تجزیه و تحلیل شبکه و گراف
با بکارگیری نرم افـــزار پایتون

مولف: محمـــد زهیر الطـــای، سیـف الدین کـــدری
مترجم: دکتر مهدی غضنفری، مهندس زهرا السادات طبایی

تجزیه و تحلیل شبکه و گراف
با بکارگیری نرم افـــزار پایتون
مولف: محمـــد زهیر الطـــای، سیـف الدین کـــدری
مترجم: دکتر مهدی غضنفری، مهندس زهرا السادات طبایی

barcode

بسم الله الرحمن الرحيم

تجزیه و تحلیل شبکه‌های اجتماعی

با بکارگیری پایتون

ترجمه و تالیف:

دکتر مهدی غضنفری،

مهندس زهرا سادات طبایی

سرشناسه : غضنفری، مهدی، ۱۳۳۹-
عنوان وپدیدآور : تحلیل شبکه‌های اجتماعی (با بکارگیری پایتون)/ مهدی غضنفری، زهرا سادات طبایی
مشخصات نشر : تهران ، ۱۳۹۸
مشخصات ظاهری : ۴۰۸ ص.: مصور، جدول، نمودار.
شابک :
فهرست نویسی :
موضوع : تجزیه و تحلیل شبکه و گراف
شناسه افزوده : طبایی، زهرا سادات
رده بندی کنگره :
رده بندی دیویی :
شماره کتابخانه ملی :

- فروشگاه :
- پست الکترونیک:

نام کتاب: تحلیل شبکه‌های اجتماعی با بکارگیری پایتون
ترجمه و تالیف : دکتر مهدی غضنفری، مهندس زهرا سادات طبایی
چاپ اول: ۱۳۹۸
شمارگان: ۱۰۰۰ جلد
قیمت:
لیتوگرافی، چاپ و صحافی:

شابک: *ISBN:*

بسم الله الرحمن الرحیم

پیشگفتار

ن وَالْقَلَمِ وَمَا یَسْطُرُونَ ...

تحولات سریع در وب و اینترنت در دهه گذشته و پیشرفت در محاسبات و ارتباطات، بشر را با شیوه‌های جدید و پیچیده‌ای مواجه کرده است. سایتهای اجتماعی جدیدی بُروز و ظهور یافته و با مشارکتی وسیع و استقبالی گسترده مواجه شده که هر یک اشکال متفاوتی از ارتباطات، تعاملات و همکاری را موجب شده‌اند. سایتهایی مانند توییتر، فیسبوک، لینکدین و مای‌اسپیس، به افراد اجازه می‌دهند ارتباطات مجازی جدیدی ایجاد کنند. ویکیها، وبلاگها و وبلاگهای ویدئویی به راحتی به کاربران امکان می‌دهند تا ایده‌ها و افکار خود را منتشر کنند، بدون اینکه نگران هزینه‌های انتشار آن باشند.

تعداد زیادی از علاقمندان امروزه می‌توانند مقالات خود را چاپ کرده و عکسها، فیلمها و لینکها را در محدوده و مقیاسی که هرگز قبلاً تصور نمی‌کردند، منتشر کنند. توصیه‌گرهای محصولات، از طریق بازارهای آنلاین مانند *eBay* و آمازون (پس از تجزیه و تحلیل رفتار کاربر) می‌تواند مصرف‌کنندگان آنلاین را ترغیب به سفارشات بیشتر کنند. مکانیزمهای برچسب زدن، به کاربران وب کمک می‌کنند تا اولویتهای خود را بیان کنند. ارسال و دریافت ایمیل، بازدید از یک صفحه وب و یا ارسال نظر در یک سایت وبلاگ، اغلب یک ردپای دیجیتالی را از خودش بجا می‌گذارد که می‌تواند توسط فرد یا گروهی که پشت آن قرار گرفته‌اند ردیابی شود. گروهها و احزاب سیاسی می‌توانند از وب سایت برای ایجاد تالارهای گفتگو و یا اشکال دیگر همکاری بین طرفداران خود استفاده کرده، ایده‌ها و تفکرات تازه‌ای را مطرح کرده و حتی جنبشهای جدیدی را براه بیندازند.

تمام این تغییرات بدون کمک فناوری *Web 2.0* هرگز ممکن نبود – اصطلاحی که توسط تیم اُ ریلی[1] بکار گرفته شد تا نشان دهد که کاربران اینترنت، بیشتر از قبل آمادگی اصلاح محتوای وب را دارند.

شبکه‌های اجتماعی یکی از عوامل مهم در ظهور چنین تعاملاتی هستند، زیرا اکثر کاربران اینترنت، بازیگران سایتهای اجتماعی بوده و بسیار منظم و فعال از سایتها استفاده می‌کنند. مطالعات اخیر نشان داده است که "شبکه‌های اجتماعی" به عنوان یکی از سه مورد استفاده رایج از اینترنت، در کنار

[1] Tim O'Reilly

جستجوی اینترنتی و ایمیل می‌باشد، که به اهمیت این روند اجتماعی و نقش آن در اجتماعات اشاره می‌کند.

در سالهای اخیر، موضوع تجزیه و تحلیل شبکه‌های اجتماعی، بعنوان زیرمجموعه مبحث شبکه‌های اجتماعی، به یک حوزه پژوهشی جذاب و بین‌رشته‌ای بدل شده است. مثالی از حوزه بین‌رشته‌ای، حوزه میان علوم کامپیوتر و جامعه‌شناسی است. در واقع دانشمندان رشته‌های کامپیوتر و جامعه‌شناسی تنها با ارتقا توان و دانش خود به سطحی بالاتر از گذشته، قادر به مواجهه با چالشهای این رشتهٔ در حال توسعه هستند. از یک سو، متخصصان رشته کامپیوتر، دانش تجزیه و پردازش داده‌ها را دارند، و از سوی دیگر جامعه‌شناسان، تجربهٔ ویرایش موثر و تفسیر درست داده‌های مفید را دارند. لذا تحلیل شبکه‌های اجتماعی در بسیاری از کاربردهای خود، محتاج دانش و تجربه توام این دو گروه است.

تکنیکهای تجزیه و تحلیل شبکه‌های اجتماعی که در این کتاب گنجانده شده است، به خوانندگان کمک می‌کند تا به طور موثر، داده‌های اجتماعی توییتر، فیسبوک، لایو ژورنال[1]، گیت‌هاب[2] و بسیاری دیگر را در سه سطح: فردی، بخشی و جمعی تحلیل کند. آنها قادر به تجزیه و تحلیل شبکه‌های سیاسی، شبه‌نظامی و انقلابی و یا شبکه‌های اقتصادی و فرهنگی خواهند بود. آنها حتی یاد خواهند گرفت چگونه ویروس ابولا[3] از طریق جوامع گسترش می‌یابد.

تجزیه و تحلیل شبکه‌های اجتماعی به طور موفقیت‌آمیزی در زمینه‌های مختلف مانند سلامت، امنیت سایبری، کسب و کار، زندگی حیوانات، بازیابی اطلاعات و ارتباطات مورد استفاده قرار گرفته است. به عنوان مثال، در حیات وحش، تجزیه و تحلیل شبکه‌های اجتماعی برای بررسی روابط و ساختار اجتماعی تجمعات حیوانات و تعاملات مستقیم و غیر مستقیم بین گروههای حیوانی بکار گرفته شده است. در مورد امنیت جوامع انسانی، تحلیل شبکه‌های اجتماعی توسط سازمانهای امنیتی بکار برده شده و به مطالعه ساختار و پویایی گروههای مختلف اجتماعی مثلا گروههای قومیتی و یا ستیزه‌جو پرداخته است.

[1] LiveJournal
[2] GitHub
[3] Ebola Virus

آموزش تحلیل شبکه‌ها به زبانی ساده

این کتاب در مورد تحلیل گراف و شبکه است و این کار را با زبانی ساده انجام داده است. کتاب حاضر عملاً دانش تئوری و کاربرد تحلیل شبکه را یکپارچه کرده است. با استفاده از روشی گام به گام، ابتدا مفاهیم ساختاری اصلی و سپس کاربرد آنها در تحقیقات اجتماعی را معرفی می‌کند. هدف این کتاب، حل مسائل گرافها و شبکه‌های اجتماعی با کاوش و بررسی دهها مثال در این حوزه، از سطح آسان، متوسط و تا سخت است. چنین روشی باعث ارائه یک دانش کاربردی در این زمینه می‌شود.

بجز فصل اول که دربرگیرنده مفاهیم نظری است، در هریک از فصلهای دوم تا هشتم، هر بحث نظری با مثالهایی از نحوه تجزیه و تحلیل گراف و شبکه از طریق زبان پایتون دنبال می‌شود. پایتون، یک زبان برنامه‌نویسی عمومی با محبوبیت روزافزون در حوزه علوم داده است. فصل نهم یک مورد کاربردی جذاب در دنیای سیاست را مورد توجه قرار داده است.

