعیین می شوند. یک فرا چارچوب در [۱۹۸] معرفی شده است تا الگوهای مهم بر مبنای توابع واقعی اختیاری تعیین شوند. یکی از رویکردهای جالب برای شناسایی الگوهای مهم، ساخت نمودار درختی جستجوی مدل ـ محور یا mbT [۷۱] است. استفاده از تقسیم و دستیابی برای شناسایی الگوهای مهم، ساخت نمودار درختی جستجوی مدل ـ محور یا mbT [۷۱] است. استفاده از تقسیم و دستیابی برای استخراج مهم ترین الگوها در یک فضای فرعی از نمونه ها، ایده اصلی این رویکرد است. این رویکرد یک درخت تصمیم می سازد که داده ها را به گره های مختلف تقسیم بندی می کند. پس از آن در هر گره، مستقیماً یک الگوی متمایز برای تقسیم تکمیلی نمونه ها به زیر مجموعه های واضح تر و محض تر شناسایی می کند. از آنجایی که تعداد نمونه های نزدیک به سطح برگ نسبتاً کم است، این رویکرد قادر است الگوهای دارای پشتیبان کلی کاملاً ضعیف که روی مجموعه داده های کلی قابل شمارش نیستند را بررسی می کند. برای بعضی از مجموعه داده های گراف که در برنامه های کشف مواد مخدر وجود دارند [۷۱]، می تواند الگوهای گراف مهم را استخراج کنند که برای بسیاری از رویکرد های دیگر دشوار است. این الگوریتم به خاطر استفاده از الگوی تکنیک MbT به گراف ها محدود نمی شود بلکه برای مجموعه داده ها و توالی ها نیز قابل اجرا است و مجموعه الگوی استخراج شده کوچک و مهم است.

یکی از چالش های مهم که در چارچوب تمام الگوریتم های استخراج الگوی پرتکرار پدید می آید مربوط به تعداد زیاد الگوهایی است که می توان از پایگاه داده ها استخراج کرد. این مسئله به ویژه در مورد گراف ها

شدیدتر است زیرا اندازه خروجی می تواند به شدت بزرگ باشد. یک راه حل برای کاهش تعداد الگوهای بازنما (نمونه) اعلام الگوهای پرتکرار بر حسب راست گوشه بودن (قائم بودن) آنهاست. مدلی تحت عنوان ORIGAMI در [۹۳] ارائه شده است، که الگوهای گراف پرتکرار را تنها در صورتی اعلام می کند که شباهت کمتر از آستانه $\propto$ باشد. اینگونه الگوها به عنوان الگوهای قائم ـ $\propto$ نیز شناخته می شوند. مجموعه الگوی ρ در صورتی بازنمای ـ β گفته می شود که به ازای هر الگوی اعلام نشده g دست کم یک الگو در ρ پیدا شود که شباهت آن به g حداقل معادل آستانه β باشد. این دو محدودیت به بسیاری از جنبه های الگوهای ساختاری مورد توجه قرار می دهند. روش معرف شده در [۹۳] مجموعه ای از تمام الگوهای قائم ـ $\propto$ و باز نمای ـ β را مشخص می کند. در اینجا، کاهش موارد زائد در مجموعه ی الگوهای اصلی مد نظر است به این منظور که درک بهتری از الگوهای اعلام شده به دست آید.

بعضی از شکل های به ویژه چالشی مسئله در چارچوب مجموعه داده های بسیار بزرگ یا گراف های داده های بزرگ ظاهر می شود. اخیراً، تکنیکی توسط [۴۶] ارائه شد که از خلاصه سازی تصادفی به منظور کاهش مجموعه داده ها به یک اندازه ی بسیار کوچکتر استفاده می کند. سپس این خلاصه سازی برای تعیین الگوهای نمودار فرعی پرتکرار از داده ها به کارگرفته می شود. محدوده ها بر مبنای مثبت های نادرست و منفی های ناردست و با استفاده از چنین رویکردی در [۴۶] استخراج می شود، یک شکل چالشی دیگر زمانی ظاهر می شود که الگوهای پرتکرار در یک گراف خیلی بزرگ نادیده گرفته می شوند به این دلیل که این الگوها ممکن است خودشان گراف های فرعی بسیار بزرگی باشند. الگوریتمی به نام Tsminer در [۱۱۰] معرفی شد تا ساختارهای پرتکرار در گراف های بسیار بزرگ تعیین شوند.

استخراج الگوی گراف دارای کاربردهای فراوانی برای انواع گوناگونی از برنامه های کاربردی است. به عنوان مثال، در مورد داده های برچسب دار از اینگونه تکنیک های استخراج الگو می توان به منظور تعیین قوانین طبقه بندی ساختاری استفاده کرد. به عنوان مثال، تکنیک ارائه شده در [۲۰۵] از این رویکرد برای طبقه بندی داده های XML استفاده می کند. در این حالت، مجموعه داده های متشکل از گراف های (XML) چندگانه داریم که هر یک از آنها یک برچسب طبقه به همراه دارد. روش معرفی شده در [۲۰۵] قوانین و اصولی را تعیین می کند که ضلع سمت چپ معرف یک ساختار است و ضلع سمت راست معرف یک برچسب طبقه است. از این روش برای اهداف طبقه بندی استفاده می شود. یکی از دیگر کاربردهای استخراج الگوی پرتکرار در [۱۲۱] مورد مطالعه قرار گرفته است؛ که در آن الگوهای مذکور به منظور ایجاد gBoost مورد استفاده قرار می گیرند. که طبقه بندی کننده ای است که به عنوان یک برنامه کاربردی تقویت کننده طراحی شده است. اثبات شده است که استخراج الگوی پرتکرار به ویژه در حوزه داده های شیمیایی و زیست شناسی مفید است [۱۲۰،۱۰۱،۶۵،۲۸]. تکنیک های استخراج الگوی پرتکرار برای اجرای توابع مهم در این حوزه از جمله طبقه بندی و تعیین مسیرهای متابولیک مورد استفاده قرار گرفته اند.

استخراج الگوی گراف پرتکرار برای ایجاد شاخص های گراف نیز سودمند است. در [۲۰۱]، ساختارهای پرتکرار در مجموعه گراف استخراج می شوند به طوری که بتوان از آنها به عنوان خاصیت هایی برای فرآیند شاخص گذاری استفاده کرد. از تشابه رفتار عضویت الگوی پرتکرار در سراسر گراف ها برای تعیین یک تابع شباهت ابتدایی به منظور انجام فرآیند فیلترنیگ استفاده می شود. یک بازنمایی معکوس بر اساس این بازنمایی خاصیت ـ محور ساخته می شود به این منظور که گراف های نامربوط برای فرآیند جستجوی شباهت فیلتر شوند. تکنیک ارائه شده در [۲۰۱] به واسطه رویکرد خاصیت ـ محور خود کارآمدتر از معیار تکنیک های رقابتی است. به طوری که، الگوریتم های استخراجی الگوی پرتکرار برای هرگونه کاربردی که بر پایه ویژگی های تلفیقی به نحو مؤثر قابل تعیین باشد، مفید خواهند بود. در کل، تکنیک های استخراج الگوی گراف دامنه کاربردی مشابه با آنچه که در مورد استخراج الگوی پرتکرار عادی انجام می دهند، دارند.

۳,۲. الگوریتم های خوشه بندی برای داده های گراف

در این بخش، انواعی از الگوریتم های خوشه بندی داده های گراف را مورد بحث قرار می دهیم. این شامل الگوریتم های سنتی خوشه بندی گراف و الگوریتم های خوشه بندی داده های XML می شود. الگوریتم های خوشه بندی کاربردهای قابل ملاحظه ای در انواع گوناگونی از نقشه های گراف از جمله ارزیابی تراکم، جهت یابی سهولت، و تلفیق داده های XML دارند [۱۲۶]. در چارچوب الگوریتم های گراف، خوشه بندی می تواند به دو شکل وجود داشته باشد:

- **الگوریتم های خوشه بندی گره:** در این حالت، یک گراف بزرگ داریم و تلاش می کنیم با استفاده از مقادیر فاصله (یا شباهت) نزدیک کران ها، گره های اصلی را خوشه بندی کنیم. در این شرایط، کران های گراف با مقادیر فاصله ای عددی بر چسب زده می شوند. این مقادیر فاصله عددی به منظور ایجاد خوشه هایی از گره ها مورد استفاده قرار می گیرند. یک حالت ویژه این است که در صورت وجود یک کران به آن مقدار تشابه برابر با ۱ نسبت داده می شود، درحالی که در صورت نبود یک کران به آن مقدار تشابه برابر با ۰ داده می شود. لازم به یادآوری است که مسئله به حداقل رساندن تشابه درون ـ خوشه ای برای تعداد ثابتی از خوشه ها لزوماً به مسئله تصمیم بندی گراف یا مسئله برش چند سویه مینیمم تنزل پیدا می کند. و این به مسئله استخراج گراف های متراکم و دسته های ساختگی نیز گفته می شود. اخیراً، مسئله در آثار حوزه پایگاه داده ها به عنوان تعیین دسته های ظاهری (ساختگی) مورد مطالعه قرار گرفته است. در این مسئله، گروه هایی از گره ها را تعیین می کنیم که تقریباً دسته هستند. به عبارت دیگر، یک کران با احتمال بالا بین هر جفت گره مجموعه وجود دارد. گروه های مختلف الگوریتم های خوشه بندی گره را در بخش دیگری مورد مطالعه قرار خواهیم داد.

- **الگوریتم های خوشه بندی گراف:** در این حالت، یک تعداد (احتمالاً برگ) از گراف ها که باید بر اساس رفتار ساختاری خود خوشه بندی شوند، در اختیار داریم. این مسئله از آن جهت چالشی است که تناظر ساختارهای گراف های اصلی و استفاده از این ساختارها برای خوشه بندی ضروری است. هر دو این الگوریتم ها در چارچوب مجموعه داده های گراف سنتی و داده های نیمه ساختاری مورد بررسی قرار می گیرند. بنابراین، هر دوی این شکل ها را بررسی خواهیم کرد.

در زیر بخش های بعدی، هر یک از الگوریتم های خوشه بندی گراف فوق الذکر را مورد بحث قرار خواهیم داد.

الگوریتم های خوشه بندی گره. تعدادی از الگوریتم های خوشه بندی گره گراف در [۷۸] بررسی شده اند. در [۷۸]، مسئله خوشه بندی گره با مسائل برش مینیمم و تقسیم بندی گراف مرتبط شده است. در این حالت، فرض بر این است که گراف های اصلی دارای بارهایی روی کران ها هستند. تقسیم بندی گراف به شیوه ای که بارهای کران ها در عرض تقسیمات به حداقل برسد، مطلوب است. ساده ترین حالت، مسئله برش مینیمم دو ـ راهی است، که در آن می خواهیم گراف را به دو خوشه تقسیم کنیم به گونه ای که بارهای کران ها در عرض تقسیمات به حداقل برسد. این نسخه از مسئله به طور مؤثر قابل حل است و با استفاده از برنامه های کاربردی پی در پی مسئله جریان بیشینه می توان آن را مرتفع کرد [۱۳]؛ زیرا جریان پیشینه بین منبع S و سینک t عامل تعیین برش مینیمم s ـ t خواهد بود. با استفاده از ترکیب متفاوت منبع و سینک، پیدا کردن برش مینیمم کلی امکان پذیر است. روش دوم برای تعیین برش مینیمم با استفاده از رویکرد نمونه برداری کران انجان می شود. این یک تکنیک احتمال گراست که در آن کران ها را به طور توالی نمونه برداری می کنیم به این منظور که گره ها را به مجموعه های بزرگتری از گره ها تجزیه کنیم. با نمونه برداری پی درپی توالی های مختلف از کران ها و انتخاب مقدار مطلوب [۱۷۷]، تعیین یک برش مینیمم کلی امکان پذیر است. هر دو تکنیک فوق کاملاً کارآمد هستند و دشواری زمانی بر حسب تعداد گره ها و کران های چند فرمولی خواهد بود [۷۸].

مسئله تقسیم بندی چند سویه گراف به طور قابل ملاحظه ای دشوار تر است و NP-hard به شمار می آید [۸۰]. در این مورد، می خواهیم یک گراف را به مؤلفه های K>۲ تقسیم کنیم به طوری که بار کلی کران هایی که انتهای آنها در قسمت های مختلفی قرار گرفته به حداقل برسد. الگوریتم Kerninghan-Lin [۱۱۶] رویکرد شناخته شده ای برای تقسیم بندی گراف است. این الگوریتم سنتی بر مبنای تپه نوردی (یا تکنیک جستجوی ـ مجاور) برای تعیین تقسیم بندی مطلوب گراف استوار است. در ابتدا، با برش تصادفی گراف شروع می کنیم. در هر تناوب، یک جفت از رأس ها در دو قسمت را مبادله می کنیم تا ببینیم که آیا مقدار کلی برش کاهش می یابد. در صورتی که مقدار برش کاهش یابد، آنگاه جابه جا کردن در دستور کار قرار می گیرد. در غیر این صورت، جفت رأس دیگری را برای جابه جایی انتخاب می کنیم. لازم است یادآوری کنیم که این مطلوب نمی تواند مطلوب کلی باشد، بلکه ممکن است فقط یک مطلوب محلی از داده های اصلی باشد. شکل اصلی در نسخه های مختلف الگوریتم Kerninghan-Lin سیاستی است که برای اجرای جابه جایی رأس ها اتخاذ می شود. باید خاطر نشان کنیم که استفاده از تعداد بیشتری از استراتژی ها پیشرفته موجب پیشرفت بهتر در عملکرد ملموس

برای هر جابه جایی می شود، اما نیازمند زمان بیشتری برای هر جابه جایی است. این یک تعادل طبیعی است که بسته به نوع برنامه ی کاربردی که در دست داریم ممکن است به گونه ای متفاوت عمل کند. لازم به ذکر است که مسئله تقسیم بندی گراف به طور گسترده در آثار این حوزه مورد مطالعه قرار گرفته است. ارزیابی و بررسی مشروح آن در [۷۷] قابل مشاهده است.

تعیین گراف فرعی متراکم در گراف های بزرگ و گسترده، مسئله ای کاملاً مرتبط است. در مجموعه داده های گراف بزرگ به طور مکرر با این مسئله مواجه می شویم. به عنوان مثال، مسئله تعیین گراف های فرعی بزرگ در گراف های وب در [۸۲] مورد مطالعه قرار گرفت. در این مقاله، از رویکرد عدد ـ مینیمم برای تعیین پلاک هایی که نشان دهنده گراف های فرعی متراکم هستند، استفاده می شود. ایده فراگیر، بیان لینک های خارجی یک گره مشخص به شکل مجموعه هاست. دو گره مشابه هستند که اگر و تنها اگر دارای تعداد زیادی لینک خارجی مشترک داشته باشند. از این رو، گره A با مجموعه لینک خارجی S_A و گره B با مجموعه لینک های خارجی S_B را درنظر بگیرید. آنگاه، شباهت بین دو گره با استفاده از ضریب Gaccard تعریف می شود، که به شکل $\frac{S_A \cap S_B}{S_A \cup S_B}$ تعیین می شود. باید اشاره کنیم که شمارش علنی تمام کران ها به منظور محاسبه این شباهت کاملاً غیر کارآمد خواهد بود. در عوض، به منظور اجرای برآورد شباهت از رویکرد عدد ـ مینیمم استفاده می شود. این رویکرد به صورت زیر اجرا می شود. فضای گره ها را با یک ترتیب تصادفی تفکیک می کنیم. برای هر یک از مجموعه گره هایی که به صورت تصادفی مرتب شده اند گره نخست یعنی First(A) را تعیین می کنیم که برای آن یک لینک خارجی از A به First(A) وجود دارد. همچنین گره نخست First(B) را تعیین می کنیم که برای آن یک لینک خارجی از B به First(B) وجود دارد. می توان نشان داد که ضریب Jaccard یک برآورد غیر جهت دار این احتمال است که First(A) و First(B) گره مشابهی باشند. با تکرار این فرآیند در جایگشت های مختلفی در فضای گره ها، برآورد دقیق ضریب Jaccard عملی خواهد بود. این کار با استفاده از تعداد ثابتی از جایگشت های مختلفی در فضای گره ها، برآورد دقیق ضریب Jaccard عملی خواهد بود. این کار با استفاده از تعداد ثابتی از جایگشت های C در ترتیب گره انجام می شود. بنابراین، برای هر گره یک اثر انگشت از اندازه C قابل تولید است. با مقایسه اثر انگشت های دو گره می توان ضرب Jaccard ار برآورد کرد. این رویکرد با استفاده از هر مجموعه عامل S که فقط دارای S_A و S_B باشد قابل تعمیم است. با استفاده از مقادیر مختلف S و C، طراحی یک الگوریتم که بتواند دو مجموعه بالاتر یا پایین تر از آستانه تعیین شده شباهت را تمیز دهد امکان پذیر خواهد بود.