همانطور که ذکر شد آموزشهای نظری این کتاب با دستورات پایتون همراه شده است. بکارگیری پایتون به این دلیل است که شرکتهای بزرگ در سراسر جهان، برای کسب بینش در مورد داده‌های خود و رسیدن به مزایای رقابتی، از "پایتون" استفاده می‌کنند. این کتاب همچنین از کتابخانه *NetworkX* استفاده می‌کند که یک بسته نرم افزاری زبان پایتون و یک ابزار منبع باز برای ایجاد، دستکاری و مطالعه ساختار، پویایی و عملکرد شبکه‌های پیچیده است. در کنار این دو، از بسته کتابخانه *Matplotlib* برای تجسم داده‌ها بهره می‌گیرد. در واقع این کتاب از سه ابزار منبع باز، برای تجزیه و تحلیل و تجسم داده‌های اجتماعی استفاده می‌کند. خوانندهٔ این کتاب در نهایت، دانش، مهارت و ابزار لازم برای تجزیه و تحلیل شبکه‌های اجتماعی در تمام زمینه‌های قابل دسترس، اعم از رسانه اجتماعی تا مدیریت کسب و کار و تاریخ را داراست.

این کتاب بویژه برای خوانندگانی مفید است که می خواهند نظریه و کارکرد تحلیل گراف و شبکه را با استفاده از یک زبان برنامه‌نویسی، یعنی پایتون، بدون عمیق شدن در روشهای ریاضی یا آماری آنها، بکار گیرند. در واقع، این کتاب متناسب با نیاز آموزشی دوره‌های تجزیه و تحلیل شبکه‌های اجتماعی در تمام رشته‌هایی است که از روش‌شناسی اجتماعی استفاده می‌کنند. به نظر می‌رسد بسیاری از خوانندگان، علاقه بیشتری به کاربردهای تحلیل شبکه اجتماعی داشته و علاقه کمتری به بررسی ویژگیهای ریاضی آن دارند.

این کتاب شامل نه فصل است:

فصل ۱- مفاهیم نظری تجزیه و تحلیل شبکه: در این فصل، به‌عنوان طولانی‌ترین فصل، مقدمه‌ای بر مفاهیم مهم نظری تحلیل شبکه ارائه شده است؛ البته تأکید فصل اول بر آن دسته از مفاهیمی است که در فصول بعد این کتاب، استفاده می‌شود.

فصل ۲- مبانی شبکه: این فصل "مفهوم شبکه" را معرفی می‌کند؛ موضوعی که عملاً هسته اصلی مبحث "تحلیل شبکه" است. ما در این فصل موضوعاتی همانند انواع شبکه‌ها، سنجه‌ها یا معیارهای شبکه، نصب و استفاده از کتابخانه *NetworkX*، نمایش داده‌های شبکه، عملیات پایه‌ای ماتریسی و تجسم بخشی یا تصویرسازی داده‌ها را مورد بحث قرار خواهیم داد.

فصل ۳- نظریه گراف: در این فصل ویژگیهای اصلی "نظریه گراف"، مبانی ریاضی کاربردها و خواص آنها را بحث می‌کنیم. موضوع گرافها در ابتدا با مطالعه بازیهای شانسی شروع شد و سپس گسترش فراوانی یافت. بنابراین موضوعاتی همانند ریشه‌های نظریه گراف، مبانی گراف، گونه‌های مختلف گراف، پیمایشهای گراف و انواع عملیات در گراف را بررسی می‌کنیم.

فصل ٤- شبکه‌های اجتماعی: این فصل، مفاهیم اصلی و برخی دیگر از موضوعات مرتبط با شبکه‌های اجتماعی مانند خواص شبکه‌های اجتماعی، جمع‌آوری داده‌ها در شبکه‌های اجتماعی، نمونه‌گیری داده‌ها و تجزیه و تحلیل شبکه‌های اجتماعی را معرفی می‌کند.

فصل ٥- تجزیه و تحلیل در سطح گره: این فصل به بسط دانش خواننده در مورد چگونگی تجزیه و تحلیل شبکه در سطح گره یا کنشگر یعنی "تحلیل فردانه" می‌پردازد. منظور از این دانش آنست که دریابیم چگونه یک شبکه اجتماعی را از ابتدا ایجاد کنیم، چگونه آن شبکه را وارد (نرم‌افزار) کنیم، چگونه بازیگران یا کنشگران کلیدی آنرا با استفاده از معیارهای مرکزیت پیدا کنیم و بالاخره اینکه چگونه آن شبکه را مشاهده‌پذیر و متجسم یا مصور کنیم. همچنین در ادامه فصل الگوریتمهای مهمی را معرفی خواهیم کرد که با استفاده از آنها می‌توان شناخت و بینش مناسبی از درون گرافها کسب کرد.

فصل ٦- تحلیل در سطح گروه: در این فصل، قصد داریم برخی مباحث و تکنیکهای شناسایی "گروههای منسجم" در شبکه‌ها (اصطلاحا تحلیل بخشانه)، مانند محافل، ضریب خوشه‌گی، تحلیل سه‌وَندی، حفره‌های ساختاری، واسطه‌گری (یا کارگزاری)، تراگذری، خوشه‌بندی سلسله‌مراتبی و مدلهای بلوک را ارائه دهیم. همه این مفاهیم بر اساس نحوه اتصال گره‌ها در شبکه هستند؛ یعنی وضعیتِ تفرق یا تجمع گروهی از گره‌ها در شبکه (و کثرت یا قلت پیوند میان آنان) موجب پدیدار شدن این مفاهیم

شده است. در میان همه موضوعات فوق، دو مبحث: تحلیل انسجام و تعیین واسطه‌گری، همچنان دو عنوان مهم پژوهشی در حوزه تحلیل شبکه اجتماعی است.

فصل ۷- تحلیل در سطح شبکه: در این فصل قصد داریم گرافها و شبکه‌ها را بعنوان یک کل و در سطح کلان مطالعه و بررسی کنیم (تحلیل جمعانه). این تحلیل با آنچه در فصلهای پیشین انجام دادیم، یعنی گرافها را در سطح گره بررسی کردیم (تحلیل فردانه) و یا در سطح گروه مطالعه کردیم (تحلیل بخشانه)، متفاوت است. لذا، در این فصل مفاهیمی مانند مولفه‌ها (سازندها) و گره‌های مجزا (ایزوله‌ها)، هسته‌ها و حاشیه (پیرامون)، چگالی (تراکم) شبکه، کوتاهترین مسیرها، دوسویگی (ارتباط متقابل)، شبکه‌های وابستگی سازمانی (وابستاری) و شبکه‌های دو-حالته (دو-بخشی) و همریختی (همگونی) مورد توجه قرار می‌گیرند.

فصل ۸- انتشار اطلاعات در شبکه‌های اجتماعی: در این فصل، درمورد دو نوع کلی انتشار اطلاعات در شبکه‌های اجتماعی، مفصلاً بحث خواهیم کرد: انتشار نوآوری و همه‌گیری (اپیدمی) . انتشار نوآوری در بسیاری از زمینه‌ها مورد مطالعه قرار گرفته است، لذا در این فصل، ما تنها از دیدگاه شبکه اجتماعی آن را در نظر می‌گیریم. سایر انواع انتشار اطلاعات که در این فصل گنجانده نشده‌اند عبارتند از: رفتار گروهی (یا اصطلاحاً رفتار گله‌ای یا توده‌ای) و آبشار اطلاعات.

فصل ۹- کاربرد تجزیه و تحلیل شبکه در نگاشت گفتگوهای سیاسی برخط: در این فصل، به مدد روشهای تجزیه و تحلیل شبکه که در فصول پیش بحث شد، قصد داریم جزئیات رفتار کاربران تالار گفتگوی جنبش پنج ستاره ایتالیا، را در دو سطح که عبارتند از سطح گره، یعنی تحلیل فردانه (شامل کاربران و پیامها) و سطح شبکه، یعنی تحلیل جمعانه (شامل سازندها و اجتماعات) بررسی کرده تا ویژگیهای شبکه مذکور و رفتار کنشگران آن را توضیح دهیم. این فصل، یک مطالعه موردی جذاب است که چگونگی استفاده از تکنیکهای تحلیل شبکه‌های اجتماعی را در معرض دید خواننده می‌گذارد.

مطالعه این کتاب برای داشجویان مقاطع کارشناسی ارشد و دکتری در رشته‌های کامپیوتر، فناوری اطلاعات، مهندسی سیستم، تجارت الکترونیک، سلامت و امثال آن مفید است. همچنین همه آنهاییکه در حوزه کاربردهای شبکه‌های اجتماعی در حوزه‌های گوناگون تحقیق می‌کنند، می‌توانند از مطالب این کتاب بهره ببرند.

علیرغم همه تلاشی که برای فقدان خطا در فهم و تایپ متن داشتیم اما همچنان نگرانیم که مبادا اشتباهی در این کار رخ داده باشد. لذا، لطفا وجود هر گونه خطا در تنظیم و تایپ کتاب را بر ما بخشیده و آنرا به نشانی رایانامه mehdi@iust.ac.ir اطلاع دهید تا در ویرایشهای بعدی، مد نظر قرار گیرد.