تکنیک کلی در [۸۲]، ابتدا مجموعه ای از C پلاک با اندازه ی S برای هر گره می سازد. فرآیند ایجاد پلاک C بسیار ساده است. هر گره به طور جداگانه پردازش می شود. از تابع عدد مینیمم برای ایجاد زیر مجموعه هایی با اندازه S از لینک های خارجی در هر گره استفاده می کنیم. این کار موجب ایجاد C زیر مجموعه برای هر گره می شود. بنابر این، برای هر گره مجموعه ای از پلاک های C خوانیم داشت. از این رو، اگر گراف دارای n گره باشد، اندازه ی کل اثر انگشت های این پلاک عبارت است از $n \times C \times SP$، که SP فضای مورد نیاز برای هر پلاک

است. به طور معمول، SP به صورت O (S) است، زیرا هر پلاک شامل S گره است. برای هر پلاک مجزا که به این شکل ایجاد شده، می توانیم فهرستی از گره ها را ایجاد کنیم دارای آن پلاک هستند. به طور کلی ممکن است مایل باشیم گروهی از پلاک ها را تعیین کنیم که حاوی تعداد زیادی از گره های مشترک باشند. به این منظور، روش مطرح شده در [۸۲] یک پلاک سازی درجه دوم را اجرا کنیم که در آن فرا پلاک هایی از پلاک ها ایجاد شوند. از این رو، این کار موجب فشردگی بیش از پیش گراف در یک ساختار داده ها با اندازه ی C×C می شود. این در اصل یک ساختار داده های دارای اندازه ثابت است. لازم به ذکر است که این گروه از فرا پلاک ها دارای این ویژگی هستند که تعداد زیادی از گروه های مشترک را شامل می شوند. آنگاه می توان گراف های فرعی را از این فرا پلاک ها استخراج کرد. جزئیات بیشتر درباره ی این رویکرد در [۸۲] قابل مشاهده است.

تعیین به ظاهر ـ دسته ها (دسته های ساختگی) در داده های اصلی می تواند مسئله ای مرتبط باشد. به ظاهر ـ دسته ها در اصل تخفیف هایی (کاهش هایی) بر مفهوم دسته ها هستند. در مورد یک دسته، گراف های فرعی که بر مجموعه ای از گره ها القاء شده اند، کامل است. از سوی دیگر، در مورد یک به ظاهر ـ دسته ی ـ γ، هر رأس در آن زیر مجموعه از گره ها دارای یک رتبه ی K ـ γ است، که γ یک قطعه است، و K معادل تعداد گره ها موجود در آن مجموعه است. اولین قدم در تعیین به ظاهر ـ دسته های ـ γ در [۵] مورد بررسی قرار گرفته، که در آن یک الگوریتم تصادفی به منظور تعیین به ظاهر ـ دسته دارای بزرگترین اندازه مورد استفاده قرار می گیرد. مسئله کاملاً مرتبط با آن، یافتن دسته های پرتکرار در مجموعه داده های چندگانه است. به عبارت دیگر، وقتی گراف های چندگانه از مجموعه داده های مختلفی به دست می آید بعضی از گراف های فرعی متراکم مکرراً در مجموعه داده های مختلف ظاهر می شوند. این گراف ها به تعیین الگوهای متراکم مهم رفتاری در منابع داده های مختلف کمک می کنند. اینگونه تکنیک ها در استخراج الگوهای مهم در تصاویر گرافیکی مشتریان کارایی خود را نشان می دهند. تشریح کاربرد این تکنیک در مسئله ی داده های بیان ـ ژن در [۱۵۳] قابل مشاهده است. الگوریتم کار آمدی برای تعیین به ظاهر ـ دسته های گراف فرعی در [۱۴۸] معرفی شده است.

الگوریتم های سنتی برای خوشه بندی داده های گراف و XML. در این بخش، انواعی از الگوریتم های خوشه بندی داده های گراف و XML را مورد بحث قرار خواهیم داد. باید یادآوری کنیم که داده های XML از نظر چگونگی سازماندهی ساختاری کاملاً مشابه داده های گراف هستند. در [۱۳۳،۱۲۶،۶۳،۸] نشان داده شده است که استفاده از این رفتار ساختاری اهمیت زیادی در پردازش مؤثر داده ها دارد. دو تکنیک عمده برای خوشه بندی اسناد XML وجود دارد. این تکنیک ها به قرار زیر هستند:

- **رویکرد ساختاری فاصله ـ محور:** این رویکرد به محاسبه فاصله ساختاری بین اسناد مبادرت می کند و برای محاسبه خوشه های اسناد از آنها استفاده می کند. این گونه رویکردهای فاصله ـ محور، تکنیک هایی کاملاً عمومی و کارآمد هستند که برای انواع گسترده ای از حوزه های غیر عددی مانند داده های زنجیره ای و صریح مورد استفاده قرار می گیرند. به همین خاطر، شناسایی این تکنیک در چارچوب

داده های گراف طبیعی به نظر می رسد. یکی از کارهای اولیه در خوشه بندی داده های دارای ساختار نمودار درختی متعلق به الگوریتم XClust [۱۲۶] است، که برای خوشه بندی چارچوب کلی XML برای تلفیق مؤثر تعداد زیادی از تعاریف اسناد گونه (DTDs) مجزا آغاز می کند و به تدریج دو خوشه ای را که دارای بیشترین تشابه هستند را در یک خوشه بزرگتر ادغام می کند. شباهت بین دو DTDs بر اساس شباهت میانی آنها تعیین می شود، که می توان آن را بر طبق اطلاعات معنایی، ساختاری و مفهومی عامل های موجود در DTDs متناظر محاسبه کرد. یکی از کمبودهای الگوریتم XClust این است که از اطلاعات ساختاری DTDs به طور کامل استفاده نمی کند، که در خوشه بندی ساختارهای درخت، مانند اهمیت فوق العاده ای دارد. روش ارائه شده در [۴۵] بر اساس فاصله ی ـ ویرایش ساختاری بین اسناد به ارزیابی مقیاس های شباهت می پردازد. فاصله ی ـ ویرایش برای محاسبه ی فاصله ی بین خوشه های اسناد مورد استفاده قرار می گیرد.

تکنیک خوشه بندی دیگری که در این گروه عمومی از روش ها قرار می گیرد، الگوریتم S-GRACE است. ایده اصلی آن استفاده از رابطه عامل ـ زیر عامل در تابع فاصله به جای استفاده آسان از فاصله ویرایش ـ نمودار درختی است که در [۴۵] ارائه شده است. S-GRACe یک الگوریتم خوشه بندی طبقاتی است [۱۳۳]. در [۱۳۳]، یک سند XML به ساختار گراف (یا گراف فرعی) تبدیل شده است، و فاصله بین دو سند XML بر حسب تعداد روابط عامل ـ زیر عامل مشترک تعیین شده است که می تواند در بعضی از موارد روابط شباهت ساختاری بهتری نسبت به فاصله ویرایش نمودار درختی به دست آورد [۱۳۳].

- **رویکرد ساختاری خلاصه ـ محور:** در اکثر موارد، ایجاد خلاصه هایی از اسناد اصلی امکان پذیر است. این خلاصه ها برای ایجاد گروه هایی از اسناد که مشابه این خلاصه ها هستند مورد استفاده قرار می گیرند. نخستین رویکرد خلاصه ـ محور برای خوشه بندی اسناد XML در[۶۳] ارائه شده است. در [۶۳]، اسناد XML به عنوان نمودارهای درختی برچسب دار منظم ریشه دار طراحی می شوند. چارچوبی برای خوشه بندی اسناد XML با استفاده از خلاصه های ساختاری ارائه شده است. هدف این چارچوب افزایش کارایی الگوریتم بدون به خطر انداختن کیفیت خوشه بندی است.

رویکرد دوم برای خوشه بندی اسناد XML در [۱۸] ارائه شده است، و تحت عنوان xproj شناخته می شود این تکنیک یک الگوریتم تقسیم بندی است. ایده اولیه در این رویکرد، استفاده از الگوریتم های استخراج الگوی پرتکرار به منظور تعیین خلاصه هایی از ساختارهای پرتکرار در داده هاست. این تکنیک از رویکرد میانگین های ـK استفاده می کند که در آن هر یک از مرکز های خوشه متشکل از مجموعه ای از الگوهای پرتکرار است که به همان قسمت مشخص شده برای خوشه تعلق دارند. الگوهای پرتکرار با استفاده از اسنادی که در آخرین تناوب به مرکز خوشه اختصاص داده شده اند، استخراج می شوند. آنگاه، براساس شباهت میانگین بین سند و مراکز خوشه ی جدیداً ایجاد شده از الگوهای پرتکرار محلی، مجدداً اسناد به یک مرکز خوشه تخصیص داده می شوند، تا زمانی که مراکز خوشه و تقصیمات سند

در یک وضعیت نهایی تلاقی پیدا کنند. در [۸] نشان داده شده است که یک رویکرد ساختاری خلاصه ـ محور برتری چشمگیری بر رویکرد شباهت ـ محور ارائه شده در [۴۵] دارد. این روش بر رویکرد ساختاری معرفی شده در [۶۳] نیز برتری دارد زیرا از بازنمایی های خلاصه های ساختاری اصلی بیشتر استفاده می کند.

۳,۳. الگوریتم های طبقه بندی برای داده های گراف

طبقه بندی، کاری محوری در استخراج داده ها و یادگیری دستگاه است. از آنجایی که گراف ها برای بیان واحدها و ارتباط بین آنها در انواع فزاینده ای از برنامه های کاربردی مورد استفاده قرار می گیرند، موضوع طبقه بندی گراف توجه زیادی را در دانشگاه و صنعت به خود معطوف کرده است. به عنوان مثال، در داروسازی و طراحی داروها، مایلیم به رابطه بین فعالیت یک ترکیب شیمیایی و ساختار ان که توسط یک گراف بیان می شود، پی ببریم. در تحلیل شبکه اجتماعی، ارتباط بین سلامت جامعه (مثلاً در حال گسترش یا کاهش است) و ساختار آن که به صورت گراف ارائه می شود، مورد مطالعه قرار می گیرد.

طبقه بندی گراف مستلزم دو فعالیت یادگیری متفاوت اما مرتبط است.

- **انتقال برچسب.** زیر مجموعه ای از گره ها در گراف بر چسب زده می شوند. فعالیت مورد نظر، یادگیری یک نمونه گره های برچسب دار و استفاده از آن برای طبقه بندی گره های بدون برچسب است.
- **طبقه بندی گراف.** زیر مجموعه ای از گراف ها در یک مجموعه داده های گرافی برچسب زده می شوند. این فعالیت به صورت یادگیری یک نمونه از گراف های برچسب دار و استفاده از آن برای طبقه بندی گراف های بدون برچسب انجام می شود.

انتقال برچسب. مفهوم انتقال برچسب یا عقیده [۲۱۰،۲۰۹،۱۷۴] یک تکنیک بنیادی است که به منظور به کار بردن ساختار گراف در طبقه بندی داده ها در بعضی از داده های ارتباطی مورد استفاده قرار می گیرد. طرح انتقال برچسب [۴۴] در بسیاری از برنامه های کاربردی یافت می شود. به عنوان مثال، تحلیل شبکه اجتماعی به عنوان ابزاری برای بازاریابی هدفمند مورد استفاده قرار می گیرد. فروشنده های خرده پا به دنبال مشتری هایی هستند که تبلیغات آنها را دریافت کرده اند. این مشتری ها که (با خرید کردن) به تبلیغات پاسخ می دهند به عنوان گره های مثبت در گراف شبکه اجتماعی برچسب زده می شوند و مشتری هایی است که بیشترین احتمال پاسخ گفتن به تبلیغات از جانب آنهاست. فرآیند مذکور را این طور خلاصه می کنیم: یادگیری یک نمونه از مشتری هایی که تبلیغات را دریافت کرده اند و پیش بینی واکنش های سایر مشتری های بالقوه ای که در شبکه اجتماعی حضور دارند. از راه شهودی، می خواهیم به چگونگی انتقال برچسب های مثبت و منفی موجود در گراف به گره های فاقد برچسب پی ببریم.

بر مبنای این فرض که گره های "مشابه" باید برچسب های مشابهی داشته باشند، چالش کلیدی برای انتقال برچسب در طراحی تابع فاصله ای که شباهت بین دو گره در گراف را اندازه گیری کند، نهفته است.

روش های طبقه بندی گراف هسته – محور. این روش بر مبنای عبور تصادفی طراحی شده است. برای هر گراف، مسیرهای آن را شمارش می کنیم و احتمال های این مسیرها را استخراج می کنیم. هسته ی اصلی گراف به مقایسه مجموعه مسیرها و احتمال های آنها بین دو گراف می پردازد. مسیر تصادفی (که به صورت توالی بر چسب های گره و کران بیان می شود) از طریق عبور تصادفی ایجاد می شود. نخست، به طور تصادفی یک گره از گراف انتخاب می کنیم. در طی و بعد از هر یک از مراحل بعدی، یا توقف می کنیم و یا یک گره مجانب را برای ادامه عبور تصادفی انتخاب می کنیم، انتخاب های ما در معرض احتمال توقف و احتمال انتقال گره هستند. با تکرار عبورهای تصادفی، جدولی از مسیرها به دست می آوریم که هر یک از آن یک احتمال به همراه دارد.

به منظور اندازه گیری شباهت بین دو گراف، باید شباهت بین گره ها، کران ها و مسیرها را اندازه بگیریم.

- **هسته اصلی گره / کران.** هسته مرکزی هویت، یکی از هسته های گره / کران است. اگر دو گره / کران دارای برچسب مشابهی باشند، آنگاه هسته ی اصلی عدد ۱ را ارسال می کند و در غیر اینصورت عدد ۰ را میفرستد. اگر برچسب های گره / کران دارای مقادیر حقیقی باشند، آنگاه یک هسته ی Gaussian قابل استفاده خواهد بود.
- **هسته مرکزی مسیر.** هر مسیر یک توالی از برچسب های گره و کران است. اگر دو مسیر دارای طول یکسان باشند، هسته مرکزی مسیر را می توان به صورت فرآورده هسته مرکزی گره و کران طراحی کرد. اگر دو مسیر دارای طول های متفاوتی باشند، هسته مرکزی مسیر به سادگی عدد ۰ را ارسال می کند.
- **هسته مرکزی گراف.** از آنجایی که هر مسیر با یک احتمال همراه است، می توانیم هسته مرکزی گراف را به صورت احتمال وقوع هسته مرکزی در تمام مسیرهای ممکن در دو گراف تعریف کنیم.

تعریف بالا از هسته مرکزی گراف کاملاً روشن است. با این وجود، شمارش تمام مسیر ها به لحاظ محاسباتی عملی نیست. به ویژه، درگراف های حلقه ای طول یک مسیر نامحدود است که شمارش را غیر ممکن می کند. از این رو، به رویکرد های کارآمد تری برای محاسبه هسته مرکزی نیاز داریم. معلوم می شود که تعریف هسته مرکزی را می توان برای نشان دادن یک ساختار ذخیره ای مجدداً تدوین کرد. در مورد گراف های مستقیم فاقد حلقه، می توان گره ها را ازنظر مکانی مرتب کرد به طوری که هیچ مسیری از گره j به i وجود نداشته باشد اگر i>j، و هسته مرکزی را می توان به صورت یک تابع تناوبی تعریف کرد، و برنامه ریزی (دینامیک می تواند این مسئله را در O(/X/0/X'/) کنترل کند، که X و X' بیانگر مجموعه گره ها در دو گراف هستند. در مورد گراف های حلقه ای، فضای خاصیت هسته مرکزی (توالی های برچسب) به خاطر لوپ ها (حلقه ها) احتمالاً نامحدود

است. محاسبه هسته مرکزی گراف حلقه ای با نظریه ی سیستم خطی و ویژگی های همگرایی هسته ی مرکزی قابل اجرا خواهد بود.