مهدی عضنفری
استاد دانشکده مهندسی صنایع
دانشگاه علم و صنعت ایران

زهرا سادات طبایی
کارشناس ارشد مهندسی صنایع
دانشگاه علم و صنعت ایران

فهرست مطالب

فصل اول: مفاهیم نظری تحلیل شبکه‌های اجتماعی ۱

۱-۱ معنای جامعه‌شناختی روابط شبکه‌ای..... ۲

۱-۲ سنجه‌های شبکه..... ۶

۱-۲-۱ همبندی شبکه..... ۶

۱-۲-۲ تراگذری..... ۷

۱-۲-۳ چندگانگی..... ۸

۱-۲-۴ همریختی..... ۱۱

۱-۲-۵ دووَندها و تقابل..... ۱۲

۱-۲-۶ سه‌وَندها و تعادل..... ۱۲

۱-۲-۷ دوسویگی..... ۱۵

۱-۳ توزیع شبکه..... ۱۶

۱-۳-۱ فاصله میان دو گره..... ۱۶

۱-۳-۲ مرکزیت درجه‌ای..... ۱۷

۱-۳-۳ مرکزیت نزدیکی..... ۱۹

۱-۳-۴ مرکزیت میانی یا بینابینی..... ۲۰

۱-۳-۵ مرکزیت بردار ویژه..... ۲۳

۱-۳-۶ رتبه-صفحه..... ۲۴

۱-۳-۷ فاصله کروی (ژئودزیک) و کوتاهترین مسیر..... ۲۶

۱-۳-۸ گریز از مرکز..... ۲۶

۱-۳-۹ چگالی..... ۲۸

۱-۴ بخش‌بندی شبکه..... ۳۰

۱-۴-۱ گروههای منسجم..... ۳۰

۱-۴-۲ محفل..... ۳۱

۱-۴-۳ k-هسته..... ۳۲

۱-۴-۴ ضریب خوشه‌گی..... ۳۳

۱-۴-۵ هسته/حاشیه..... ۳۵

۱-۴-۶ مدلهای بلوک..... ۳۷

۱-۴-۷ خوشه‌بندی سلسله مراتبی ۳۸
۱-۵ تحولات اخیر در حوزه تحلیل شبکه ۴۰
۱-۵-۱ انجمن‌یابی ۴۰
۱-۵-۲ پیش‌بینی پیوند ۴۳
۱-۵-۳ شبکه‌های فضایی ۴۵
۱-۵-۴ شبکه‌های تعامل پروتئین به پروتئین ۴۶
۱-۵-۵ سیستمهای توصیه‌گر ۴۸
۱-۶ بسته *iGraph* ۵۰

فصل دوم: مبانی شبکه ۵۵
۲-۱ شبکه چیست؟ ۵۶
۲-۲ انواع شبکه‌ها ۵۶
۲-۳ ویژگیهای شبکه‌ها ۵۹
۲-۴ سنجه‌های شبکه ۶۱
۲-۵ کتابخانه *NetworkX* ۶۳
۲-۶ نصب ۶۵
۲-۷ ماتریسها ۶۹
۲-۸ انواع ماتریس در شبکه‌های اجتماعی ۷۱
۲-۸-۱ ماتریس مجاورت ۷۱
۲-۸-۲ ماتریس فهرست یال ۷۳
۲-۸-۳ فهرست مجاورت ۷۵
۲-۸-۴ ماتریس *Numpy* ۷۸
۲-۸-۵ ماتریس تُنُک ۷۹
۲-۹ عملیات پایه‌ای ماتریسها ۷۹
۲-۱۰ تجسم داده‌ها ۸۱

فصل سوم: تئوری گراف ۸۳
۳-۱ ریشه‌های نظریه گراف ۸۴

۳-۲ مبانی گراف ۸۷
۳-۳ راسها ۸۹
۳-۴ انواع گراف ۹۱
۳-۵ پیمایشهای گراف ۹۶
۳-۵-۱ پیمایش اول-عمق ۹۷
۳-۵-۲ پیمایش اول-سطح ۱۰۰
۳-۵-۳ الگوریتم دایکسترا ۱۰۳
۳-۶ عملیات بر روی گرافها ۱۰۵

فصل چهارم: شبکه‌های اجتماعی ۱۰۷
۴-۱ شبکه‌های اجتماعی ۱۰۸
۴-۲ خواص یک شبکه اجتماعی ۱۰۹
۴-۲-۱ شبکه‌های بدون مقیاس ۱۰۹
۴-۲-۲ شبکه‌های جهان کوچک ۱۱۳
۴-۲-۳ ناوبری شبکه ۱۱۵
۴-۲-۴ عدد دانبار ۱۱۵
۴-۳ جمع‌آوری داده‌ها در شبکه‌های اجتماعی ۱۱۵
۴-۴ شش درجه جدایی ۱۱۸
۴-۵ شبکه‌های اجتماعی آنلاین ۱۱۸
۴-۶ جمع‌آوری داده‌های اجتماعی آنلاین ۱۱۹
۴-۷ نمونه‌گیری داده ۱۲۱
۴-۸ تجزیه و تحلیل شبکه اجتماعی ۱۲۴
۴-۹ تحلیل شبکه اجتماعی در مقابل تحلیل پیوند ۱۲۶
۴-۱۰ توسعه تاریخی ۱۲۶
۴-۱۱ اهمیت تحلیل شبکه اجتماعی ۱۲۹
۴-۱۲ ابزارهای مدلسازی تحلیل شبکه اجتماعی ۱۳۰

فصل پنجم: تجزیه و تحلیل در سطح گره ۱۳۳
۵-۱ تجزیه و تحلیل شبکه فردانه ۱۳۴

۵-۲ شناسایی افراد تاثیرگذار در شبکه ۱۵۲
۵-۲-۱ مرکزیت درجه‌ای ۱۵۴
۵-۲-۲ مرکزیت نزدیکی ۱۶۰
۵-۲-۳ مرکزیت بینابینی ۱۶۳
۵-۲-۴ مرکزیت بردار ویژه ۱۶۵
۵-۳ رتبه-صفحه ۱۶۷
۵-۴ همسایگان ۱۷۵
۵-۵ پلها ۱۷۵
۵-۶ از کدام الگوریتم مرکزیت استفاده شود؟ ۱۷۷

فص ششم: تحلیل در سطح گروه ۱۷۹
۶-۱ زیرگروههای منسجم ۱۷۹
۶-۲ محافل ۱۸۱
۶-۳ ضریب خوشه‌گی ۱۸۶
۶-۴ تجزیه و تحلیل سه‌وَندی ۱۸۹
۶-۶ واسطه‌گری ۱۹۴
۶-۷ تراگذری ۱۹۹
۶-۸ هسته‌ای بودن ۲۰۴
۶-۹ اجتماعات همپوشان ۲۰۵
۶-۱۰ یافتن اجتماع پویا ۲۰۷
۶-۱۱ m-قطاع ۲۰۸
۶-۱۲ k-هسته‌ها ۲۰۹
۶-۱۳ اجتماع‌یابی ۲۰۹
۶-۱۳-۱ بخش‌بندی گراف ۲۱۰
۶-۱۳-۲ خوشه‌بندی سلسله‌مراتبی ۲۱۱
۶-۱۴ مدلهای بلوکه‌ای ۲۲۱
۶-۱۴-۱ بهینه‌سازی پیمانه‌ای ۲۲۹
۶-۱۵ روش لوون ۲۳۰

فصل هفتم: تحلیل در سطح شبکه ۲۳۳
۷-۱ مولفه تحلیل در سطح شبکه..... ۲۳۳
۷-۱ مولفه‌ها/ایزوله‌ها..... ۲۳۴
۷-۲ هسته/حاشیه..... ۲۳۴
۷-۳ چگالی..... ۲۳۵
۷-۴ کوتاهترین مسیر..... ۲۳۶
۷-۵ دوسویگی..... ۲۳۸
۷-۶ شبکه‌های وابستاری..... ۲۳۹
۷-۷ شبکه‌های دو-حالته..... ۲۴۲
۷-۸ همریختی..... ۲۴۵

فصل هشتم: انتشار اطلاعات در شبکه های اجتماعی ۲۵۹
۸-۱ انتشار..... ۲۶۰
۸-۲ واگیری یا سرایت..... ۲۶۲
۸-۳ انتشار نوآوری..... ۲۶۳
۸-۴ پذیرش نوآوریها..... ۲۶۵
۸-۵ مدلهای انتشار نوآوری..... ۲۶۵
۸-۶ مدل جریان دو مرحله‌ای..... ۲۶۶
۸-۷ شیوع اجتماعی..... ۲۶۹
۸-۸ نرخ پذیرش..... ۲۷۰
۸-۹ طبقه‌بندی و آستانه پذیرش..... ۲۷۰
۸-۱۰ میزان مواجهه..... ۲۷۱
۸-۱۱ پذیرندگان و پذیرش..... ۲۷۳
۸-۱۲ جِرم بحرانی..... ۲۷۶
۸-۱۳ همه‌گیری..... ۲۸۰
۸-۱۴ مدلهای همه‌گیری..... ۲۸۱
۸-۱۵ مدلهای بخش‌بندی..... ۲۸۲
۸-۱۶ مدل سار *(SIR)*..... ۲۸۳
۸-۱۷ خواص مدل سار *(SIR)*..... ۲۸۶