روش های طبقه بندی گراف تشدید ـ محور. هر چند روش مبتنی بر هسته مرکزی یک راه حل دقیق برای طبقه بندی گراف ارائه می دهد، صراحتاً مشخص نمی کند که کدام خاصیت های (زیر ساختارهای) گراف متناسب و مرتبط با طبقه بندی هستند. برای پرداختن به این موضوع، رویکرد جدیدی از طبقه بندی گراف بر مبنای استخراج الگو معرفی می شود. اجرای طبقه بندی گراف بر زیر ساختارهای مهم گراف مد نظر است. می توانیم یک بردار خاصیت دوتایی بر اساس حضور یا عدم حضور یک زیر ساختار مشخص ایجاد کنیم و یک طبقه بندی کننده غیر قفسه ای را به کار بگیریم.

از آنجایی که، کل مجموعه گراف های فرعی اغلب بسیار بزرگ است، باید بر زیر مجموعه کوچکی از خاصیت های مرتبط و متناسب تمرکز کنیم. ساده ترین رویکرد برای پیدا کردن خاصیت های جالب از طریق استخراج الگوی پرتکرار انجام می گیرد. با این وجود، الگوهای پرتکرار لزوماً الگوهای مرتبط و متناسب نیستند. به عنوان مثال، در داده های شیمیایی، الگوهای فراگیر مانند C-C یا C-C-C پر تکرارند، اما هیچ اهمیتی در پیش بینی ویژگی های مهم ترکیب های شیمیایی از جمله فعالیت، سم زدایی و غیره ندارند.

از تشدید برای انتخاب خودکار مجموعه ای از گراف های فرعی مانند خاصیت هایی برای طبقه بندی استفاده می شود. LPBoost (تشدید برنامه خطی) یک تابع تشخیص خطی برای انتخاب خاصیت بیان می کند. برای به دست آوردن یک قانون قابل تفسیر، لازم است یک بردار ارزش نا متراکم به دست آوریم که در آن فقط تعدادی از ارزش ها مقادیری غیر از صفر دارند. در [۱۶۲] نشان داده شده است که تشدید گراف می تواند دقت بهتری از هسته های مرکزی گراف به دست آورد، و دارای مزیت شناسایی زیر ساختارهای کلیدی به طور همزمان است.

مسئله طبقه بندی گراف کاملاً مرتبط با مسئله طبقه بندی XML است؛ به این خاطر که داده های XML را می توان به عنوان نمونه ای از گراف های ارزشمند در نظر گرفت، که در آن گره ها و کران ها دارایی خاصیت هایی همراه با خود هستند. در نتیجه، اکثر روش های طبقه بندی XML برای طبقه بندی گراف های ساختاری نیز قابل اجرا خواهند بود. در [۲۰۵]، یک طبقه بندی کننده قانون ـ محور (به نام XRules) معرفی شد که در آن خاصیت های ساختاری در ضلع سمت چپ را با برچسب های طبقه روی ضلع سمت راست همراه می کنیم. خاصیت های ساختاری روی ضلع دست چپ با محاسبه خاصیت های ساختاری در گراف که هم پرتکرار و هم تشخیص دهنده برای اهداف طبقه بندی هستند، تعیین می شوند. این خاصیت های ساختاری به منظور ایجاد یک لیست اولویت بندی شده از قوانین مورد استفاده در طبقه بندی به کار گرفته می شوند. قوانین و اصول دارای K بیشینه بر مبنای رفتار تشخیصی تعیین می شوند و اکثریت برچسب های طبقه روی ضلع دست راستی این قوانین K به عنوان نتیجه ی نهایی گزارش می شوند.

سایر کارها مرتبط. مسئله طبقه بندی گره در تعدادی از چارچوب های کاربردی مختلف از جمله طبقه بندی داده های ارتباطی، طبقه بندی شبکه اجتماعی و طبقه بندی بلاک دیده می شود. تکنیکی در [۱۳۸] پیشنهاد شده است که از شباهت اتصال ـ محور برای طبقه بندی گره در چارچوب داده های ارتباطی استفاده می کند. این رویکرد خاصیت های اتصال را از ساختار اصلی ایجاد می کند و به منظور ایجاد یک مدل کار آمد طبقه بندی از این خاصیت ها استفاده می کند. اخیراً، از این تکنیک در طبقه بندی اتصال ـ محور بلاگ ها نیز استفاده شده است [۲۳]، با این وجود، تمام این تکنیک ها فقط از روش های اتصال ـ محور بهره می گیرند. از آنجایی که اکثر این تکنیک ها در زمینه داده های متنی مطرح می شوند، طبیعی است بررسی کنیم که آیا می توان از این محتوا برای بهبود دقت طبقه بندی استفاده کرد. روشی برای اجرای طبقه بندی گروهی قوانین گفتار ایمیل در [۳۹] معرفی شده است. نشان داده شده است که تحلیل جنبه های ارتباطی ایمیل (مانند ایمیل ها در یک زنجیره مشخص) به طور چشمگیری باعث افزایش دقت طبقه بندی می شود. همچنین، در [۲۰۶] نشان داده شده است که استفاده از ساختار های گراف در طی گروه بندی می تواند دقت طبقه بندی صفحات وب را افزایش دهد. فعالیتی دیگر [۲۵] به بررسی مسئله فراگیری برچسب در چارچوب طبقه بندی گروهی می پردازد.

۳,۴. فعالیت های گراف های تکامل ــ زمان

بسیاری از شبکه ها در برنامه های کاربردی واقعی در چارچوب واحد های شبکه ای مانند وب، شبکه های موبایل، شبکه های نظامی، شبکه های اجتماعی پدیدار می شوند. در این گونه موارد، بررسی جنبه های مختلف فعالیت های تکاملی شبکه های واقعی مانند وب یا شبکه های اجتماعی می تواند مفید واقع شود. از این رو، این گستره تحقیقی بر طراحی ویژگی های تکاملی عمومی گراف های بزرگ که به طور واقعی با آنها روبه رو می شویم، تمرکز می کند. تحقیق های فراوانی به بررسی ویژگی های تکامل کلی پرداخته اند، ویزگی هایی که در شبکه های فراگیری مانند شبکه های وب، شبکه های فراخوانی و شبکه های اجتماعی معتبر هستند. برخی نمونه های این ویژگی ها عبارتند از:

تراکم. اکثر شبکه های واقعی مثل وب و شبکه های اجتماعی با گذشت زمان متراکم تر می شوند [۱۲۹]. این در اصل به این معناست که این شبکه ها به اضافه کردن لینک ها (بیش از حذف لینک ها) در طول زمان ادامه می دهند. این پیامد طبیعی این حقیقت است که بیشتر شبکه های وب و رسانه های اجتماعی پدیده ای نسبتاً جدیدی هستند که با گذشت زمان برنامه های کاربردی جدیدی برای آنها پیدا می شود. در حقیقت، بسیاری از گراف های واقعی یک قانون توان تراکم به نمایش می گذارند که تغییر در رفتار تراکم در طول زمان را ترسیم می کند، این قانون اعلام می کند که تعداد گره های موجود در شبکه به طور خطی با تعداد گره ها در طول زمان افزایش می یابد. به عبارت دیگر، اگر $n(t)$ و $e(t)$ بیانگر تعداد کران ها و گره های شبکه در زمان ۴ باشند، آنگاه داریم:

$$e(t) \propto n(t)^{\propto} \qquad (۲۰۱)$$

مقدار توان $\propto$ بین ۱ و ۲ قابل قبول است.

قطرهای انقباضی. پدیده دنیای کوچک گراف ها به خوبی شناخته شده است. به عنوان مثال، در [۱۳۰] نشان داده شده است که طول میانگین مسیر بین دو کاربر مسنجر MSN حدوداً ۶۰۶ است. این را می توان تأیید قانون شناخته شده "درجه جداسازی" در شبکه های اجتماعی در نظر گرفت. در [۱۲۹] نشان داده شده است قطرهای شبکه های فراگیر و گسترده ای مانند وب در طول زمان کاهش می یابند. ممکن است تعجب کنید، زیرا انتظار می رود که قطرهای شبکه با افزودن گره ها مرتباً رشد کنند. با این وجود، باید به خاطر داشته باشید که کران ها با سرعت بیشتری از گره ها به شبکه افزوده می شوند (که معادله ۲۰۱ آنرا بیان می کند). هر چه کران های بیشتری به گراف اضافه می شود عبور از یک گره به گره دیگر با استفاده از تعداد کمتری از کران ها امکان پذیر خواهد بود.

در حالی که تفکر فوق الذکر درک بهتری از بعضی جنبه های کلیدی تکامل دراز مدت گراف های گسترده و فراگیر ارائه می دهد، درباره اینکه چگونه تکامل در شبکه های اجتماعی را می توان به شیوه ای همه جانبه و فراگیر طراحی کرد هیچ ایده ای مطرح نمی کند. روشی که در [۱۳۱] پیشنهاد شده است از اصل احتمال بیشینه برای ترسیم رفتار تکاملی شبکه های اجتماعی فراگیر استفاده می کند. این روش از استراتژی های داده های ـ استخراج شده برای طراحی رفتار آنلاین شبکه ها استفاده می کند. روش مذکور به مطالعه ی رفتار ۴ مدل مختلف شبکه می پردازد، و از اطلاعات این شبکه ها استفاده می کند تا یک مدل تکامل اصلی به وجود آورد. همچنین نشان می دهد که موقعیت کران نقش مهمی در تکامل شبکه های اجتماعی بازی می کند. یک نمونه کامل از رفتار گره در طول دوره زندگی اش در شبکه در این اثر مورد بررسی قرار گرفته است.

یک گستره تحقیق دیگر در این حوزه به مطالعه روش هایی برای توصیف تکامل نمودارهای خاص باز می گردد. به عنوان مثال، در یک شبکه اجتماعی، ممکن است تعین جوامع تازه تشکیل شده یا تازه منقرض شده در شبکه های اصلی سودمند باشد [۱۷۳،۱۷۱،۱۳۵،۱۳۱،۱۱۷،۷۴،۶۹،۵۰،۱۶،۹]. در [۹] نشان داده شده است که چگونه جوامع در حال گسترش یا در حال انقباض در شبکه اجتماعی ممکن است از طریق بررسی رفتار نسبی کران ها به گونه ای که در یک زنجیره گراف دینامیک (پویا) دریافت شده اند، ترسیم شوند. تکنیک مطرح شده در این گزارش، رفتار ساختاری گراف رشد یابنده درون یک بازه زمانی معین را ترسیم می کند و از آن برای تعیین تولد و مرگ جوامع در زنجیره گراف بهره می گیرد. این نخستین بخش کار است که مسئله تکامل در زنجیره های سریع گراف ها را مورد مطالعه و تحقیق قرار می دهد. به خاطر پیچیدگی و دشواری ترکیبی ذاتی تحلیل ساختاری گراف، که طرح زنجیره را در خود جای نمی دهد، در مطالعه زنجیره با چالش روبه رو خواهیم شد.

فعالیت مطرح شده در [۶۹]، از تحلیل و تجسم آماری برای ارائه یک ایده بهتر برای ساختار جامعه در حال تغییر در شبکه اجتماعی تکاملی استفاده می کند. روش موجود در [۱۷۱]، استخراج بدون ـ پارامتر گراف های

بزرگ تکاملی در طول زمان را اجرا می کند. این تکنیک می تواند جوامع در حال تکامل در شبکه ها و تغییرات جدی در طول زمان را تعیین کند. یکی از ویژگی های کلیدی این روش این است که فاقد پارامتر است و این قابلیت استفاده از آن در بسیاری از طرح ها را امکان پذیر می کند. با استفاده از اصل MDL در فرآیند استخراج می توان به این هدف دست یافت. یک تکنیک مرتبط نیز قادر است تحلیل بدون پرامتر تکامل در شبکه های گسترده و فراگیر [۷۴] را با استفاده از اصل MDL اجرا کند. این تکنیک می تواند تعیین کند که کدام جوامع در طول زمان منقبض شده اند، منشعب شده اند یا پدید آمده اند.

مسئله تکامل در گراف ها معمولاً در چارچوب خوشه بندی مورد مطالعه قرار می گیرد، زیرا خوشه ها یک خلاصه طبیعی برای درک گراف اصلی و تغییرات در طی فرآیند تکامل در اختیار ما می گذارند. نیاز به این گونه توصیف ها در شرایط شبکه های فراگیر از جمله گراف های برهم کنش [۱۶]، ارزیابی جامعه در شبکه های اجتماعی [۱۷۳، ۱۳۵، ۵۰، ۹]، و تغییرات کلی خوشه بندی در شبکه های اطلاعاتی اتصال یافته [۱۱۷] دیده می شود. تحقیق انجام شده توسط [۱۶]، یک چارچوب رویداد ـ محور ارائه می دهد که درک بهتری از رویدادهای معمولی که در شبکه های واقعی در هنگام تشکیل، تکامل یا فروپاشی جوامع رخ می دهد، در اختیار ما می گذارد. از این رو، این روش می تواند روش ساده ای در اختیار ما می گذارد تا به سرعت تعیین کنیم که آیا تغییرات خاصی در یک شبکه خاص روی داده است یا نه. تکنیک مهمی که توسط بسیاری از روش ها مورد استفاده قرار می گیرد تحلیل جوامع در داده ها در برش های زمانی خاص و سپس تعیین تغییر بین برش های زمانی مختلف به منظور تشخیص ماهیت تکامل زیر بنایی است. روش معرفی شده در [۱۳۵] از این رویکرد دو مرحله ای فاصله می گیرد و یک چارچوب یکپارچه برای تعیین جوامع با استفاده از بهترین قالب مدل یکپارچگی ـ گذرا ایجاد می کند. تحقق ارائه شده در [۵۰] یک روش طیفی برای خوشه بندی تکاملی معرفی می کند. که بر اساس مفهوم یکپارچگی گذرا طراحی شده است. روش موجود در [۱۷۳] به بررسی تکنیک هایی برای توصیف تکاملی شبکه ها در گراف های چند وجهی می پردازد. سرانجام، روش نوظهور که در [۱۱۷] پیشنهاد شده، مسئله خوشه بندی و تحلیل تکاملی را در یک چارچوب ادغام می کند، و نشان می دهد چگونه خوشه های در حال تکامل در یک محیط دینامیک (پویا) را تعیین کنیم. روش موجود در [۱۱۷] از یک توصیف تراکم ـ محور به منظور ایجاد خوشه های نانو که برای تحلیل تکاملی به کار می روند؛ استفاه می کند.

استفاده از تکنیک های استخراج قانون ـ محور [۲۲]، رویکردی متفاوت محسوب می شود. الگوریتم مورد نظر یک سلسله عکس از یک گراف در حال تکامل می گیرد، و پس از آن تلاش می کند قوانین و اصولی مشخص کند که تغییرات در گراف اصلی را تعیین کنند. توالی های پرتکرار تغییرات در گراف اصلی به عنوان راهنما های مهم برای تعیین قانون در نظر گرفته می شوند. به علاوه، الگوهای پرتکرار تجزیه می شوند به منظور بررسی این اطمینان که یک توالی خاص از گام ها در گذشته منجر به یک جابه جایی ویژه می شود. احتمال چنین جابجایی، "اطمینان" نامیده می شود. سپس قوانین در گراف اصلی به منظور توصیف تکامل کلی شبکه مورد استفاده قرار می گیرند.