فصل نهم: کاربرد تجزیه و تحلیل شبکه در نگاشت گفتگوهای سیاسی برخط ۲۹۱
۹–۱ تاریخچه جنبش سیاسی پنج ستاره..... ۲۹۲
۹–۲ آمار تالار گفتگو..... ۲۹۴
۹–۳ تجزیه و تحلیل شبکه..... ۲۹۹
۹–۳–۱ شبکه دوحالته: مضامین و کاربران..... ۲۹۹
۹–۳–۲ تصویر یک‌حالته: طبقه‌بندی مضامین..... ۳۰۲
۹–۳–۳ تصویر یک‌حالته: کاربران..... ۳۰۴
۹–۴ تجزیه و تحلیل مناظره‌ها..... ۳۰۶
۹–۴–۱ مهاجرت..... ۳۰۸
۹–۴–۲ دولت ائتلافی..... ۳۱۰
۹–۵ نتیجه‌گیری..... ۳۱۱

فصل ۱۰–پیوستها ۳۱۵
پیوست الف: پایتون $X.3$، یک راهنمای نحوی (یا ساختاری) سریع..... ۳۱۵
قواعد نحوی پایتون..... ۳۱۷
متغیرها..... ۳۱۷
اعداد..... ۳۱۸
رشته‌ها..... ۳۱۸
فهرستها..... ۳۱۹
چندتاییها یا تاپلها..... ۳۲۰
دیکشنریها..... ۳۲۰
شرطها..... ۳۲۱
حلقه‌ها..... ۳۲۲
توابع پایتون..... ۳۲۳
مدیریت فایل..... ۳۲۴
مدیریت استثناها..... ۳۲۵
ماژولها..... ۳۲۵
کلاسها..... ۳۲۶

پیوست ب: آموزش *NetworkX* ۳۲۷
انواع گراف ۳۲۸
گره‌ها ۳۲۹
یالها ۳۲۹
گراف جهتدار ۳۳۰
گراف مشخصه‌دار ۳۳۱
گراف وزندار ۳۳۲
گراف چندگانه ۳۳۲
عملیات گراف کلاسیک ۳۳۲
تولیدکننده‌های گراف ۳۳۳
تحلیل شبکه پایه‌ای ۳۳۳
معیارهای مرکزیت ۳۳۴
رسم گراف ۳۳۵
بسته الگوریتمها (الگوریتمهای *NetworkX*) ۳۳۵
خواندن و نوشتن ۳۳۶

فصل ۱۱- واژه‌نامه ۳۳۷

منابع ۳۸۵

فصل اول

مفاهیم نظری تحلیل شبکه‌های اجتماعی

شبکه، در یک بیان کلی، مجموعه‌ای از اشیاء و اتصالات[1] بین آنهاست. این اشیاء (که به آنها گره[2] یا راس[3] نیز می‌گویند) ممکن است افراد، سازمانها، جاده‌ها، کامپیوترها، شهرها و غیره باشند. همچنین اتصالات شبکه (که به آنها پیوند[4] یا یال[5] می‌گویند) نشاندهنده "روابط میان گره‌ها" هستند، روابطی[6] همانند روابط دوستی، پیوند خویشاوندی، روابط کاری، جریان اطلاعات و غیره. بعنوان نمونه‌هایی از انواع شبکه‌ها می‌توان به شبکه‌های کامپیوتری، شبکه‌های عصبی، شبکه‌های معنایی، شبکه غذایی،

[1] Linkes
[2] Nodes
[3] Vertices
[4] Ties
[5] Edges
[6] Relation

شبکه‌های زنجیره تأمین، شبکه‌های ارتباطی، شبکه‌های اطلاعاتی و غیره اشاره کرد. برخی از نمادهای مورد استفاده در شبکه‌ها (مانند گره‌ها و یالها) از تئوری گراف و برخی دیگر (مانند نظریه کنشگر–شبکه[1]) از نظریه‌های اجتماعی سرچشمه می‌گیرند.

یک شبکه را می‌توان به‌صورت یک گراف یا ماتریس نشان داد تا روابط بین گره‌ها مشخص شود. با چنین نمایشی از شبکه، استفاده از روشهای مختلف ریاضی، محاسباتی و آماری برای استخراج و تحلیل ویژگیهای اصلی شبکه امکانپذیر است.

این فصل درآمدی بر "مفاهیم نظری تحلیل شبکه" است؛ البته نه همه مفاهیم نظری موجود بلکه مفاهیم اصلی و پایه‌ای. در واقع تأکید فصل اول بر مفاهیمی است که در فصول بعد این کتاب، مورد استفاده قرار می‌گیرد. به این دلیل، نمی‌توان ادعا کرد که این فصل یک فهرست جامع از همه موضوعاتی که در تحلیل گراف و شبکه وجود دارند را ارائه نموده است چرا که ادبیات مبحث شبکه‌ها، بسیار مفصل و غنی است.

۱-۱ معنای جامعه‌شناختی روابط شبکه‌ای

شبکه، از لحاظ جامعه‌شناختی، مجموعه‌ای از روابط میان کنشگران[2] است. منظور از کنشگران، همان افراد، گروهها، سازمانها، دولتها و امثال آن هستند. کنشگران موجود در یک شبکه ممکن است با هم رابطه داشته و یا نداشته باشند. رابطه بین گره‌ها در یک شبکه، پیوند نامیده می‌شود. یک پیوند می‌تواند معانی مختلفی داشته باشد. برای مثال، می‌تواند به معنای رابطه خویشاوندی بین افراد، وجود محبت یا خصومت، مبادله سود یا وام، عضویت در یک باشگاه، یا حضور با هم در رویداد خاصی باشد. پیوند

[1] Actor-Network Theory

[2] Actors

در کتب و منابع انگلیسی از واژه‌های گوناگونی برای بیان مفهوم گره و اتصال در شبکه و گراف استفاده کرده‌اند. برای اولی از واژه‌هایی همچون: گره، راس، کنشگر، عامل، فرد، بازیگر، فعال، نهاد، موجودیت و غیره استفاده شده است و برای دومی از واژه‌های مختلفی همچون: اتصال، ارتباط، لینک، یال، پیوند، رابطه، لبه، کمان، قوس، بردار، پیکان و امثال آن بهره برده شده است. همانطور که در پاورقی بالا دیدید، ما سعی کرده‌ایم برای هر کدام از کلمات انگلیسی، یک معادل فارسی برگزینیم و متناسب با واژه اصلی، آن معادلها را استفاده کنیم. هر چند در متن اصلی، گاهی این واژه‌ها برای سلاست و روانی کلام، بجای هم بکار رفته‌اند.

می‌تواند جهت‌دار یا بدون‌جهت باشد. پیوندهایی از قبیل آشنایی، خویشاوندی یا عضویت توام (یا مشترک) در سازمانها معمولاً بی‌جهت هستند، اما پیوند نویسندگان معمولاً جهت‌دار است.

شبکه‌های اجتماعی به‌واقع مدلهای نظری برای تحلیل و تجسم "روابط بین کنشگران" می‌باشند. روابط بین افراد ممکن است به لحاظ «شدت»[1] متفاوت باشد، یعنی گاهی اوقات، برخی کنشگران دارای روابط قوی‌تری از بقیه هستند. همچنین این امکان وجود دارد که بیش از یک نوع رابطه بین کنشگران یک شبکه داشته باشیم. بعنوان مثال، دانشجویان یک دانشکده ممکن است از طریق دوستی، یا دروس مشترک، یا عضویت در باشگاه و یا غیره با یکدیگر مرتبط باشند، ولیکن نوع رابطه در هر یک از این حوزه‌ها به لحاظ شدت و ضعف با یکدیگر متفاوت است.

مفهوم مهم دیگری که به روابط میان کنشگران در شبکه‌های اجتماعی مربوط می‌شود، تراگذری[2] است. به‌طور مثال فرض کنیم که می‌دانیم شخص a شخص b را می‌شناسد. شخص b نیز c را می‌شناسد. آیا می‌توان نتیجه گرفت که شخص a، شخص c را نیز می‌شناسد؟ (این موضوع در ادامه همین فصل و نیز در فصل ۶ بطور مفصل بحث خواهد شد).

ساختار رابطه‌ای[3] یک شبکه اجتماعی را می‌توان به‌صورت یک گراف از گره‌ها $N=\{1,2,3\ldots n\}$ نشان داد که در آنها هر جفت گره، توسط یک یال به همدیگر متصل می‌شوند. گراف را همچنین می‌توان بعنوان "ماتریس مجاورت"[4] مثل X نمایش داد، به طوریکه X_{ij}، نشاندهنده رابطه بین گره i (فرستنده) و گره j (گیرنده) است. X_{ij}، می‌تواند یک مقدار باینری یک یا صفر باشد که نشانگر وجود یا عدم وجود رابطه بین i و j است.

ازآنجا که کنشگران در یک شبکه اجتماعی احتمالاً بر اساس ویژگیهایی همچون سن، جنس، مذهب، زبان، آموزش و غیره با یکدیگر مرتبط هستند، این امکان وجود دارد که آنها را بر اساس یک معیار

[1] Intensity
[2] Transitivity
بعنوان معادل این کلمه، واژگانی همچون تراگذری، انتقال‌پذیری، تریا، متعددی و غیره مطرح شده که بر مبنای انتقال یک مفهوم یا یک رابطه در طی یک زنجیره سه‌تایی است. وقتی یک رابطه و یا یک مفهوم از گره اول به دوم و نیز از گره دوم به سوم برقرار باشد، آنگاه در صورت وجود تراگذری کامل، این رابطه یا مفهوم، همچنین بین گره اول و سوم نیز وجود دارد.
[3] Relational Structure
[4] Adjacency Matrix

مشابهت، خوشه‌بندی[1] کنیم. این مسئله چندین کاربرد دارد. بعنوان مثال، سازمانها می‌توانند از خوشه‌بندی کاربران در گروهها استفاده کرده و برای هر گروه از مصرف‌کنندگان، یک محصول متفاوت توسعه دهند. این امر همچنین امکان یافتن کنشگران کلیدی و کاربران متنفذ را فراهم می‌آورد که احتمالاً در محافل اجتماعی محلی خود تأثیرگذار هستند. (موضوع خوشه‌بندی در ادامه همین فصل و در فصل ۶ بحث خواهد شد).

انواع مختلفی از شبکه‌های اجتماعی وجود دارد که در آنها، نوع رابطه بین کنشگران از یک شبکه به یک شبکه دیگر متفاوت است. چند نمونه از آنها در زیر آمده است:

- در بسیاری از شبکه‌های اجتماعی آنلاین (بعنوان مثال، فیسبوک) اغلبِ روابط، نوعی رابطه دوستی بین دو دوست، یعنی یک "رابطه متقابل" را نشان می‌دهند. در این حالت می‌توان روابط را بصورت یک یال جهت‌دار (با دو سر) نشان داد، اما اگر به دنبال نمایش برابری یا تعادل میان دو گره هستیم، می‌توان از یک یال بدون جهت استفاده کرد.

- در برخی از شبکه‌های دیگر (بعنوان مثال، توییتر) روابط بنوعی از جنس دنبال کردن[2] هستند، که ضرورتاً متقابل نیستند، یعنی "رابطه نامتقابل". به این معنی که مثلاً فرد a می‌تواند افراد b و c و d و غیره را دنبال کند، اما لازم نیست که این افراد، فرد a را دنبال کنند. این نوع رابطه به‌صورت یک یال با یک سر (اصطلاحاً یال جهت‌دار) نشان داده می‌شود.

- یک شبکه ارتباطات اجتماعی[3] نوعی شبکه اجتماعی است که در آن گره‌ها، کاربران یک خدمت خاص بوده (بعنوان مثال، کاربران تلفن) و روابط (یا یالها) عملاً ارتباطات میان کاربران در یک رویداد (مثلاً صحبت کردن) را نشان می‌دهند. این شبکه‌ها را همچنین می‌توان به عنوان نوعی از انواع "گرافهای اجتماعی" در نظر گرفت. این بدان دلیل است که می‌توان انواع مختلف روابط اجتماعی را بصورت یک گراف ترسیم کرد تا نمایشگر شبکه‌های ارتباطات یا دوستی بین کنشگران باشد. چنین گرافی، امکان اتصال کاربران و توزیع محتوای آنها را فراهم می‌کند. در این حالت،

[1] Clustering
[2] Follow
[3] Social Communication Network

قدرت رابطه بین دو کاربر مثلاً از طریق شمارش تعداد تماسهای برقرارشده بین آن دو، بدست می‌آید.

- "شبکه‌های اعتماد"[1] نیز نوع دیگری از گرافهای اجتماعی هستند که در آن رابطه بین دو موجودیت اجازه می‌دهد تا یک موجودیت (یعنی اعتماد کننده[2]) بر اقدامات انجام‌شده توسط نهاد دیگر (یعنی معتمد[3]) تکیه (یا اعتماد) کند. این موجودیتها یا نهادها می‌توانند دو شخص، یک شخص و یک شیء یا گروهی از خانواده‌ها، شرکتها، کشورها و غیره باشند.

- در "شبکه‌های استنادی"[4]، یک یال جهت‌دار یا فلش می‌تواند برای توصیف رابطه استنادی استفاده شود؛ یعنی یک یال از سوی کنشگر a (که استناد کرده است) به سمت کنشگر b (که مورد استناد واقع شده) کشیده شود.

- در وبلاگ‌نویسی، تنها وقتی می‌توان رابطه‌ای بین فرد a و b ایجاد کرد که b در مورد یک پست منتشرشده توسط a اظهارنظر کند.

یک شبکه اجتماعی گاه حاوی میلیونها کاربر است. برخی از این کاربران فعال‌تر و قدرتمندتر از سایر کاربران هستند. به‌منظور ارزیابی و تحلیل کاربران مختلف در یک شبکه اجتماعی، چندین معیار سنجش شبکه مانند مرکزیت درجه‌ای[5]، مرکزیت نزدیکی[6]، مرکزیت میانی یا بینابینی[7] و مرکزیت بردار ویژه[8] توسعه داده شده‌اند. این سنجه‌ها بر اساس یک معیار مشخص، کاربرانی را که از بقیه مهم‌تر هستند، شناسایی می‌کنند. برای مثال، "مرکزیت درجه‌ای"، تعداد لینکهای مستقیم به کاربران دیگر را بعنوان معیار در نظر می‌گیرد. "مرکزیت نزدیکی" میزان نزدیکی یک کاربر به کاربران دیگر در شبکه را اندازه‌گیری می‌کند. "مرکزیت میانی یا بینابینی" میزان اهمیت یک کاربر بعنوان داشتن نقش رابط میان

[1] Trust Networks
[2] Trustor
[3] Trustee
[4] Citation Networks
[5] Degree Centrality
[6] Closeness Centrality
[7] Betweenness Centrality
[8] Eigenvector Centrality

بخشهای مختلف شبکه را مشخص می‌کند و "مرکزیت بردار ویژه"، میزان مهم بودن یک کاربر را بر اساس ارتباط داشتن با کاربران مهم و متنفذ درون شبکه تعریف می‌نماید. (معیارهای سنجش هم در این فصل و هم در فصول دیگر و بویژه در فصل ۵ بحث می‌شوند).

۲-۱ سنجه‌های شبکه

تعداد زیادی از ویژگیها یا مشخصات قابل اندازه‌گیری شبکه به منظور شناخت بیشتر و بینش عمیق‌تر شبکه‌ها توسعه‌یافته‌اند که بسیاری از آنها ریشه در مطالعات اجتماعی بویژه در حوزه روابط میان فعالان اجتماعی دارند. در این بخش، در مورد سه دسته از سنجه‌هایی که برای تحلیل شبکه اجتماعی تعریف شده‌اند، بحث خواهیم کرد:

- **همبندی یا اتصال شبکه**[۱] که شامل معیارهایی همانند تراگذری، چندگانگی[۲]، همریختی، دووَندها، سه‌وَندها و دوسویگی[۳] است.

- **توزیع شبکه** که شامل معیارهایی همچون فاصله بین گره‌ها، مرکزیت درجه‌ای، مرکزیت نزدیکی، مرکزیت میانی یا بینابینی، مرکزیت بردار ویژه، رتبه–صفحه [۴]، فاصله کروی (ژئودزیک)[۵] و کوتاهترین مسیر، گریز از مرکز (یا برون‌مرکزی)[۶] و چگالی است.

- **بخش‌بندی شبکه**[۷] که شامل سنجشهایی همچون زیرگروههای منسجم[۸]، گروههای محفلی[۹]، ضریب خوشه‌گی، k–هسته‌ای، هسته و حاشیه، مدلهای بلوکه‌ای[۱۰] و خوشه‌بندی سلسله مراتبی است.