شکل دیگری از تکامل در شبکه ها در قالب گردش پیام ها (اطلاعات) اصلی ظاهر می شود. از آنجایی که گردش پیام ها و اطلاعات به طور ضمنی یک گراف (زنجیره) را تعیین می کند، فعالیت های این رفتار ممکن است برای بسیاری از برنامه های کاربردی مختلف جالب توجه باشد. اینگونه رفتارها اغلب در انواعی از شبکه های اطلاعات از جمله شبکه های اجتماعی، بلاگ ها، یا گراف های نقل قول نویسنده پدید می آیند. در بسیاری از موارد، تکامل ممکن است شکل اطلاعات سرریز شده از گراف های اصلی به خود بگیرد. هدف این است که اطلاعات به وسیله شبکه ای اجتماعی از طریق ارتباط بین واحدهای مختلف شبکه انتقال پیدا کند. تکامل این گردش اطلاعات همزمان با انتشار عیب ها در شبکه، تعدادی شباهت را به اشتراک می گذارد. در بخش بعدی بیشتر درباره این موضوع سخن خواهیم گفت. این تکامل در [۱۲۸] مورد مطالعه قرار گرفته است، که بررسی می کند چگونه باید رفتار تکامل در گراف های بلاگ را توصیف کرد.

۴. کاربردهای گراف

در این بخش، کاربرد بسیاری از الگوریتم های استخراج پیش گفته در انواع مختلفی از برنامه های کاربردی حوزه گراف را بررسی خواهیم کرد. بسیاری از حوزه های داده ها از قبیل داده های شیمیایی، داده های زیستی، و وب معمولاً به شکل گراف سازماندهی می شوند. بنابراین، طبیعی است که بسیاری از برنامه های کاربردی که پیش تر به بررسی آنها پرداختیم قابل استفاده در این حوزه ها هستند. در این بخش، برنامه های کاربردی گوناگونی که از کمک تکنیک های استخراج گراف بهره مند می شوند را بررسی خواهیم کرد. همچنین خواهیم دید به رغم اینکه این برنامه های کاربردی از حوزه های مختلفی به دست آمده اند، رگه های مشترکی وجود دارند که می توان از آنها برای ارتقاء کیفیت نتایج زیر ساختی بهره گرفت.

۴,۱. برنامه های کاربردی شیمیایی و زیستی

کشف دارو فعالیتی زمان بر و به شدت پرهزینه است. گراف ها، بازنمایی های طبیعی برای ترکیب های شیمیایی هستند. در گراف های شیمیایی، گره ها نشان دهنده اتم ها و کران ها بیانگر پیوند بین اتم ها هستند. گراف های زیستی معمولاً در سطح بالاتری قرار دارند که گره ها بیانگر آمینو اسید ها و کران ها معرف پیوستگی یا ارتباط بین آمینو اسیدها هستند. یک فرایندی مهم، که به عنوان اصل رابطه فعالیت ساختار [SAR] شناخته می شود، این است که ویژگی ها و فعالیت های زیستی ترکیبات شیمیایی به ساختار آنها بستگی دارد. از این رو، استخراج گراف ممکن است به شناسایی ویزگی های شیمیایی و زیستی مانند فعالیت، سم زایی، جذب، سوخت و ساز، و غیره کمک کند [۳۰]، و فرآیند ساخت دارو را تسهیل کند. به همین دلیل، تحقیقات دانشگاهی و صنعت داروسازی تلاش هایی در زمینه ی استخراج گراف های انجام داده اند به این امید که زمان و هزینه کشف دارو را به طور چشمگیری کاهش دهند.

اگر چه گراف ها بازنمایی طبیعی ساختارهای شیمیایی و زیستی هستند، هنوز هم به بازنماهای کارآمد تری تحت عنوان "معرف" داریم که منشأ فعالیت هایی است که از جستجوی شباهت تا پیش بینی های مختلف ساختارهای استخراج شده را در بر می گیرد. تنها تعداد اندکی معرف تاکنون معرفی شده اند. به عنوان مثال، اثر انگشت های نشانه [۲۰۱] نوعی بازنمای برداری هستند. با در نظر گرفتن یک گراف شیمیایی، یک اثر انگشت نشانه با شمارش انواع معین ساختارهای پایه ای (مانند حلقه ها و مسیرها) در گراف ها و تجزیه ی آنها به یک زنجیره کوچک ایجاد می کنیم. در یک تحقیق دیگر، محققان از روش های استخراج داده ها برای پیدا کردن گراف های فرعی پرتکرار [۱۵۰] در یک پایگاه های داده های گراف شیمیایی استفاده می کند و هر گراف شیمیایی را به شکل یک بردار در فضای خاصیت با مجموعه ای از گراف های فرعی پرتکرار بیان می کند. توضیح و مقایسه ی مشروح معرف های مختلف را می توانید در [۱۹۰] مشاهده کنید.

جستجوی شباهت یکی از فعالیت های زیر بنایی در ترکیب های شیمیایی است. الگوریتم های تناظر گراف مختلف برای موارد زیر به خدمت گرفته شده اند: i) بازیابی ـ رتبه، یعنی جستجوی یک پایگاه داده های بزرگ برای پیدا کردن ترکیب های شیمیایی که دارای فعالیت زیستی مشابهی مثل یک ترکیب مورد سوال هستند؛ و ii) جهش ـ سکو، یعنی پیدا کردن ترکیب های دارای فعالیت زیستی مشابه ولی ساختار متفاوت از ترکیب مورد سوال هستند. جهش ـ سکو برای شناسایی ترکیب هایی به کار می رود که جایگزین خوبی برای ترکیب مورد سوال (مورد مطالعه) هستند، و یا دارای چند ویژگی نامطلوب هستند (مثلاً سم زایی)، یا از فضای شیمیایی انحصاری موجود آمده اند. از آنجایی که ساختار شیمیایی فعالیت زیستی را تعیین می کند (اصل SAR). "جهش ـ سکو" چالشی پیش روی کاربر قرار می دهد زیرا ترکیب های شیمیایی باید به لحاظ ساختاری آنقدر شبیه باشند که فعالیت زیستی مشابهی به نمایش بگذارند اما در عین حال باید تفاوت آنها به قدری باشد که یک گونه شیمیایی جدید به شمار آیند. رویکردهای کنونی برای تناظر شباهت را می توان در گروه طبقه بندی کرد. یکی از این گروه ها، تناظر شباهت را مستقیماً در فضای معرف انجام می دهد [۲۰۷، ۱۷۰، ۱۹۲]. گروه دیگر، تناظر غیر مستقیم را اجرا می کند: اگر یک ترکیب شیمیایی C از نظر ساختاری مشابه ترکیب مورد سوال q باشد، و ترکیب شیمیایی 'C از نظر ساختاری مشابه ترکیب C باشد آنگاه 'C و q متناظرهای غیر مستقیم هستند یعنی به طور غیر مستقیم با یکدیگر تناظر دارند، بدیهی است که تناظر غیر مستقیم مستعد شناسایی ترکیب هایی است که به لحاظ عملکرد مشابه هستند ولی از نظر ساختاری متفاوتند، که این موضوع در "جهش ـ سکو" اهمیت دارد [۱۹۱، ۱۸۹].

پیش بینی ساختار استخراج شده یکی دیگر از حوزه های کاربردی مهم برای استخراج گراف های شیمیایی و زیستی است. هدف این فرآیند، پیش بینی فعالیت یا عدم فعالیت یک ساختار شیمیایی است، یا پیش بینی اینکه ساختار مورد نظر دارای ویژگی های معین مثل سمی بودن یا غیر سمی بودن، غیره است، ثابت شده که روش های مبتنی بر SVM (سیستم های برداری پشتیبان) در این حوزه کارایی زیادی دارند. توابع هسته ای بر مبنای فضاهای برداری گوناگون از قبیل تابع رایج شعاعی و تابع هسته ای Min-Max برای اندازه گیری شباهت

بین ترکیب های شیمیایی که توسط بردارها بیان می شوند، مورد استفاده قرار می گیرند. به جای فعالیت در فضای بردار، گروه دیگری از روش های SVM از هسته های مرکزی گراف برای مقایسه ی دو ساختار شیمیایی بهره می گیرند. به عنوان مثال، در [۱۶۰]، اندازه گراف فرعی مشترک بیشینه دو گراف به عنوان مقیاس شباهت مورد استفاده قرار می گیرد.

در اواخل ۱۹۸۰، صنعت داروسازی، یک الگوی جدید کشف دارو به نام کشف داروی هدف ـ محور با پذیرفت. هدف آن ساختن دارویی است که اثرات ژن بیماری زا یا فرآورده ژن را بدون تأثیر بر سایر ژن ها یا مکانیسم های مولکولی در موجود زنده کم کند. این هدف احتمالاً استفاده از تکنیک نمایش ظرفیت بالا (HTS) امکان پذیر شده است چرا که تکنیک HTS قادر است تعداد زیادی از ترکیب ها را براساس فعالیت آنها در مقابل یک هدف معین به سرعت مورد آزمایش قرار دهد. با این وجود، HTS به جای افزایش بازدهی ساخت دارو آن را تا میزان زیادی کاهش می دهد. یکی از دلایل این است که تعداد زیادی از داوطلب های نمایش ممکن است اثرات فنوتیپی غیر قابل قبولی مانند سمی بودن و تعداد روابط داشته باشند که این موارد می توانند در مراحل بعدی کشف دارو هزینه ی تأیید را افزایش دهند [۱۶۳]. تکنیک"صید هدف" [۱۰۹] با استفاده از تکنیک های محاسباتی برای نمایش مستقیم مولکول ها از نظر اثرات فنوتیپ نامطلوب می تواند نقص های فوق الذکر را بر طرف کند. در [۱۹۰]، توضیح مبسوطی از اینگونه روش ها مثل مدل های چند ـ خاصیتی بایسیان [۱۴۹]، رتبه بندی SVM [۱۸۸]، Cascade SVM [۸۴، ۱۸۸]، و Ranking perceptron [۱۸۸، ۶۲] ارائه می دهیم.

۴,۲. برنامه های کاربردی Web

وب جهانی طبیعتاً به شکل یک گراف سازماندهی شده است که در آن صفحات وب همان گره ها هستند و لینک ها معادل کران ها هستند. ساختار اتصال وب حجم زیادی از اطلاعات که قابل استفاده در فعالیت های گوناگون استخراج داده ها هستند را در خود نگه می دارد. مشهورترین برنامه کاربردی که از ساختار اتصال وب استفاده می کند الگوریتم pageRank [۱۵۱،۲۹] است. این الگوریتم یکی از اسرار کلیدی در موفقیت موتور جستجوی گوگل است. ایده ی اصلی پشت الگوریتم pageRank این است که اهمیت یک صفحه روی وب با توجه به تعداد و اهمیت هایپر لینک های معرف آن قابل ارزیابی است. هدف مستقیم الگوریتم طراحی یک موج سوار تصادفی است که لینک های روی صفحات را با احتمال برابر دنبال می کند. بدیهی است که موج سوار به صفحاتی که دارای مسیرهای متعددی هستند بیشتر از سایر صفحات می رسد. تفسیر مستقیم رتبه بندی صفحه عبارتست از: احتمال اینکه یک موج سوار تصادفی در عبور تصادفی به یک صفحه مشخص برسد. بنابراین، رتبه بندی صفحه در اصل یک توزیع احتمال روی صفحات وب تشکیل می دهد به طوری که مقدار رتبه بندی صفحه روی تمام صفحات وب معادل ۱ است. به علاوه، گاهی اوقات معبر مخابراتی اضافه می کنیم، که در آن می توانیم هر یک از صفحات وب را کاملاً تصادفی جابه جا کنیم.

فرض کنید A مجموعه ای از کران ها در گراف باشد. فرض کنید π_i بیانگر احتمال حالت پایای گره i در عبور تصادفی باشد و $p=[p_{ij}]$ نیز معرف ماتریکس انتقال برای فرآیند عبور تصادفی باشد. فرض کنید $\propto$ بیانگر احتمال معبر مخابراتی در یک مرحله معین و q_i بیانگر مقدار i ام بردار احتمال برای تمام گره هایی باشد که احتمال حضور معبر مخابراتی در گره i در هر مرحله ی معین را مشخص می کنند. در حال حاضر، فرض را بر این می گذاریم که همه مقادیر q_i یکسان و برابر با 1/n باشند، که n معادل تعداد کلی گره ها است. آنگاه برای گره معین i می توانیم رابطه ی حالت پایای زیر را استخراج کنیم:

$$\pi_i = \sum_{j:(j,i)\epsilon A} \pi_j . \rho_{ji} . (1-\propto) + \propto . q_i \qquad (۲۰۲)$$

لازم به ذکر است که می توانیم این معادله را برای هر یک از گره ها استخراج کنیم؛ که موجب یک سیستم خطی از معادله های مربوط به احتمال جابه جایی می شود. راه حل ها برای این سیستم باعث ایجاد بردار رتبه بندی صفحه π می شود. این سیستم خطی دارای n متغیر و n محدودیت متفاوت است و به همین خاطر می توان آن را در فضای n^2 در بدترین حالت ممکن تعریف کرد. راه حل برای این سیستم خطی نیازمند عملیات ماتریکس است که در تمام گره ها دست کم درجه دو (و حداکثر درجه ۳) است. این می تواند در عمل بسیار پرهزینه باشد. البته، از آنجایی که رتبه بندی صفحه باید در مرحله گروهی فقط یک بار محاسبه و ارزیابی شود، اجرای آن با استفاده از تعداد اندکی تکنیک های ماتریکس دارای طراحی دقیق امکان پذیر خواهد بود. الگوریتم pageRank [۱۵۱،۲۹] از یک رویکرد تناوبی استفاده می کند که بردارهای اصلی ماتریکس اتصال متعارف شبکه را محاسبه می کند. توضیح درباره ی الگوریتم رتبه بندی صفحه [Page Rank] را می توانید در [۱۵۱] مشاهده کنید.

لازم به ذکر است که الگوریتم pageRank در طی فرآیند رتبه بندی فقط به ساختار اتصال توجه می کند و فاقد هرگونه اطلاعاتی درباره محتوی صفحات اصلی وب است. یک مفهوم کاملاً نزدیک مربوط به رتبه بندی حساس به موضوع [۹۵] است که در آن از موضوعات صفحات وب در فرآیند رتبه بندی استفاده می کنیم. ایده اصلی در این گونه روش ها، ایجاد امکان معبر مخابراتی اختصاصی (یا جهش های اختصاصی) در طی فرایند گردش تصادفی است. در هر مرحله از عبور تصادفی مجاز به یک جابه جایی به مجموعه S صفحات که مرتبط با موضوع جستجو هستند، خواهیم بود. در غیر اینصورت، عبور تصادفی به شیوه ی استاندارد خود با احتمال ($\propto$ ـ ۱) ادامه پیدا می کند. این عمل به سادگی با تعدیل بردار $q=(q_1...q_n)$ محقق می شود، به طوری که اجزاء و مؤلفه های مناسب در این بردار را برابر با ۱ و سایر مؤلفه ها را برابر با ۰ قرار می دهیم. آخرین احتمالات حالت پایا با این عبور تصادفی تعدیل شده. رتبه بندی صفحه حساس به موضوع را تعیین می کند. هر چه احتمال $\propto$ بیشتر باشد، آخرین رتبه بندی صفحه بیشتر به سمت مجموعه S سوق پیدا می کند. از آنجایی که هر بردار خصوصی سازی حساس به موضع نیازمند ذخیره یک بردار رتبه بندی بسیار بزرگ است، ممکن است با استفاده از چند نمونه یا صفحات معتبر اقدام به پیش محاسبه بردار به شیوه ای محدود کند. هدف این است که از تعداد محدودی از این بردارهای خصوصی سازی q استفاده کنیم و بردارهای رتبه بندی صفحه اختصاصی متناظر π را برای این

صفحات معتبر تعیین کنیم. ترکیب سنجیده ای از این بردارهای مختلف رتبه بندی اختصاصی (برای صفحات معتبر) به منظور تعیین پاسخ به مجموعه پرس و جوی معین مورد استفاده قرار می گیرد. چند نمونه از این رویکردها در [۱۰۸، ۹۵] مورد بحث قرار گرفته اند. البته، این رویکرد از نظر سطح غیر یکنواخیت که می تواند در آن اختصاصی سازی را اجرا کنید با محدودیت هایی مواجه است. در [۷۹] نشان داده شده است که رتبه بندی کاملاً اختصاصی که در آن می توانیم عبور تصادفی را به سمت مجموعه ای دلخواه از صفحات وب سوق دهیم، همواره نیازمند فضای دست کم درجه دوم در بدترین حالت است. بنابراین، رویکرد معرفی شده در [۷۹] نشان می دهد که استفاده از نمونه برداری Monte-card می تواند الزامات و محدودیت های فضا را به شدت کاهش دهد بدون اینکه کیفیت را تحت تأثیر قرار دهد. روش مطرح شده در [۷۹] نمونه های Monte-carlo عبورهای تصادفی ویژه گره را پیش ذخیره می کند، که به عنوان اثر انگشت نیز شناخته می شوند. در [۷۹] نشان داده شده است که در فضای محدود می توان با استفاده از این اثر انگشت ها به سطح بالایی از دقت دست پیدا کرد. کار بعدی که در [۲۱، ۱۷۵، ۸۷، ۴۲] ارائه شده بر مبنای این ایده در بسیاری از طرح ها بنیان گذاری شده و نشان داده که این گونه تکنیک های رتبه بندی اختصاص دنیامیک را می توان کارآمدتر و مؤثرتر کرد. بررسی های مبسوطی از تکنیک های مختلف برای محاسبه رتبه بندی صفحه در [۲۰] قابل مشاهده است.