[1] Network Connection
[2] Multiplexity
[3] Reciprocity
[4] PageRank
[5] Geodesic Distance
[6] Eccentricity
[7] Network Segmentation
[8] Cohesive Subgroups
[9] Cliques
[10] Blockmodels

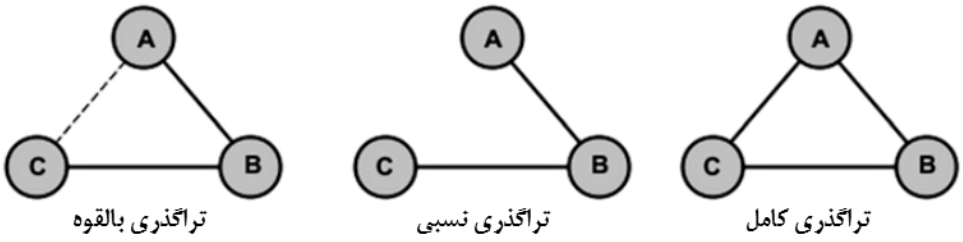
۱-۲-۱ همبندی شبکه

همبندی یا اتصال شبکه (و بعبارتی پیوستگی یا متصل بودن آن) به توانایی حرکت از یک گره به گره دیگر در یک شبکه اشاره دارد. همبندی شبکه، در عمل، نسبت بین فاصله معمولی یک مسیر به فاصله کروی (کمینه یا ژئودزیک) آنست. این معیار را می‌توان به‌صورت محلی (برای بخشی از شبکه) و یا در سطح فرامحلی (برای کل شبکه) محاسبه کرد. برخی از سنجه‌ها یا معیارهای مهم همبندی شبکه در ادامه بررسی می‌شود.

۱-۲-۲ تراگذری

تراگذری یک ویژگی شبکه است که نشان می‌دهد رابطه بین دو گره (کنشگر) در شبکه تا چه حد دارای خاصیت انتقال‌پذیری می‌باشد. این معیار در شبکه‌های اجتماعی بسیار مهم است اما در سایر شبکه‌ها اهمیت کمتری دارد. در شبکه‌های اجتماعی، اصطلاح تراگذری مفهوم دوستِ یک دوست را نشان می‌دهد و گاهی اوقات بعنوان مترادف ضریب خوشه‌گی کل شبکه[1] استفاده می‌شود.

فرض کنید که فرد (کنشگر) A از طریق یک یال به فرد (کنشگر) B متصل است و B با فرد C ارتباط دارد. آیا می‌توان نتیجه گرفت که A با C رابطه دارد؟ (شکل۱-۱)



شکل ۱-۱ تراگذری بین گره‌ها

[1] Whole-Network Clustering Coefficient

پاسخ بله به سؤال فوق مستلزم آن است که سه‌وَند مستخرج از شبکه (یعنی زیرگراف مستخرجی که متشکل از سه گره است) یک رابطه تراگذری ایده‌آل را نشان دهد و پاسخ خیر گویای آن است که سه‌وَند فوق، فاقد یک رابطه منسجم در میان گره‌ها باشد. متوسط تراگذری در یک شبکه اجتماعی به‌صورت نسبت تعداد مثلثها به تعداد سه‌وَندهای همبند تعریف می‌شود.

$$transitivity = \frac{3 \times (\text{تعداد مثلثهای موجود})}{\text{تعداد سه‌وندهای متصل}} \qquad ۱-۱$$

ضریب ۳ در صورت کسر به این دلیل است که هر مثلث با سه تا از سه‌وندهای همبند در گراف ارتباط دارد (که هر یک از آنها در اطراف یک گره تمرکز یافته‌اند). مقدار ۱ برای این تساوی به این معنی است که این شبکه تمام یالهای ممکن را دارا هست (گراف کامل است). در شبکه‌های اجتماعی واقعی، معمولاً مقدار شاخص تراگذری، بین ۰٫۳ تا ۰٫۶ است.

تراگذری یک گراف به‌شدت وابسته به ضریب خوشه‌گی است. ضریب خوشه‌گی، شاخصی است که نشان می‌دهد تا چه حد گره‌ها تمایل دارند زیرگرافِ چگال یا متراکم[1] تشکیل دهند. در واقع هردوی این شاخصها، فراوانی نسبی تعداد مثلثها را اندازه‌گیری می‌کنند. این دو معیار رایجترین شاخص جهت محاسبه تعداد مثلثها در یک شبکه می‌باشند.

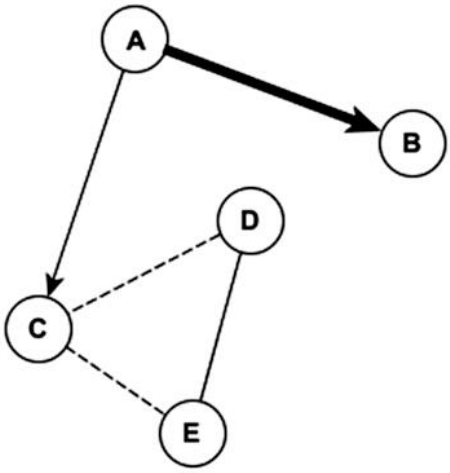
۱-۲-۳ چندگانگی

پیوندها یا روابط می‌توانند قدرت داشته باشند، که قدرت آنها را می‌شود از طریق شدت رابطه یا تنوع محتوای جابجا شده بر روی یالها سنجید. پیوندی با تنها یک بُعد را پیوند یگانه[2] می‌نامند. درعین‌حال، پیوندی با بیش از یک بُعد را پیوند چندگانه[3] می‌نامند. معمولاً پیوندهای چندگانه از پیوندهای یگانه قویتر

[1] Dense Subgraphs
[2] Uniplex
[3] Multiplex

هستند. گره‌ها و پیوندها را می‌توان کنار یکدیگر قرارداد تا یک نمودار خاص (همانند شکل ۲-۱) تشکیل دهد که به آن جامعه‌نگار یا جامعه‌نما (سوسیوگرام[1]) می‌گویند.

در شکل ۲-۱، دایره‌ها، نشاندهنده گره‌ها و خطوط میان آنها، نشاندهنده پیوندها هستند. خط پررنگ، نشانگر ارتباطات قوی بوده، خط تیره (یا خط فاصله) ارتباطات ضعیف داشته و خطوط معمولی، نشاندهنده ارتباطات با قدرت میانی هستند. وجود پیوندی بصورت یک فلش یا پیکان، بدین معناست که یک پیوند جهتدار وجود دارد. در اینجا A، فی‌المثل مطلب B را دوست دارد، (اگر شکل ما یک جامعه‌نمای دوستی یا پسندیدن باشد)، اما B، علاقه‌ای به مطلب A ندارد.



شکل ۲-۱ جامعه‌نمای ساده

پیوندهایی که دارای جهت نیستند را روابط بی‌جهت یا بی‌طرف[2] گویند و معمولاً بدان معناست که آنها، روابطی دوجانبه یا متقابل هستند. برای مثال، E در اینجا D را دوست دارد و D هم متقابلاً E را دوست دارد. از طرف دیگر، برای برخی پیوندهای دوجانبه یا متقابل (مانند برادر بودن) تعریف جهت‌دار بودن بی‌معنی می‌باشد.

[1] Sociogram
[2] Directionless

جامعه‌نماها یک روش خوب برای نگاه کردن به تصویر ساختار اجتماعی هستند. با این‌حال، استفاده از آنها برای تحلیل شبکه به‌ویژه برای شبکه‌های بزرگ، آسان نیست. روش بهتر برای انجام تجزیه و تحلیل شبکه بالا، استفاده از ماتریسهاست. یک ماتریس به بیان ساده، جدولی از اعداد است که در سطرها و ستونها چیده شده‌اند (جدول۱-۱).

جدول ۱-۱ ماتریس مربوط به جامعه‌نما

	A	*B*	*C*	*D*	*E*
A	۰	۱	۱	۰	۰
B	۰	۰	۰	۰	۰
C	۰	۰	۰	۱	۱
D	۰	۰	۱	۰	۱
E	۰	۰	۱	۱	۰

تقاطع هر سطر و ستون ماتریس، یک سلول نامیده می‌شود. هر سلول دارای مقداری است که می‌تواند صفر یا یک باشد. اگر کنشگری در یک ردیف، یالی به سمت کنشگری در یک ستون، داشته باشد، مقدار سلول مربوطه، عدد یک و در غیر این صورت، صفر است. در قطر اصلی ماتریس، سلولهایی هستند که گره‌های ارسال‌کننده و دریافت‌کننده آنها یکسان هستند.

جدول ۲-۱ ماتریس مربوط به سوسیوگرام که قدرت ارتباطات را نیز بیان می‌کند.

	A	*B*	*C*	*D*	*E*
A	۰	۳	۲	۰	۰
B	۰	۰	۰	۰	۰
C	۰	۰	۰	۱	۱
D	۰	۰	۱	۰	۲
E	۰	۰	۱	۲	۰

از آنجا که ما اجازه نمی‌دهیم تا گره‌ها یالی به سمت خود داشته باشند، مقدار هرکدام از این سلولها (در قطر ماتریس) صفر در نظر گرفته می‌شود. با این‌حال، برای نمایش گره‌هایی که به آنها اجازه خود-

اتصالی داده‌ایم، می‌توان آنها را به شکل پیکانهایی که به خود گره‌ها بازمی‌گردند (همانند حلقه یا طوقه)، ترسیم کرد. در این حالت، مقدار قطر اصلی برای هر یک از این گره‌ها دیگر صفر نیست.

جدول ۲-۱، علاوه بر نمایش ساده روابط بین کنشگران با استفاده از اعداد صفر و یک (که نمایشگر وجود یا عدم وجود ارتباط بین آنهاست)، می‌تواند قدرت روابط را نیز نشان دهد. بعنوان مثال، اگر پیوندها بصورت چندگانه باشند، مقادیر موجود در سلولها می‌توانند شماره مربوط به انواع مختلف روابط بین گره‌ها باشند. واحدهای سنجش ساده‌ای همچون ضعیف، متوسط یا قوی نیز می‌تواند برای نشان دادن شدت یک رابطه بین جفت گره‌ها استفاده شوند؛ که شماره آنها به ترتیب می‌تواند یک، دو، سه در نظر گرفته شود. سلولهای مربوط به گره‌هایی که هیچ پیوندی بین آنها وجود ندارند نیز با صفر نشان داده می‌شوند (جدول ۲-۱).

۱-۲-٤ همریختی

یک نیاز مهم برای مطالعه شبکه‌های اجتماعی، همانا شناسایی گروهها و بررسی روابط بین آنها است. رابطهٔ میان گروهها در واقع محرک مهمی برای حفظ ثبات در جوامع انسانی و نیز حیات وحوش در طول زمان است و به همین جهت، معیار مهمی برای ایجاد یال در شبکه‌های اجتماعی محسوب می‌شود.

همگونی یا همریختی[۱] تمایل افراد برای برقراری ارتباط با کسانی است که نگرش و عقاید یکسانی دارند. تمایل افراد برای ارتباط با دیگران بر اساس ویژگی مشترک همچون جنسیت، تحصیلات، نژاد یا سایر خصوصیات اجتماعی-اقتصادی در انجمنهای اجتماعی بسیار رایج است. هماهنگی و همکاری معمولاً بین افرادی که شباهتهایی با هم دارند، موفق‌آمیزتر است. اینگونه افراد احتمالاً در موضوعاتی مانند شنیدن ایده‌های جدید از زبان هم و یا درخواست کمک از یکدیگر، مشتاق‌ترند.

[۱] Homophily

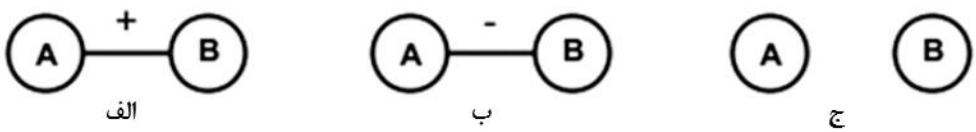
بعنوان معادل واژه هموفیلی، واژگانی همچون همگونی، همریختی، یک ریختی، تجانس، متجانس، هم‌شکلی، یکسانی و همسانی مطرح شده، که ما بیشتر از واژه همریختی و بعضاً از واژه همگونی استفاده کرده‌ایم.

همگونی یا همریختی در حوزه شبکه اجتماعی آنلاین را می‌توان بعنوان شباهت کاربران شبکه به لحاظ سن، سابقه تحصیلات، منطقه یا تخصص مشاهده کرد. همریختی در حوزه شبکه‌های همکارانه[1]، بعنوان شباهت شرایط حرفه‌ای یا دانشگاهی افراد همکار تعبیر می‌شود.

۱-۲-۵ دووَندها و تقابل

دووَند (یا دوگاه)[2]، یک جفت کنشگر در یک شبکه است که بطور بالقوه، بواسطه یک رابطه اجتماعی به یکدیگر متصل هستند. در واقع دووَند ساده‌ترین گروه اجتماعی موجود است که در آن انتقال اطلاعات بین دو گره، سریع و آسان است. دووَندها ازجمله مفاهیمی هستند که وضعیت شبکه‌های اجتماعی را مشخص می‌کنند. وجود پیوندهای دووَندی (یا پیوندهای دووَندانه) بین سازمانها، در مبحث همکاری میان شرکتها، ممکن است اجازه تبادل اطلاعات و دانش، سرمایه‌گذاری مشترک، استفاده از منابع ملموس و ناملموس یکدیگر، ارتباطات هیئت‌مدیره و بسیاری دیگر از کارهای مشترک را بدهد. یک دووَند می‌تواند در یکی از این سه حالت باشد: یک رابطه مثبت، یک رابطه منفی و فقدان ارتباط بین گره‌های دووَند (دیود خالی).

در بخشهای الف و ب شکل ۳-۱، گره‌ها نشاندهنده افراد و یال بین آنها نشاندهنده ارتباط مثبت یا منفی میان آنها هست. بخش ج شکل نشان می‌دهد که گره A و گره B با هیچ یالی متصل نیستند.



شکل ۳-۱ انواع روابط بین دو گره (الف: رابطه مثبت، ب: رابطه منفی، ج: بی رابطه)

[1] Corporate Networks

[2] کلمه Dyad به معنای "دوتا بودن" است، که در اینجا برای یک"ترکیب دو گرهی" استفاده شده است. کلمات زیادی در زبان انگلیسی هستند که معنای مشابهی را دارند. کلماتی مانند bipartite, double, dual, duplex binary , و امثال آن. در زبان فارسی نیز برای این کلمات معادلهایی همچون زوجی، جفتی، دوتایی، دوگانه، دودویی و همانند آن را مطرح می‌کنند. ما در اینجا برای تمایز بین اینها، واژه "دووند" و "دوگاه" را پیشنهاد داده‌ایم. کلمه "دووَند" را بعنوان مخفف کلمه "دوآوند" گرفته‌ایم که برخی معنای کلمه "آوند" را کوزه، ظرف و منبع گفته‌اند. واژه "دوگاه" نیز یک اصطلاح موسیقیایی است که بعضاً بعنوان مخفف کلمه "دوگانه" بکار می‌رود.

۱-۲-٦ سه‌وَندها و تعادل

سه‌وَند (یا سه‌گاه)[۱]، یک ساختار شبکه‌ای متشکل از سه کنشگر و سه دووَند است. با داشتن یک گراف کامل از سه کنشگر (سه‌وَند)، ما می‌توانیم چهار نوع مختلف از روابط را با توجه به تعداد روابط منفی بین گره‌ها شناسایی کنیم: ۱- یک دوستِ دوستِ من، دوست من است، ۲- یک دشمنِ دشمنِ من، دوست من است، ۳- یک دوستِ دوستِ من، دشمن من است و ۴- یک دشمنِ دشمنِ من، دوست من است.

وقتی یک علامت به یک پیوند داده می‌شود، ممکن است سؤال بعدی این باشد که چگونه این علامت در ارتباط با سایر علائم در همسایگی محلی یا با علائم دیگر در سطح شبکه، ارتباط برقرار می‌کند؟

نظریه تعادل ساختاری[۲]، مفهوم کلیدی در بسیاری از کاربردهای نظریه گراف علامتدار[۳] است. این روش سعی دارد توضیح دهد که چگونه یک الگوی خاص از علائم منفی و مثبت می‌تواند منجر به نوع متفاوتی از روابط شود. این نظریه، روشهایی را بررسی می‌کند که با استفاده از آنها می‌توان سه‌وَندها را علامتدار کرد: سه‌وَندهایی که تعداد یالهای با علامت مثبت (+) آنها فرد است، متعادل[۴] هستند، در حالیکه سه‌وَندهایی که تعداد یالهای با علامت منفی (-) آنها فرد است، نامتعادل هستند. موقعیت گراف نامتعادل معمولاً باعث ایجاد تنش برای افراد (گره‌های) گراف مربوطه می‌شود.

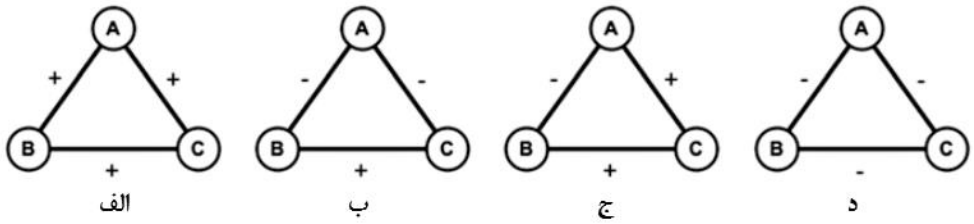
هردوی این ایده‌ها، یعنی "نظریه تعادل ساختاری" و "گرافهای علامتدار"، برای حل مسئله زیرگروهبندی[۵] در روانشناسی اجتماعی[۶] توسعه داده شده‌اند. بعدها متوجه شدند که این ایده‌ها، در حوزه‌های دیگری نیز کاربرد دارند. یکی از این حوزه‌ها، پیش‌بینی و توضیح تغییرات دوستی و دشمنی در

---

[۱] کلمه Triad به معنای "سه‌تا بودن" است که در اینجا برای یک"ترکیب سه گرهی" استفاده شده است. کلمات زیادی در زبان انگلیسی هستند که معنای تقریباً مشابهی دارند. کلماتی مانند trinity, trio, triple, triplet, trilogy, triptych, triplex, triplicate و امثال آن. در زبان فارسی برای این کلمات، معادلهایی همچون سه‌تایی، سه‌گانه و همانند آن را مطرح می‌کنند. ما در اینجا برای تمایز بین اینها، واژه "سه‌وَند" و "سه‌گاه" را پیشنهاد داده‌ایم. مبحث دووَندها و سه‌وَندها در فصل شش بیشتر تشریح می‌شود.