سایر رویکردهای مرتبط شامل استفاده از مقیاس هایی مثل زمان برخورد برای تعیین و رتبه بندی مجاورت متن ـ محور گره ها است. زمان برخورد بین گره i و گره j به صورت تعداد جهش های مورد انتظار که یک موج سوار نیاز دارد تا از گره j به گره i برسد، تعریف می شود. بدیهی است که زمان برخورد فقط تابعی از طول کوتاه ترین مسیر نیست، بلکه تابعی از تعداد مسیرهای ممکن که از گره i تا گره j وجود دارند نیز هست. از این رو، زمان برخورد در مقایسه با استفاده از فواصل کوتاه ترین مسیر مقیاس بهتری برای تعیین شباهت بین موارد اتصال یافته، است. نسخه کوتاه شده زمان برخورد، تابع حقیقی را به محدود کردن آن به فاصله هایی که در آن ها زمان برخورد کمتر از آستانه تعیین شده است، مشخص می کند. هرگاه زمان برخورد بالاتر از آستانه تعیین شده باشد، تابع را به سادگی در مقدار آستانه قرار می دهیم. الگوریتم های سریع برای محاسبه شکل کوتاه شده زمان برخورد در [۱۶۴] مورد بررسی قرار گرفته اند. موضوع قابلیت محاسبه در الگوریتم های عبور تصادفی بسیار جدی و با اهمیت است زیرا این گراف ها بزرگ و دینامیک هستند و ما تمایل داریم قابلیت و توانایی رتبه بندی سریع انواع ویژه ای از پرس و جو را کسب کنیم. روش ارائه شده در [۱۶۵] تکنیک باز ـ رتبه بندی دینامیک سریعی معرفی می کند برای زمانی که واکنش کاربر به ثبت رسیده است. مسئله مرتبط با آن، بررسی رفتار عبورهای تصادفی از طول های ثابت و معین است.

پرس و جوی ترکیبی را می توان "نسخه وارونه" زمان برخورد در نظر گرفت، که تعداد جهش ها را ثابت می کنیم و سعی می کنیم به جای تعداد جهش های برخورد تعداد برخوردها را تعیین کنیم. یکی از مزیت های این کار این است که به طور خودکار فقط عبورهای تصادفی کوتاه شده را که در آن طول عبور پایین تر از آستانه تعیین شده h است را در نظر می گیرد؛ همچنین، به لحاظ بررسی عبورهای مختلف به شیوه ای یکپارچه؛ تعیین

بهتری در مقایسه با زمان برخورد کوتاه شده است. روش ارائه شده در [۲۰۳] گره هایی را مشخص می کند که با استفاده از یک چارچوب ترکیب مجاورهای محلی به نام LONA (Local Neigborhood Aggregation)، دارای بالاترین مقادیر K - بیشینه در تمام مجاورهای جهش - h هستند. این چارچوب از ویژگی های مکانی در فضای شبکه برای ایجاد یک شاخص کار آمد برای پرس و جو بهره می گیرد.

ایده دیگر برای تعیین رتبه بندی معتبر به مدل مرجع [۱۱۸] تعلق دارد. تکنیک رتبه بندی صفحه با استفاده از رفتار اتصال به صورت نشانه مرجع به تعیین مرجع می پردازد. روش مطرح شده در [۱۱۸] پیشنهاد می کند که صفحات وب یکی از دو نوع زیر خواهند بود:

- **قطب ها** صفحاتی هستند که به صفحات معتبر متصل می شوند.
- **مرجع ها** صفحاتی هستند که به وسیله قطب های معتبر متصل می شوند.

هر امتیاز، قطب ها و مرجع ها را مطابق با اعتبار آنها برای قطب بودن یا مرجع بودن به همراه دارد. امتیازات قطب ها بر امتیازات مرجع ها اثر می گذارد و بالعکس. از رویکرد متناوبی به منظور محاسبه امتیازات قطب ها و مرجع ها استفاده می شود. الگوریتم HITS که در [۱۱۸] پیشنهاد شده است از این دو امتیاز برای محاسبه قطب ها و مرجع های موجود در گراف وب استفاده می کند.

بسیاری از این برنامه های کاربردی در گراف های دینامیک که گره ها و کران های گراف در طول زمان دریافت می شوند، مطرح می شوند. به عنوان مثال، در مورد شبکه اجتماعی که در آن لینک های جدید به طور پی در پی و مداوم ایجاد می شوند، ارزیابی رتبه بندی صفحه ذاتاً یک مسئله دینامیک (پویا) است. از آنجایی که الگوریتم رتبه بندی صفحه به شدت وابسته به رفتار گردش های تصادفی است، الگوریتم رتبه بندی زنجیره ای صفحه [۱۶۶]، گره ها را به طور جداگانه نمونه برداری می کند به این منظور که گردش های تصادفی کوتاه از هر یک از گره ها ایجاد کند. این گردش ها را می توان بعداً در هم ادغام کرد تا گردش ها تصادفی طولانی تری ایجاد شود. با راه اندازی چندین گردش تصادفی به این شکل، می توان رتبه بندی صفحه را با کارایی بالا ارزیابی کرد؛ زیرا رتبه بندی صفحه صرفاً احتمال مشاهده یک گره در گردش تصادفی است و الگوریتم نمونه برداری می تواند این فرآیند را به خوبی شبیه سازی کند. چالش جدی پیش روی این الگوریتم این است که ممکن است در طی فرآیند گردش تصادفی گیر کند. این به این خاطر است که فرآیند نمونه برداری هم گره ها و هم کران ها را به عنوان نمونه انتخاب می کند و احتمال یک کران به گونه ای پیموده شود که نقطه انتهایی آن در نمونه گره موجود نباشد. علاوه بر این، مجاز به عبور تکراری از گره ها به منظور حفظ و تداوم تصادفی بودن نیستیم. این گره هایی گیر افتاده را می توان با نگه داشتن حساب مجموعه S از گره های نمونه برداری شده که گردش آنها قبلاً برای تعمیم گردش تصادفی به کار گرفته شده، کنترل کرد. کران های جدید از گره های گیر افتاده و گره های مجموعه S نمونه برداری می شوند. از این نمونه ها استفاده می شود به این منظور که تا جایی که امکان دارد گردش ها تعمیم داده شوند. اگر نقطه ـ انتهایی جدید به شکل یک گره نمونه برداری شده ظاهر

شود، آن گاه فرآیند ادغام گره ها را ادامه می دهیم. در غیر این صورت، فرآیند نمونه برداری کران ها از مجموعه S و تمام گره های گیر افتاده ای که از آخرین گردش تا به حال مشاهده شده اند را تکرار می کنیم.

تحلیل لوگ های جریان پرس و جو، رویکرد دیگری است که در چارچوب استخراج گراف عموماً با آن مواجه می شویم لازم به ذکر است که شیوه ای متداول که بسیاری از کاربرها برای جهت یابی در وب به کار می گیرند استفاده از موتورهای جستجو برای شناسایی صفحات وب و پس از آن کلیک روی بعضی از هایپر لینک های موجود در نتایج جستجو است. از رفتار گراف های به دست آمده می توان برای تعیین پراکندگی موضوع و ارتباط های معنایی بین موضوعات مختلف استفاده کرد.

در بسیاری از برنامه های کاربردی وب، تعیین خوشه های صفحات وب یا بلاگ ها سودمند خواهد بود. به این منظور، کمک گرفتن از ساختار اتصال وب می تواند مفید واقع شود. تکنیک متداولی که اغلب برای خوشه بندی اسناد وب مورد استفاده قرار می گیرد تکنیک تخته پوش کردن است [۸۲، ۳۲]. در این حالت، از رویکرد نشانه ـ مینیمم برای تعیین نواحی به شدت پیوسته در وب مورد استفاده قرار می گیرد. علاوه بر این، از هر یک از تکنیک های ایجاد به ظاهر ـ دسته ها [۱۵۳،۱۴۸، ۵] می توان برای تعیین نواحی متراکم گراف بهره گرفت.

شبکه اجتماعی. شبکه های اجتماعی در اصل گراف های بسیار بزرگی هستند که با توجه به افرادی که به شکل گره ها ظاهر می شوند و لینک هایی که متناظر با ارتباطات یا روابط بین این افراد هستند تعریف می شوند. از لینک های موجود در شبکه اجتماعی می توان برای تعیین جوامع مرتبط، اعضاء با مجموعه های تخصصی ویژه، و جریان اطلاعات در شبکه اجتماعی استفاده کرد. این کاربردها را یک به یک مورد بررسی قرار خواهیم داد.

مسئله بررسی جامعه در شبکه های اجتماعی به مسئله خوشه بندی گره در گراف های خیلی بزرگ مرتبط است. در این حالت مایلیم خوشه های متراکم گره ها را بر اساس ساختار اصلی اتصال مشخص کنیم [۱۵۸]. شبکه های اجتماعی به واسطه اندازه بزرگ گراف های اصلی به چالش ویژه در مسئله خوشه بندی تبدیل شده اند. مانند گراف های وب، هر یک از ورش های ایجاد تخته پوش یا به ظاهر ـ دسته ها [۱۵۳، ۱۴۸، ۸۲، ۳۲، ۵] را می توان برای تعیین جوامع مرتبط در شبکه به کار گرفت. تکنیکی در [۱۶۷] به منظور استفاده از محرک های جریان تصادفی در تعیین خوشه ها در گراف های اصلی معرفی شده است. روش برای تعیین ساختار خوشه بندی با استفاده از ساختار ـ ایجن ماتریکس اتصال به منظور تعیین ساختار اجتماع، در [۱۶۴] ارائه شده است. یکی از ویژگی های مهم شبکه های بزرگ این است که اغلب می توان آنها را با توجه به ماهیت گراف های فرعی زیر بنایی توصیف کرد، در [۲۷]، تکنیکی برای محاسبه تعداد گراف های فرعی گونه خاصی از شبکه های بزرگ معرفی شده است. نشان داده شده است که این توصیف در خوشه بندی شبکه های بزرگ مفید است. این دقت و وضوح با استفاده از ویژگی های مکانی محقق نمی شود. بنابراین، این رویکرد برای بررسی اجتماع در شبکه های فراگیر و بزرگ نیز قابل استفاده است. مسئله بررسی اجتماعی به ویژه در مورد تحلیل دنیامیک. شبکه های در تکامل که در آن تلاش می کنیم چگونگی تغییر جوامع شبکه در طول زمان را مشخص کنیم. بعضی از روش

های اخیر برای اینگونه مسائل را می توانید در [۱۳۱،۱۳۵،۱۷۱،۱۷۳، ۱۱۷، ۷۴، ۶۹، ۵۰، ۱۶، ۹] مشاهده کنید. فعالیت موجود در [۹] نیز به بررسی این مسئله در زمینه زنجیره های گراف در حال تکامل می پردازد. بسیاری از این تکنیک ها به ارزیابی مسئله بررسی اجتماعی و بررسی تغییر در یک چارچوب می پردازد. با این کار قادر خواهیم بود تغییرات در شبکه اصلی را به شیوه کوتاه و خلاصه ارائه دهیم.

الگوریتم های خوشه بندی گره کاملاً مرتبط با مفهوم تحلیل مرکزیت در شبکه ها است. به عنوان مثال، تکنیکی که در [۱۵۸] مورد بحث واقع شده، از یک رویکرد K-medoids استفاده می کند که K نقطه مرکزی شبکه را نشان می دهد. این نوع رویکرد ها در شبکه های مختلف مفید واقع می شوند حتی اگر چارچوب شبکه ها نیز با هم متفاوت داشته باشند. در مورد شبکه های اجتماعی، این نقطه های مرکزی به طور معمول اعضا کلیدی در شبکه هستند که به خوبی به سایر اعضاء اجتماع متصل شده اند. از تحلیل مرکزیت می توان برای تعیین نقطه های مرکزی در جریان های اطلاعات نیز بهره گرفت. از این رو، بدیهی است که گونه مشابهی از الگوریتم تحلیل ساختاری می تواند منجر به دیدگاه های متفاوت در شبکه های مختلف شود.

ارزیابی مرکزیت ارتباط نزدیکی با مسئله جریان اطلاعات منتشر شده در شبکه های اجتماعی دراد. مشاهده شده است که اکثر تکنیک های تحلیل جریان ویروسی که اخیراً طراحی شده اند [۱۴۷، ۱۲۷، ۴۰] در چارچوب انواع مختلفی از برنامه های کاربردی مرتبط با جریان اطلاعات شبکه اجتماعی قابل استفاده اند. این کاربرد به این خاطر است که برنامه های کاربردی مرتبط با جریان اطلاعات با مدل های رفتاری مشابه با انتشار ویروس ها قابل درک هستند. این برنامه های کاربردی عبارتند از: (۱) ما تمایل داریم مؤثرترین اعضاء شبکه اجتماعی را مشخص کنیم؛ یعنی اعضایی که باعث جریان بیشینه اطلاعات به خارج می شود. (۲) اطلاعات در رفتار اجتماعی اغلب به شیوه ای مشابه اپیدمی سرریز می کند و منتشر می شود. ما تمایل داریم میزان سر ریز اطلاعات از طریق شبکه اجتماعی را اندازه گیری کنیم و اثر منابع مختلف بر اطلاعات را تعیین کنیم. هدف این است که نظارت موجب ارتقاء ارزیابی اولیه از جریان های اطلاعات شود و این برای فردی که بتواند آن را ارزیابی کند مفید خواهد بود. رفتار سرریز و انتشار به ویزه در مورد گراف بلاگ که در آن رفتار سرریز و انتشار به شکل لینک های افزوده در طول زمان بازتاب می یابد، قابل مشاهده است. از آنجایی که کنترل و نظارت برتمام بلاگ ها به طور همزمان امکان پذیر نیست، به حداقل رساندن هزینه نظارت بر بلاگ های مختلف با فرض یک هزینه ثابت به ازای هر گره مطلوب خواهد بود، این مسئله NP ـ دشوار است [۱۲۷]، زیرا مسئله پوشش ـ رأس را می توان به آن تقلیل داد. ایده اصلی در [۱۲۸]، استفاده از اکتشاف تقریبی به منظور به حداقل رساندن هزینه است. این رویکرد به طرح بلاگ محدود نمی شود بلکه برای سایر طرح ها از جمله نظارت بر تبادل اطلاعات در شبکه های اجتماعی، و نظارت بر قطع جریان در شبکه های ارتباطی نیز قابل استفاده است. (۳) ما می خواهیم شرایطی که منجر به نیاز فوری برای انتقال غیر کنترل شده اطلاعات می شوند را مشخص کنیم. چند تکنیک برای توصیف این شرایط، در [۱۸۷، ۴۰] مورد بررسی قرار گرفته اند. فعالیت ارائه شده در [۱۸۷] با ساختار ماتریکس تجانب برای میزان انتقال پذیری به منظور اندازه گیری میزان انتشار اطلاعات در شبکه اصلی از اهمیت

ویژه ای برخوردار است. در [۱۸۷] نشان داده شده است که ساختار ایجن ماتریکس تجانب می تواند مستقیماً با آستانه تعیین شده برای اپیدمی ارتباط داشته باشد.

سایر برنامه های شبکه کامپیوتری. اکثر این تکنیک ها را می توان برای انواع دیگری از شبکه ها مانند شبکه های ارتباطی نیز مورد استفاده قرار داد. تحلیل ساختاری و توان شبکه های ارتباطی تا حد زیادی طراحی گراف اصلی شبکه بستگی دارد. طراحی دقیق گراف اصلی می تواند در جلوگیری از نقص های شبکه، تراکم، و سایر نقاط ضعف در کل شبکه مؤثر واقع شود. به عنوان مثال، تحلیل مرکزیت [۱۵۸] در مورد یک شبکه اجتماعی برای تعیین نقاط حساس نقص و ناکارآمدی مورد استفاده خواهد بود. همچنین، تکنیک هایی برای انتشار جریان اطلاعات در شبکه های اجتماعی برای مدل انتشار ویروس در شبکه های اجتماعی نیز قابل استفاده خواهد بود. تفاوت عمده در این است که احتمال آلودگی به ویروس را در امتداد یک کران در شبکه اجتماعی طراحی می کنیم به جای اینکه تلاش کنیم احتمال جریان اطلاعات را در امتداد یک کران در شبکه اجتماعی مدل سازی کنیم.

اکثر تکنیک های قابلیت دسترسی [۱۸۴، ۵۴، ۵۳، ۴۹، ۴۸، ۱۰] برای تعیین تصمیمات استخراج بهینه در شبکه های کامپیوتری نیز کاربرد خواهد داشت. این به مسئله تعیین اتصال ـ گره جفت محور [۷] در شبکه های کامپیوتری نیز مرتبط است. تکنیک ارائه شده در [۷] از خلاصه سازی بر مبنای فشرده کردن برای ایجاد یک شاخص اتصال مؤثر در گراف های فراگیر ذخیره شده روی دیسک استفاده می کند. این کار می تواند در شبکه های ارتباطی مفید واقع شود که در آنها نیاز است تعداد مینیمم کران هایی که باید به منظور قطع ارتباط یک جفت ویژه از گره ها حذف شوند را مشخص کنیم.

۴,۳. مکان یابی حفره های نرم افزاری

یکی از کاربردهای طبیعی الگوریتم های استخراج گراف مربوط به مکان یابی حفره های نرم افزاری از نقطه نظر اعتبار و دشواری نرم افزار یکی از برنامه های کاربردی مهم به شمار می آید. جریان کنترل برنامه ها را می توان در قالب گراف های پیام (خبر) مدل سازی کرد. هدف تکنیک های مکان یابی حفره های نرم افزاری، استخراج این گراف های پیام به منظور تعیین حفره ها در برنامه های نرم افزاری است. گراف های پیام به گروه تقسیم می شوند:

- **گراف های پیام استاتیک (ثابت)** از کد منبع یک برنامه مشخص قابل استنباط است. تمام روش ها، پروسه ها، و عملکردها در برنامه به شکل گره ها وجود دارند و رابطه بین روش های مختلف به صورت کران ها تعریف می شوند. همچنین، تعیین گره ها برای عناصر داده ها، و طراحی روابط بین عناصر داده های مختلف و کران ها نیز امکان پذیر است. در مورد گراف های پیام استاتیک، اغلب استفاده از

نمونه های واقعی ساختار برنامه به منظور تعیین قسمت هایی از نرم افزار که ممکن است در آن پیشامدهای غیر واقعی روی دهد، امکان پذیر است.

- **گراف های پیام دینامیک (پویا)** در طی اجرای برنامه ایجاد می شوند و بیانگر ساختار تقاضا هستند. به عوان مثال، یک پیام از پروسه ای به پروسه دیگر موجب کرانی می شود که معرف رابطه تقاضا بین دو پروسه است. این گونه گراف های پیام می توانند در برنامه های نرم افزاری فراگیر به شدت بزرگ و گسترده باشند، زیرا این برنامه ها شامل هزاران تقاضا بین پروسه های مختلف می باشند. در اینگونه موارد، از تفاوت در رفتار ساختاری، فراوانی توالی تقاضا های موفق و ناموفق می توان برای مکان یابی حفره های نرم افزاری استفاده کرد. این گراف های پیام به ویژه در مکان یابی حفره های تصادفی که ممکن است در بعضی تقاضاها و نه در همه آنها روی دهند، سودمند هستند.

متذکر می شویم که مکان یابی حفره ها از نظر انواع خطاهایی که می تواند به دام اندازد جامع و فراگیر نیست. به عنوان مثال، خطاهای منطقی در یک برنامه که ناشی از ساختار برنامه نباشند و توالی یا ساختار اجرای روش های مختلف را تحت تأثیر قرار ندهند قابل مکان یابی با این تکنیک ها نیستند. علاوه بر این، مکان یابی حفره نرم افزاری مهارتی دقیق محسوب نمی شود. تا اندازه ای می توان از این تکنیک برای ایجاد تست نرم افزاری که متخصص حفره های موجود باشد استفاده کرد، و این تست ها می توانند از این تکنیک برای تصحیح ها و اصلاحات مرتبط استفاده کنند.

یکی از موارد جالب، حالتی است که در آن اجراهای مختلف برنامه منجر به ساختار، توالی و فراوانی تکنیک هایی می شود که مختص ناکامی ها و موفقیت های اجرای نهایی برنامه می شود. این ناکامی ها و موفقیت ها ممکن است نتیجه خطاهای منطقی باشد که موجب تغییراتی در ساختار و توالی پیام ها می شود. در اینگونه موارد، مکان یابی حفره نرم افزاری به صورت یک مسئله طبقه بندی قبل طراحی است. مرحله اول شامل ایجاد گراف های پیام از تکینک های اجرایی است. این پروسه با جستجوی تکنیک های اجرای برنامه در فرآیند تست و آزمایش محقق می شود. لازم به ذکر است که این دسته از گراف های پیام ممکن است برای کار با الگوریتم های استخراج گراف بیش از اندازه غول پیکر و گسترده باشند. اندازه های بزرگ گراف های پیام، چالش برای پروسه های استخراج گراف پدید می آورد؛ زیرا الگوریتم های استخراج گراف اغلب برای گراف های نسبتاً کوچک طراحی شده اند، در حالی که اندازه گراف های پیام ممکن است بسیار بزرگ باشد. بنابراین، کاهش اندازه گراف های پیام با استفاده از رویکرد مقایسه ـ محور می تواند راه حل طبیعی برای چالش فوق باشد. این کاهش به طور طبیعی منجر به مفقود شدن اطلاعات می شود و در بعضی از موارد که صدمه به اطلاعات همه جانبه و گسترده باشد باعث ضعف و ناتوانی در استفاده مؤثر از رویکرد مکان یابی می شود.

گام بعدی شامل استفاده از الگوریتم های استخراج گراف فرعی در داده های آموزشی به منظور تعیین الگوهایی است که به طور مکرر در تکنیک های اجرایی پر اشتباه ظاهر می شوند. باید خاطر نشان کنیم که این تا حدودی شبیه به تکنیکی است که اغلب در طبقه کننده های اصل ـ محور که سعی می کنند الگوهای خاص و شرایط

ویژه را به برچسب های طبقه خاص متصل کنند، به کار گرفته می شود. سپس این الگوها با روش های مختلفی همراه می شوند و برای ایجاد فرآیند رتبه بندی روش ها و تابع های گوناگون در برنامه های دارای حفره های نرم افزاری مورد استفاده قرار می گیرند. این کار به آشنایی و درک حفره های موجود در برنامه های زیر بنایی نیز منجر خواهد شد.

باید اشاره کنیم که فرآیند فشرده سازی در ایجاد توانایی برای پردازش مؤثر گراف های اصلی از اهمیت ویژه ای برخوردار است. یکی از روش های طبیعی برای کاهش اندازه گراف های متناظر این است که گره های چندگانه در گراف پیام را با یک گره مجزا انطباق دهیم. به عنوان مثال، در فرآیند کاهش کلی، تمام گره های موجود در گره پیام را با هم تطبیق می دهیم که با همان روش در گره گراف فشرده متناظر است. از این رو، تعداد کلی گره ها در نمودار حداکثر برابر با تعداد روش هاست. به منظور کاهش اندازه گراف پیام از چنین تکنیکی در [۱۳۶] استفاده شده است. دومین روشی که ممکن است مورد استفاده قرار گیرد فشرده سازی ساختارهای اجرایی تناوبی مانند لوپ ها (حلقه ها) در یک گراف مجزا است. این یک رویکرد طبیعی است، در حالی که ساختار اجرایی تناوبی یکی از متداول ترین بخش های گراف های پیام است. تکنیک دیگر، تبدیل نمودارهای درختی فرعی به گره های مجزاست. انواع مختلفی از استراتژی های مکان یابی که از این گونه تکنیک های تبدیلی و کاهشی استفاده می کنند در [۶۷، ۷۲،۶۸] بررسی شده اند.

در نهایت، گراف های تبدیل شده استخراج می شوند به این منظور که ساختارهای تشخیص دهند برای امکان یابی حفره ها تعیین شوند. روش ارائه شده در [۷۲] بر اساس تعیین نمودارهای درختی فرعی از داده ها عمل می کند. به بیان دقیق تر، این روش تمام نمودارهای درختی فرعی که در اجرا های ناموفق پرتکرارند ولی در اجراهای صحیح پرتکرار نیستند را پیدا می کند. سپس، از آنها برای ایجاد قوانینی که ممکن است برای نمونه های خاصی از راه اندازی های برنامه های طبقه بندی مورد استفاده قرار گیرند، استفاده می شود. مهم تر اینکه، این قوانین شناخت و درک بهتر از تصادفی بودن حفره ها در اختیار ما می گذارد و این شناخت می تواند برای تأیید اصلاح خطا های زیربنایی مورد استفاده قرار گیرد.

تکنیک فوق برای پیدا کردن ویژگی های ساختاری تکنیک اجرایی که در ایزوله کردن حفره های نرم افزاری مورد استفاده قرار می گیرد طراحی شده است. با این وجود، در اکثر موارد ویژگی های ساختاری ممکن است تنها خاصیت هایی باشند که با مکان یابی حفره ها ارتباط دارند. به عنوان مثال، خاصیت مهمی که ممکن است در تعیین حفره های به کار گرفته شود "فراوانی نسبی" تقاضای روش های مخلف است. مثلاً، تقاضا هایی که دارای حفره اند ممکن است روش ویژه ای که پر تکرار تر از سایرین است را فرا بخوانند. شیوه ای معمول برای یادگیری این روش این است که بارهای کران را به گراف پیام مربوط کنیم. این بارها با فراوانی تقاضا متناظرند. سپس، از این بارهای کران برای تحلیل تقاضا هایی که بیشترین تناسب را با متمایز کردن تکنیک های اجرایی موفق و نا موفق دارند، استفاده می کنیم. تعدادی از روش های موجود برای این گروه از تکنیک ها در [۶۸، ۷۶] بررسی شده اند.

لازم به ذکر است که ساختار و فراوانی دو جنبه ی متفاوت از داده ها هستند که برای اجرای مکان یابی حفره ها می توان آنها را در هم ادغام کرد. بنابراین، تلفیق این رویکردها به منظور بهبورد فرآیند مکان یابی کاملاً منطقی به نظر می رسد. تکنیک های ارائه شده در [۶۸، ۶۷] امتیازی برای خاصیت های ساختار ـ محور و فراوانی ـ محور به وجود می آورد. آنگاه، ترکیب این امتیاز ها برای فرآیند مکان یابی حفره مورد استفاده قرار می گیرد. نشان داده شده است [۶۸، ۶۷] که این رویکرد مؤثر تر و کار آمدتر از استفاده از یکی از دو خاصیت (یا ساختار و یا فراوانی) است.

ویژگی مهم دیگری که در کارهای آینده قابل بررسی و پژوهش است، تحلیل توالی پیام های برنامه است به جای اینکه به سادگی به تحلیل ساختار پیام دینامیک یا فراوانی پیام های روش های مختلف بپردازیم. برخی کارهای اولیه [۶۴] در این راستا نشان می دهد که استخراج توالی می تواند اطلاعات فوق العاده ای را برای مکان یابی نرم افزار حتی با بهره گیری از روش های ساده به رمز در آورد.

با این وجود؛ این تکنیک های استخراج گراف پیشرفته برای تلفیق این توالی اطلاعاتی بهره نمی گیرد. بنابراین، این می تواند مسیری پربار و پرفایده برای تحقیق های آینده در زمینه ی ادغام اطلاعات زنجیره ای و متوالی با تکنیک های استخراج گراف موجود باشد.

تحلیل کد منبع استاتیک به جای گراف های پیام دنیامیک یکی دیگر از خطوط کلی تحلیل است. در چنین شرایطی، توجه به گروه های خاصی از حفره های نرم افزاری به جای سعی در ایزوله کردن منشأ خطای اجرایی منطقی تر خواهد بود. به عنوان مثال، حالت های فراموش شده در برنامه های نرم افزار [۴۳] می تواند موجب ضعف و کمبود شود. به عنوان مثال، گزارش وضعیت در برنامه نرم افزاری به همراه یک حالت مفقود شده یکی از مداول ترین حفره های نرم افزاری است. در این شرایط، طراحی تکنیک های مختص به حوزه برای مکان یابی حفره ها طبیعی به نظر برسد. برای این هدف، تکنیک هایی برمبنای گراف های استاتیک وابسته به برنامه به کار گرفته می شوند. این گراف ها متفاوت از گراف های دنیامیکی هستند که در بالا مورد بررسی قرار گرفتند؛ به تعبیری دومی نیازمند اجرای برنامه برای ایجاد گراف هاست، در حالی که در مورد اول گراف های استاتیک به شیوه ای استاتیک ساخته می شوند. گراف های وابسته به برنامه در اصل یک بازنمایی گرافیکی از رابطه های بین روش های مختلف و عناصر داده های یک برنامه ایجاد می کنند. انواع مختلف و متفاوتی از کران ها برای توصیف کنترل و وابستگی های داده ها مورد استفاده قرار می گیرند. گام اول، تعیین قوانین شرطی [۴۳] در برنامه ای است که وابستگی های پرتکرار در پروژه را شرح می دهد. سپس به دنبال تحریک های استاتیکی درون پروژه می گردیم که این قوانین را نقض می کنند. در اکثر موارد، این تحریک ها متناظر با حالت های نادیده گرفته شده در برنامه نرم افزاری هستند.

حوزه مکان یابی حفره نرم افزاری با تعدادی چالش کلیدی روبروست. یکی از چالش های عمده این است که کار در این حوزه عمدتاً بر پروژه های نرم افزاری کوچک تر تمرکز می کند. برنامه بزرگتر و فراگیرتر چالش

محسوب می شوند زیرا ممکن است گراف های پیام متناظر بسیار بزرگ و غول پیکر باشند و فرآیند فشرده سازی گراف ممکن است حجم زیادی از اطلاعات را از دست بدهد. در حالی که بعضی از این چالش ها ممکن است با طراحی تکنیک های استخراج کارآمد تا اندازه ای کم اثر شوند، چندین مزیت نیز ممکن است با استفاده از بازنمایی بهتر "سطح مدل سازی" کسب شوند. به عنوان مثال، گره های گراف را می توان در سطح پایین تری از تجزیه و خردشدگی در فاز مدل سازی بیان کرد. از آنجایی که فرآیند مدل سازی با درک بهتر احتمالات بروز حفره های نرم افزاری محقق می شود (که با فرآیند فشرده سازی خودکار قابل مقایسه است). فرض بر این است که این رویکرد اطلاعات کمتری را در فرآیند مکان یابی حفره ها از دست می دهد. هدف دوم، تلفیق تکنیک های گراف ـ محور با سایر تکنیک های آماری مؤثر [۱۳۷] به منظور ایجاد طبقه بندی کننده های منسجم تر و قوی تر است. در تحقیق آینده، منطقی است انتظار داشته باشیم که تحلیل پروژه های نرم افزاری بزرگتر فقط با استفاده از این تکنیک های تلفیقی که قادرند از ویژگی های مختلف داده های اصلی استفاده کنند، امکان پذیر شود.

۵. نتیجه گیری و تحقیق آینده

در این فصل، یک نمای کلی از برنامه های کاربردی در حوزه استخراج و مدیریت گراف در اختیار شما قرار دادیم. همچنین، ارزیابی از برنامه های کاربردی متداول که از کاربردهای استخراج گراف نشأت می گیرند، ارائه دادیم. بخش اعظم فعالیت ها در سال های اخیر بر گراف های کوچک و قابل ذخیره سازی در حافظه تمرکز کرده اند. بسیاری از چالش های آینده در مورد گراف های بسیار بزرگی که روی دیسک ذخیره می شوند پدید می آید. برنامه های کاربردی دیگری در چارچوب زنجیره های گراف فراگیر و گسترده مطرح می شوند. زنجیره های گراف در چارچوب بعضی از برنامه های کاربردی از قبیل شبکه اجتماعی پدید می آیند که ارتباطات بین گروه های بزرگی از کاربران به شکل گراف ها ذخیره می شوند. این گونه برنامه ها بسیار چالشی هستند زیرا نمی توان کل داده ها را به منظور تحلیل ساختاری روی دیسک مکان یابی کرد. از این رو، به تکنیک های جدیدی برای خلاصه سازی رفتار ساختاری زنجیره های گراف نیاز داریم، و می توان از این تکنیک ها برای انواع مختلفی از نقشه های تحلیلی بهره گرفت. انتظار داریم که تحقیق آینده بر طرح های مقیاس ـ بزرگ و زنجیره ـ محور برای استخراج گراف تمرکز کند.

مراجع

[1] Chemaxon. Screen, Chemaxon Inc., 2005.

[2] Daylight. Daylight Toolkit, Daylight Inc, Mission Viejo, CA, USA, 2008.

[3] Oracle Spatial Topology and Network Data Models 10g Release 1 (10.1) URL: http://www.oracle.com/technology/products/spatial /pdf/10g networkmodeltwp.pdf

[4] Semantic Web Challenge.URL:http://challenge.semanticweb.org/

[5] J. Abello, M. G. Resende, S. Sudarsky, Massive quasi-clique detection. Proceedings of the 5th Latin American Symposium on Theoretical Informatics (LATIN) (Cancun, Mexico). 598-612, 2002.

[6] S. Abiteboul, P. Buneman, D. Suciu. Data on the web: from relations to semistructured data and XML. Morgan Kaufmann Publishers, Los Altos, CA 94022, USA, 1999.

[7] C. Aggarwal, Y. Xie, P. Yu. GConnect: A Connectivity Index for Massive Disk-Resident Graphs,VLDB Conference, 2009.

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

[9] C. Aggarwal, P. Yu. Online Analysis of Community Evolution in Data Streams.SIAM Conference on Data Mining, 2005.

[10] R. Agrawal, A. Borgida, H.V. Jagadish. Efficient Maintenance of Transitive Relationships in Large Data and Knowledge Bases,ACM SIGMOD Conference, 1989.

[11] R. Agrawal, R. Srikant. Fast algorithms for mining association rules in large databases, VLDB Conference, 1994.

[12] S. Agrawal, S. Chaudhuri, G. Das. DBXplorer: A system for keywordbased search over relational databases.ICDE Conference, 2002.

[13] R. Ahuja, J. Orlin, T. Magnanti. Network Flows: Theory, Algorithms, and Applications,Prentice Hall, Englewood Cliffs, NJ, 1992.

[14] S. Alexaki, V. Christophides, G. Karvounarakis, D. Plexousakis. On Storing Voluminous RDF Description Bases. InWebDB, 2001.

[15] S. Alexaki, V. Christophides, G. Karvounarakis, D. Plexousakis. The ICS-FORTH RDFSuite: Managing Voluminous RDF Description Bases. InSemWeb, 2001.

[16] S. Asur, S. Parthasarathy, and D. Ucar. An event-based framework for characterizing the evolutionary behavior of interaction graphs.ACM KDD Conference, 2007.

[17] R. Baeza-Yates, A Tiberi. Extracting semantic relations from query logs. ACM KDD Conference, 2007.

[18] Z. Bar-Yossef, R. Kumar, D. Sivakumar. Reductions in streaming algorithms, with an application to counting triangles in graphs. ACM SODA Conference, 2002.

[19] D. Beckett. The Design and Implementation of the Redland RDF Application Framework.WWW Conference, 2001.

[20] P. Berkhin. A survey on pagerank computing. Internet Mathematics, 2(1), 2005.

[21] P. Berkhin. Bookmark-coloring approach to personalized pagerank computing.Internet Mathematics, 3(1), 2006.

[22] M. Berlingerio, F. Bonchi, B. Bringmann, A. Gionis. Mining GraphEvolution Rules,PKDD Conference, 2009.

[23] S. Bhagat, G. Cormode, I. Rozenbaum. Applying link-based classification to label blogs. WebKDD/SNA-KDD, pages 97–117, 2007.

[24] G. Bhalotia, C. Nakhe, A. Hulgeri, S. Chakrabarti, S. Sudarshan. Keyword searching and browsing in databases using BANKS.ICDE Conference, 2002.

[25] M. Bilgic, L. Getoor. Effective label acquisition for collective classification. ACM KDD Conference, pages 43 51, 2008.

[26] S. Boag, D. Chamberlin, M. F. Fern«andez, D. Florescu, J. Robie, J. Sim«eon. XQuery 1.0: An XML query language. URL:W3C, http://www.w3.org/TR/xquery/, 2007.

[27] I. Bordino, D. Donato, A. Gionis, S. Leonardi. Mining Large Networks with Subgraph Counting.IEEE ICDM Conference, 2008.

[28] C. Borgelt, M. R. Berthold. Mining molecular fragments: Find- ing Relevant Substructures of Molecules.ICDM Conference, 2002.

[29] S. Brin, L. Page. The Anatomy of a Large Scale Hypertextual Search Engine,WWW Conference, 1998.

[30] H.J. Bohm, G. Schneider. Virtual Screening for Bioactive Molecules. Wiley-VCH, 2000.

[31] B. Bringmann, S. Nijssen. What is frequent in a single graph? PAKDD Conference, 2008.

[32] A. Z. Broder, M. Charikar, A. Frieze, M. Mitzenmacher. Syntactic clustering of the web, WWW Conference, Computer Networks, 29(8– 13):1157–1166, 1997.

[33] J. Broekstra, A. Kampman, F. V. Harmelen. Sesame: A Generic Architecture for Storing and Querying RDF and RDF Schema. InISWC Conference, 2002.

[34] H. Bunke. On a relation between graph edit distance and maximum common subgraph.Pattern RecognitionLetters, 18: pp. 689–694, 1997.

[35] H. Bunke, G. Allermann. Inexact graph matching for structural pattern recognition.Pattern Recognition Letters, 1: pp. 245–253, 1983.

[36] H. Bunke, X. Jiang, A. Kandel. On the minimum common supergraph of two graphs. Computing, 65(1): pp. 13 25, 2000.

[37] H. Bunke, K. Shearer. A graph distance metric based on the maximal common subgraph.Pattern Recognition Letters, 19(3): pp. 255–259, 1998.

[38] J. J. Carroll, I. Dickinson, C. Dollin, D. Reynolds, A. Seaborne, K. Wilkinson. Jena: implementing the Semantic Web recommendations. In WWW Conference, 2004.

[39] V. R. de Carvalho, W. W. Cohen. On the collective classification of email "speech acts".ACM SIGIR Conference, pages 345–352, 2005.

[40] D. Chakrabarti, Y. Wang, C. Wang, J. Leskovec, C. Faloutsos. Epidemic thresholds in real networks. ACM Transactions on Information Systems and Security, 10(4), 2008.

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

[42] S. Chakrabarti. Dynamic Personalized Pagerank in Entity-Relation Graphs,WWW Conference, 2007.

[43] R.-Y. Chang, A. Podgurski, J. Yang. Discovering Neglected Conditions in Software by Mining Dependence Graphs.IEEE Transactions on Software Engineering, 34(5):579–596, 2008.

[44] O. Chapelle, A. Zien, B. Sch- olkopf, editors.Semi-Supervised Learning. MIT Press, Cambridge, MA, 2006.

[45] S. S. Chawathe. Comparing Hierachical data in external memory. Very Large Data Bases Conference, 1999.

[46] C. Chen, C. Lin, M. Fredrikson, M. Christodorescu, X. Yan, J. Han, Mining Graph Patterns Efficiently via Randomized Summaries,VLDB Conference, 2009.

[47] L. Chen, A. Gupta, M. E. Kurul. Stack-based algorithms for pattern matching on dags.VLDB Conference, 2005.

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

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

[50] Y. Chi, X. Song, D. Zhou, K. Hino, B. L. Tseng. Evolutionary spectral clustering by incorporating temporal smoothness.KDD Conference, 2007.

[51] C. Chung, J. Min, K. Shim. APEX: An adaptive path index for XML data. InSIGMOD Conference, 2002.

[52] J. Clark, S. DeRose. XML Path Language (XPath). URL:W3C, http://www.w3.org/TR/xpath/, 1999.

[53] 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.

[54] E. Cohen, E. Halperin, H. Kaplan, U. Zwick. Reachability and Distance Queries via 2-hop Labels,ACM Symposium on Discrete Algorithms, 2002.

[55] S. Cohen, J. Mamou, Y. Kanza, Y. Sagiv. XSEarch: A semantic search engine for XML.VLDB Conference, 2003.

[56] M. P. Consens, A. O. Mendelzon. GraphLog: a visual formalism for real life recursion. InPODS Conference, 1990.

[57] D. Conte, P. Foggia, C. Sansone, M. Vento. Thirty Years of Graph Matching in Pattern Recognition. International Journal of Pattern Recognition and Artificial Intelligence, 18(3): pp. 265–298, 2004.

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

[59] B. F. Cooper, N. Sample, M. Franklin, G. Hjaltason, M. Shadmon. A fast index for semistructured data. InVLDB Conference, pages 341–350, 2001.

[60] L.P. Cordella, P. Foggia, C. Sansone, M. Vento. A (Sub)graph Isomorphism Algorithm for Matching Large Graphs.IEEE Transactions on Pattern Analysis and Machine Intelligence, 26(20): pp. 1367–1372, 2004.

[61] G. Cormode, S. Muthukrishnan. Space efficient mining of multigraph streams.ACM PODS Conference, 2005.

[62] K. Crammer Y. Singer. A new family of online algorithms for category ranking. Journal of Machine Learning Research., 3:1025–1058, 2003.

[63] T. Dalamagas, T. Cheng, K. Winkel, T. Sellis. Clustering XML Documents Using Structural Summaries.Information Systems, Elsevier, January 2005.

[64] V. Dallmeier, C. Lindig, A. Zeller. Lightweight Defect Localization for Java. InProc. of the 19th European Conf. on Object-Oriented Programming (ECOOP), 2005.

[65] M. Deshpande, M. Kuramochi, N. Wale, G. Karypis. Frequent Substructure-based Approaches for Classifying Chemical Compounds. IEEE Transactions on Knowledge and Data Engineering, 17: pp. 1036– 1050, 2005.

[66] E. W. Dijkstra. A note on two problems in connection with graphs. Numerische Mathematik, 1 (1959), S. 269 271.

[67] F. Eichinger, K. B- ohm, M. Huber. Improved Software Fault Detection with Graph Mining. Workshop on Mining and Learning with Graphs, 2008.

[68] F. Eichinger, K. B- ohm, M. Huber. Mining Edge-Weighted Call Graphs to Localize Software Bugs. PKDD Conference, 2008.

[69] T. Falkowski, J. Bartelheimer, M. Spilopoulou. Mining and Visualizing the Evolution of Subgroups in Social Networks, ACM International Conference on Web Intelligence, 2006.

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

[71] W. Fan, K. Zhang, H. Cheng, J. Gao. X. Yan, J. Han, P. S. Yu O. Verscheure. Direct Mining of Discriminative and Essential Frequent Patterns via Model-based Search Tree.ACM KDD Conference, 2008.

[72] G. Di Fatta, S. Leue, E. Stegantova. Discriminative Pattern Mining in Software Fault Detection.Workshop on Software Quality Assurance, 2006.

[73] J. Feigenbaum, S. Kannan, A. McGregor, S. Suri, J. Zhang. Graph Distances in the Data-Stream Model.SIAM Journal on Computing, 38(5): pp. 1709–1727, 2008.

[74] J. Ferlez, C. Faloutsos, J. Leskovec, D. Mladenic, M. Grobelnik. Monitoring Network Evolution using MDL.IEEE ICDE Conference, 2008.

[75] M. Fiedler, C. Borgelt. Support computation for mining frequent subgraphs in a single graph.Workshop on Mining and Learning with Graphs (MLG'07), 2007.

[76] M.A. Fischler, R.A. Elschlager. The representation and matching of pictorial structures.IEEE Transactions on Computers, 22(1): pp 67–92, 1973.

[77] P.-O. Fjallstrom. Algorithms for Graph Partitioning: A Survey,Linkoping Electronic Articles in Computer and Information Science, Vol 3, no 10, 1998.

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

[79] D. Fogaras, B. R«acz, K. Csalog«any, T. Sarl«os. Towards scaling fully personalized pagerank: Algorithms, lower bounds, and experiments. Internet Mathematics, 2(3), 2005.

[80] M. S. Garey, D. S. Johnson. Computers and Intractability: A Guide to the Theory of NP-completeness,W. H. Freeman, 1979.

[81] T. Gartner, P. Flach, S. Wrobel. On graph kernels: Hardness results and efficient alternatives.16th Annual Conf. on Learning Theory, pp. 129–143, 2003.

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

[83] R. Giugno, D. Shasha, GraphGrep: A Fast and Universal Method for Querying Graphs.International Conference in Pattern recognition (ICPR), 2002.

[84] S. Godbole, S. Sarawagi. Discriminative methods for multi-labeled classification. PAKDD Conference, pages 22–30, 2004.

[85] R. Goldman, J. Widom. DataGuides: Enable query formulation and optimization in semistructured databases.VLDB Conference, pages 436–445, 1997.

[86] L. Guo, F. Shao, C. Botev, J. Shanmugasundaram. XRANK: ranked keyword search over XML documents.ACM SIGMOD Conference, pages 16– 27, 2003.

[87] M. S. Gupta, A. Pathak, S. Chakrabarti. Fast algorithms for top-k personalized pagerank queries.WWW Conference, 2008.

[88] R. H. Guting. GraphDB: Modeling and querying graphs in databases. In VLDB Conference, pages 297–308, 1994.

[89] M. Gyssens, J. Paredaens, D. van Gucht. A graph-oriented object database model. InPODS Conference, pages 417–424, 1990.

[90] J. Han, J. Pei, Y. Yin. Mining Frequent Patterns without Candidate Generation.SIGMOD Conference, 2000.

[91] S. Harris, N. Gibbins. 3store: Efficient bulk RDF storage. InPSSS Conference, 2003.

[92] S. Harris, N. Shadbolt. SPARQL query processing with conventional relational database systems. InSSWS Conference, 2005.

[93] M. Al Hasan, V. Chaoji, S. Salem, J. Besson, M. J. Zaki. ORIGAMI: Mining Representative Orthogonal Graph Patterns.ICDM Conference, 2007.

[94] D. Haussler. Convolution kernels on discrete structures.Technical Report UCSC-CRL-99-10, University of California, Santa Cruz, 1999.

[95] T. Haveliwala. Topic-Sensitive Page Rank,World Wide Web Conference, 2002.

[96] H. He, A. K. Singh. Query Language and Access Methods for Graph Databases, appears as a chapter in Managing and Mining Graph Data, ed. Charu Aggarwal, Springer, 2010.

[97] H. He, Querying and mining graph databases.Ph.D. Thesis, UCSB, 2007.

[98] H. He, A. K. Singh. Efficient Algorithms for Mining Significant Substructures from Graphs with Quality Guarantees.ICDM Conference, 2007.

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

[100] J. Huan, W. Wang, J. Prins, J. Yang. Spin: Mining Maximal Frequent Subgraphs from Graph Databases.KDD Conference, 2004.

[101] J. Huan, W. Wang, D. Bandyopadhyay, J. Snoeyink, J. Prins, A. Tropsha. Mining Spatial Motifs from Protein Structure Graphs. Research in Computational Molecular Biology (RECOMB), pp. 308–315, 2004.

[102] V. Hristidis, N. Koudas, Y. Papakonstantinou, D. Srivastava. Keyword proximity search in XML trees.IEEE Transactions on Knowledge and Data Engineering, 18(4):525–539, 2006.

[103] V. Hristidis, Y. Papakonstantinou. Discover: Keyword search in relational databases. VLDB Conference, 2002.

[104] A. Inokuchi, T. Washio, H. Motoda. An Apriori-based Algorithm for Mining Frequent Substructures from Graph Data.PKDD Conference, pages 13–23, 2000.

[105] H. V. Jagadish. A compression technique to materialize transitive closure.ACM Trans. Database Syst., 15(4):558–598, 1990.

[106] H. V. Jagadish, S. Al-Khalifa, A. Chapman, L. V. S. Lakshmanan, A. Nierman, S. Paparizos, J. M. Patel, D. Srivastava, N. Wiwatwattana, Y. Wu, C. Yu. TIMBER: A native XML database. InVLDB Journal, 11(4):274–291, 2002.

[107] H. V. Jagadish, L. V. S. Lakshmanan, D. Srivastava, K. Thompson. TAX: A tree algebra for XML.DBPL Conference, 2001.

[108] G. Jeh, J. Widom. Scaling personalized web search. InWWW, pages 271–279, 2003.

[109] J. L. Jenkins, A. Bender, J. W. Davies. In silico target fishing: Predicting biological targets from chemical structure.Drug Discovery Today, 3(4):413–421, 2006.

[110] R. Jin, C. Wang, D. Polshakov, S. Parthasarathy, G. Agrawal. Discovering Frequent Topological Structures from Graph Datasets.ACM KDD Conference, 2005.

[111] R. Jin, H. Hong, H. Wang, Y. Xiang, N. Ruan. Computing LabelConstraint Reachability in Graph Databases.Under submission, 2009.

[112] R. Jin, Y. Xiang, N. Ruan, D. Fuhry. 3-HOP: A high-compression indexing scheme for reachability query.SIGMOD Conference, 2009.

[113] V. Kacholia, S. Pandit, S. Chakrabarti, S. Sudarshan, R. Desai, H. Karambelkar. Bidirectional expansion for keyword search on graph databases.VLDB Conference, 2005.

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

[115] R. Kaushik, P. Bohannon, J. Naughton, H. Korth. Covering indexes for branching path queries. InSIGMOD Conference, June 2002.

[116] B.W. Kernighan, S. Lin. An efficient heuristic procedure for partitioning graphs,Bell System Tech. Journal, vol. 49, Feb. 1970, pp. 291-307.

[117] M.-S. Kim, J. Han. A Particle-and-Density Based Evolutionary Clustering Method for Dynamic Networks,VLDB Conference, 2009.

[118] J. M. Kleinberg. Authoritative Sources in a Hyperlinked Environment. Journal of the ACM, 46(5):pp. 604–632, 1999.

[119] R.I. Kondor, J. Lafferty. Diffusion kernels on graphs and other discrete input spaces. ICML Conference, pp. 315–322, 2002.

[120] M. Koyuturk, A. Grama, W. Szpankowski. An Efficient Algorithm for Detecting Frequent Subgraphs in Biological Networks.Bioinformatics, 20:I200–207, 2004.

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

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

[123] M. Kuramochi, G. Karypis. Frequent subgraph discovery. ICDM Conference, pp. 313–320, Nov. 2001.

[124] M. Kuramochi, G. Karypis. Finding frequent patterns in a large sparse graph.Data Mining and Knowledge Discovery, 11(3): pp. 243–271, 2005.

[125] J. Larrosa, G. Valiente. Constraint satisfaction algorithms for graph pattern matching. Mathematical Structures in Computer Science, 12(4): pp. 403–422, 2002.

[126] M. Lee, W. Hsu, L. Yang, X. Yang. XClust: Clustering XML Schemas for Effective Integration.CIKM Conference, 2002.

[127] J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. VanBriesen, N. S. Glance. Cost-effective outbreak detection in networks.KDD Conference, pp. 420–429, 2007.

[128] J. Leskovec, M. McGlohon, C. Faloutsos, N. Glance, M. Hurst. Cascading Behavior in Large Blog Graphs,SDM Conference, 2007.

[129] J. Leskovec, J. Kleinberg, C. Faloutsos. Graphs over time: Densification laws, shrinking diameters and possible explanations. ACM KDD Conference, 2005.

[130] J. Leskovec, E. Horvitz. Planetary-Scale Views on a Large InstantMessaging Network,WWW Conference, 2008.

[131] J. Leskovec, L. Backstrom, R. Kumar, A. Tomkins. Microscopic Evolution of Social Networks,ACM KDD Conference, 2008.

[132] Q. Li, B. Moon. Indexing and querying XML data for regular path expressions. InVLDB Conference, pages 361 370, September 2001.

[133] W. Lian, D.W. Cheung, N. Mamoulis, S. Yiu. An Efficient and Scalable Algorithm for Clustering XML Documents by Structure,IEEE Transactions on Knowledge and Data Engineering, Vol 16, No. 1, 2004.

[134] L. Lim, H. Wang, M. Wang. Semantic Queries in Databases: Problems and Challenges.CIKM Conference, 2009.

[135] Y.-R. Lin, Y. Chi, S. Zhu, H. Sundaram, B. L. Tseng. FacetNet: A framework for analyzing communities and their evolutions in dynamic networks. WWW Conference, 2008.

[136] C. Liu, X. Yan, H. Yu, J. Han, P. S. Yu. Mining Behavior Graphs for "Backtrace" of Noncrashing Bugs.SDM Conference, 2005.

[137] C. Liu, X. Yan, L. Fei, J. Han, S. P. Midkiff. SOBER: Statistical Model-Based Bug Localization.SIGSOFT Software Engineering Notes, 30(5):286–295, 2005

[138] Q. Lu, L. Getoor. Link-based classification. ICML Conference, pages 496–503, 2003.

[139] F. Manola, E. Miller. RDF Primer. W3C, http://www.w3.org/TR/rdfprimer/, 2004.

[140] A. McGregor. Finding Graph Matchings in Data Streams. APPROXRANDOM, pp. 170–181, 2005.

[141] T. Milo and D. Suciu. Index structures for path expression. In ICDT Conference, pages 277–295, 1999.

[142] S. Navlakha, R. Rastogi, N. Shrivastava. Graph Summarization with Bounded Error.ACMSIGMOD Conference, pp. 419–432, 2008.

[143] M. Neuhaus, H. Bunke. Self-organizing maps for learning the edit costs in graph matching. IEEE Transactions on Systems, Man, and Cybernetics, 35(3) pp. 503–514, 2005.

[144] M. Neuhaus, H. Bunke. Automatic learning of cost functions for graph edit distance.Information Sciences, 177(1), pp 239–247, 2007.

[145] M. Neuhaus, H. Bunke. Bridging the Gap Between Graph Edit Distance and Kernel Machines.World Scientific, 2007.

[146] M. Newman. Finding community structure in networks using the eigenvectors of matrices.Physical Review E, 2006.

[147] M. E. J. Newman. The spread of epidemic disease on networks, Phys. Rev. E 66, 016128, 2002.

[148] J. Pei, D. Jiang, A. Zhang. On Mining Cross-Graph Quasi-Cliques,ACM KDD Conference, 2005.

[149] Nidhi, M. Glick, J. Davies, J. Jenkins. Prediction of biological targets for compounds using multiple-category bayesian models trained on chemogenomics databases.J Chem Inf Model, 46:1124–1133, 2006.

[150] S. Nijssen, J. Kok. A quickstart in frequent structure mining can make a difference.Proceedings of SIGKDD, pages 647–652, 2004.

[151] L. Page, S. Brin, R. Motwani, T. Winograd. The PageRank Citation Ranking: Bringing Order to the Web.Technical report, Stanford Digital Library Technologies Project, 1998.

[152] Z. Pan, J. Heflin. DLDB: Extending relational databases to support Semantic Web queries. InPSSS Conference, 2003.

[153] J. Pei, D. Jiang, A. Zhang. Mining Cross-Graph Quasi-Cliques in Gene Expression and Protein Interaction Data,ICDE Conference, 2005.

[154] E. Prud'hommeaux and A. Seaborne. SPARQL query language for RDF. W3C,URL:http://www.w3.org/TR/rdf-sparql-query/, 2007.

[155] L. Qin, J.-X. Yu, L. Chang. Keyword search in databases: The power of RDBMS.SIGMOD Conference, 2009.

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

[157] S. Ranu, A. K. Singh. GraphSig: A scalable approach to mining significant subgraphs in large graph databases.ICDE Conference, 2009.

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

[159] P. R. Raw, B. Moon. PRIX: Indexing and querying XML using pr- ufer sequences. ICDE Conference, 2004.

[160] J. W. Raymond, P. Willett. Maximum common subgraph isomorphism algorithms for the matching of chemical structures. J. Comp. Aided Mol. Des., 16(7):521–533, 2002.

[161] K. Riesen, X. Jiang, H. Bunke. Exact and Inexact Graph Matching: Methodology and Applications, appears as a chapter inManaging and Mining Graph Data, ed. Charu Aggarwal, Springer, 2010.

[162] H. Saigo, S. Nowozin, T. Kadowaki, T. Kudo, and K. Tsuda. GBoost: A mathematical programming approach to graph classification and regression.Machine Learning, 2008.

[163] F. Sams-Dodd. Target-based drug discovery: is something wrong?Drug Discov Today, 10(2):139–147, Jan 2005.

[164] P. Sarkar, A. Moore, A. Prakash. Fast Incremental Proximity Search in Large Graphs,ICML Conference, 2008.

[165] P. Sarkar, A. Moore. Fast Dynamic Re-ranking of Large Graphs,WWW Conference, 2009.

[166] A. D. Sarma, S. Gollapudi, R. Panigrahy. Estimating PageRank in Graph Streams,ACM PODS Conference, 2008.

[167] V. Satuluri, S. Parthasarathy. Scalable Graph Clustering Using Stochastic Flows: Applications to Community Discovery,ACM KDD Conference, 2009.

[168] R. Schenkel, A. Theobald, G. Weikum. Hopi: An efficient connection index for complex XML document collections. EDBT Conference, 2004.

[169] J. Shanmugasundaram, K. Tufte, C. Zhang, G. He, D. J. DeWitt, J. F. Naughton. Relational databases for querying XML documents: Limitations and opportunities. VLDB Conference, 1999.

[170] N. Stiefl, I. A. Watson, K. Baumann, A. Zaliani. Erg: 2d pharmacophore descriptor for scaffold hopping.J. Chem. Info. Model., 46:208–220, 2006.

[171] J. Sun, S. Papadimitriou, C. Faloutsos, P. Yu. GraphScope: Parameter Free Mining of Large Time-Evolving Graphs,ACM KDD Conference, 2007.

[172] S. J. Swamidass, J. Chen, J. Bruand, P. Phung, L. Ralaivola, P. Baldi. Kernels for small molecules and the prediction of mutagenicity, toxicity and anti-cancer activity.Bioinformatics, 21(1):359–368, 2005.

[173] L. Tang, H. Liu, J. Zhang, Z. Nazeri. Community evolution in dynamic multi-mode networks.ACM KDD Conference, 2008.

[174] B. Taskar, P. Abbeel, D. Koller. Discriminative probabilistic models for relational data. InUAI, pages 485–492, 2002.

[175] H. Tong, C. Faloutsos, J.-Y. Pan. Fast random walk with restart and its applications. InICDM, pages 613–622, 2006.

[176] S. TrißI, U. Leser. Fast and practical indexing and querying of very large graphs.SIGMOD Conference, 2007.

[177] A. A. Tsay, W. S. Lovejoy, D. R. Karger. Random Sampling in Cut, Flow, and Network Design Problems,Mathematics of Operations Research, 24(2):383-413, 1999.

[178] K. Tsuda, W. S. Noble. Learning kernels from biological networks by maximizing entropy.Bioinformatics, 20(Suppl. 1):i326–i333, 2004.

[179] K. Tsuda, H. Saigo. Graph Classification, appears as a chapter inManaging and Mining Graph Data, Springer, 2010.

[180] J.R. Ullmann. An Algorithm for Subgraph Isomorphism.Journal of the Association for Computing Machinery, 23(1): pp. 31–42, 1976.

[181] N. Vanetik, E. Gudes, S. E. Shimony. Computing Frequent Graph Patterns from Semi-structured Data.IEEE ICDM Conference, 2002.

[182] R. Volz, D. Oberle, S. Staab, and B. Motik. KAON SERVER : A Semantic Web Management System. InWWW Conference, 2003.

[183] H. Wang, C. Aggarwal. A Survey of Algorithms for Keyword Search on Graph Data. appears as a chapter in Managing and Mining Graph Data, Springer, 2010.

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

[185] H. Wang, S. Park, W. Fan, P. S. Yu. ViST: A Dynamic Index Method for Querying XML Data by Tree Structures.In SIGMOD Conference, 2003.

[186] H. Wang, X. Meng. On the Sequencing of Tree Structures for XML Indexing.In ICDE Conference, 2005.

[187] Y. Wang, D. Chakrabarti, C. Wang, C. Faloutsos. Epidemic Spreading in Real Networks: An Eigenvalue Viewpoint, SRDS, pp. 25-34, 2003.

[188] N. Wale, G. Karypis. Target identification for chemical compounds using target-ligand activity data and ranking based methods. Technical Report TR-08-035, University of Minnesota, 2008.

[189] N. Wale, G. Karypis, I. A. Watson. Method for effective virtual screening and scaffold-hopping in chemical compounds. Comput Syst Bioinformatics Conf, 6:403–414, 2007.

[190] N. Wale, X. Ning, G. Karypis. Trends in Chemical Graph Data Mining, appears as a chapter inManaging and Mining Graph Data, Springer, 2010.

[191] N. Wale, I. A. Watson, G. Karypis. Indirect similarity based methods for effective scaffold-hopping in chemical compounds.J. Chem. Info. Model., 48(4):730–741, 2008.

[192] N. Wale, I. A. Watson, G. Karypis. Comparison of descriptor spaces for chemical compound retrieval and classification. Knowledge and Information Systems, 14:347–375, 2008.

[193] C. Weiss, P. Karras, A. Bernstein. Hexastore: Sextuple Indexing for Semantic Web Data Management. InVLDB Conference, 2008.

[194] K. Wilkinson. Jena property table implementation. InSSWS Conference, 2006.

[195] K. Wilkinson, C. Sayers, H. A. Kuno, and D. Reynolds. Efficient RDF storage and retrieval in Jena2. In SWDB Conference, 2003.

[196] Y. Xu, Y. Papakonstantinou. Efficient LCA based keyword search in XML data.EDBT Conference, 2008.

[197] Y. Xu, Y.Papakonstantinou. Efficient keyword search for smallest LCAs in XML databases. ACM SIGMOD Conference, 2005.

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

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

[200] X. Yan, J. Han. Gspan: Graph-based Substructure Pattern Mining. ICDM Conference, 2002.

[201] X. Yan, P. S. Yu, J. Han. Graph indexing: A frequent structure-based approach.SIGMOD Conference, 2004.

[202] X. Yan, P. S. Yu, J. Han. Substructure similarity search in graph databases.SIGMOD Conference, 2005.

[203] X. Yan, B. He, F. Zhu, J. Han. Top-K Aggregation Queries Over Large Networks,IEEE ICDE Conference, 2010.

[204] J. X. Yu, J. Cheng. Graph Reachability Queries: A Survey, appears as a chapter inManaging and Mining Graph Data, Springer, 2010.

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

[206] T. Zhang, A. Popescul, B. Dom. Linear prediction models with graph regularization for web-page categorization. ACM KDD Conference, pages 821–826, 2006.

[207] Q. Zhang, I. Muegge. Scaffold hopping through virtual screening using 2d and 3d similarity descriptors: Ranking, voting and consensus scoring. J. Chem. Info. Model., 49:1536–1548, 2006.

[208] P. Zhao, J. Yu, P. Yu. Graph indexing: tree + delta >=graph. VLDB Conference, 2007.

[209] D. Zhou, J. Huang, B. Sch- olkopf. Learning from labeled and unlabeled data on a directed graph.ICML Conference, pages 1036–1043, 2005.

[210] D. Zhou, O. Bousquet, J. Weston, B. Sch- olkopf. Learning with local and global consistency. Advances in Neural Information Processing Systems (NIPS) 16, pages 321–328. MIT Press, 2004. [211] X. Zhu, Z. Ghahramani, J. Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. ICML Conference, pages 912–919, 2003.