[2] Structural Balance Theory
[3] Signed Graph Theory
[4] Balanced
[5] Subgrouping
[6] Social Psychology

جوامع انسانی است؛ یعنی در تجزیه و تحلیل دشمنی در جنگهای قبیله‌ای، درگیریهای سیاسی یا روابط بین‌المللی می‌توان از آنها بهره برد. گرافی را با چهار حالت به‌صورت شکل ۴-۱ در نظر بگیرید:



شکل ٤-۱ گرافی با سه گره و در چهار حالت

گرافهای شکل ۴-۱ نوعی از گرافهای علامتدار هستند که از دهه ۱۹۵۰ مورد بررسی قرار گرفته‌اند. در واقع نوع خاصی از گرافهای علامتدار هستند که یالها می‌توانند یکی از دو مقدار علامت متضاد را داشته باشند تا احساسات مثبت یا منفی را منتقل کنند. نمونه‌هایی از گرافهای علامتدار عبارتند از: دوست/دشمن، اعتماد/بی‌اعتمادی، پسندیدن/نپسندیدن، ستایش/سرزنش، اثرگذار/تأثیرمنفی و غیره. این گرافها در جامعه‌شناسی و روانشناسی کاربرد بسیاری دارند، اما در زمینه‌هایی مثل فیزیک و شیمی، کمتر رایج هستند.

- در شکل الف، هر سه کنشگر علامت (احساسات) مثبت دارند و هیچ جایی برای درگیری بین آنها وجود ندارد. این پیکره‌بندی، منسجم بوده و فاقد تنشهای داخلی بین اعضاست.

- شکل ب نیز پایدار است زیرا دو کنشگر (B و C) احساس منفی مشابهی نسبت به کنشگر A دارند، اما آنها همدیگر را دوست دارند.

- شکل ج ناپایدار است زیرا کنشگران A و B احساس منفی نسبت به یکدیگر دارند، در حالیکه هر دو احساس مثبتی نسبت به کنشگر C دارند و کنشگر C باید وفاداری خود را بین دو بازیگر دیگر تقسیم کند.

- شکل د نیز ناپایدار بوده و درنهایت تجزیه می‌شود چون تعداد زیادی علائم منفی دارد.

در شکل ۴-۱ تصاویر الف و ب انواع زیرگرافهای متعادل را نشان می‌دهند، درحالی‌که تصاویر ج و د انواع گرافهای نامتعادل را نشان می‌دهند. یک راه بدیهی برای اجتناب از عدم تعادل در گراف، انتقال علائم[1] است، که شامل تغییر علامت می‌شود؛ مثلاً تغییر دشمنیها (علائم منفی) به دوستیها (علائم مثبت) و یا بالعکس. در شبکه‌های واقعی، تعداد پیکره‌بندیهای پایدار[2] به‌مراتب بیشتر از پیکره‌بندی ناپایدار است.

لازم به ذکر است که وجود علامت منفی بین دو گره، به معنای عدم ارتباط بین آنها نیست، بلکه یک علامت منفی بین دو گره به معنای یک رابطه خصمانه است، در حالیکه عدم وجود یال بین دو گره، نشاندهنده عدم وجود تعامل یا ارتباط بین آنها است.

۱-۲-۷ دوسویگی

معیار دوسویگی (یا دوجانبه‌گی یا عمل متقابل)، بمعنای سنجش تمایل به ایجاد ارتباطات دوجانبه بین دو کنشگر است. درواقع معیار دوسویگی معادل تعداد ارتباطات متقابل یا دوجانبه برای یک کنشگر مشخص در یک شبکه هست. بعنوان مثال، اگر u به v متصل شود، آنگاه v به u متصل می‌شود و بالعکس. در موقعیتهای زندگی واقعی، گاهی مثلاً مهم است که بدانیم آیا کمک دریافت شده، توسط کمک گیرنده جبران می‌شود و یا نه؛ یعنی زمانی که به شخصی کمک می‌کنیم، آیا وی آنرا تنها بعنوان یک کمک تصور می‌کند و یا برداشت دیگری داشته و بدنبال جبرانش می‌باشد.

برای یک گره معین v، دوسویگی به‌صورت نسبت تعداد گره‌های ورودی و خروجی از/به v، به تعداد گره‌هایی است که تنها اتصالات ورودی از v دارند.

دوسویگی برای کل شبکه، به‌صورت نسبتی از گره‌ها محاسبه می‌شود که متقابل یا دوجانبه هستند. متوسط دوسویگی از طریق میانگین دوسویگی تمامی گره‌های موجود در شبکه محاسبه می‌شود.

[1] Sign Shifting
[2] Stable Configurations

دوسویگی می‌تواند شاخصی برای اهمیت رابطه بین دو کنشگر باشد. در حالیکه یک رابطهٔ یکطرفه یا یکسویه[1] منجر به انتقال اطلاعات (بعنوان مثال، پیامها) از یک طرف به طرف دیگر می‌شود، یک رابطهٔ متقابل با این تعریف، قوی‌تر است زیرا به انتقال اطلاعات در هر دو جهت منجر می‌شود. در برخی موارد، رابطه نامتوازن[2] بوده و به سمت یکی از دو عامل سوگیری یا چولگی دارد، (بعنوان مثال، انتقال اطلاعات در یک جهت بیشتر از جهت دیگر است) که می‌تواند نشانه‌ای از تفاوت در وضعیت و قدرت بین دو طرف باشد.

در شبکه‌های دوستی مانند فیسبوک، از آنجا که انتظار می‌رود همه ارتباطات دوستی، متقابل باشند، مقدار معیار دوسویگی ۱ (یا ۱۰۰ ٪) است. با این حال، چنین مطلبی برای دیگر شبکه‌های اجتماعی آنلاین (مانند توییتر) صادق نیست، زیرا در این شبکه‌ها، پیروی (دنبال کردن) تنها مدلی است که برای نشان دادن روابط مورد استفاده قرار می‌گیرد.

۱-۳ توزیع شبکه

معیارهای مربوط به توزیع شبکه نشان می‌دهند گره‌ها و یالها چگونه در یک شبکه توزیع شده‌اند.

۱-۳-۱ فاصله میان دو گره

فاصله میان دو گره بعنوان معیاری برای تحلیل شبکه، به محاسبه تعداد یالهای میان هر جفت گره در یک شبکه می‌پردازد. اندازه‌گیری فواصل بین گره‌ها در گرافها، برای بسیاری از تحلیلها مانند خوشه‌بندی گراف و تشخیص داده‌های پرت، حیاتی است. گاهی از اندازه‌گیری فاصله برای بررسی تشابه میان دو گره استفاده می‌شود. هر یک از الگوریتمهای معمولی محاسبه کوتاه‌ترین مسیر (بعنوان مثال، دایکسترا[3]) را می‌توان برای شناسایی کوتاه‌ترین مسیر در یک شبکه و نیز محاسبه طول آن استفاده کرد.

[1] One-Way Relationship
[2] Skewed
[3] Dijkstra

همچنین می‌توان از اندازه‌گیری فاصله برای محاسبه "گریز از مرکز" یا برون‌مرکزی که حداکثر فاصله از یک گره معین به همه گره‌های دیگر در یک شبکه است، بهره برد. بعلاوه می‌توان "قطر شبکه"[1] را محاسبه کرد که معادل با بیشترین مقدار گریز از مرکز گره‌هاست و درنتیجه بیشترین فاصله بین گره‌ها را نشان می‌دهد.

در اغلب شبکه‌های اجتماعی، معیار کوتاه‌ترین مسیر بر اساس هزینهٔ انتقال از یک گره به سایر گره‌ها محاسبه می‌شود، در اینصورت هر چه مسافت مسیر طولانی‌تر باشد، هزینه آن بیشتر می‌شود.

در حالیکه ممکن است یالهای زیادی بین گره‌های یک اجتماع یا انجمن[2] (یعنی در درون انجمن) وجود داشته باشد، اما یالهای بین اجتماعات[3] (یعنی از یک انجمن به انجمن دیگر) بسیار کمتر است.

۱-۳-۲ مرکزیت درجه‌ای

در معیار مرکزیت درجه‌ای، اهمیت یک گره توسط تعداد گره‌هایی که به آن متصلند، تعیین می‌شود؛ یعنی بنوعی اندازه‌گیری تعداد ارتباطهای مستقیم یک کنشگر به کنشگران دیگر، در شبکه است. بواقع هرچه تعداد گره‌های مجاور یک گره بیشتر باشند، آن گره اهمیت بیشتری دارد؛ زیرا این گره مستقل از دیگر کنشگران می‌تواند با بخشهای بزرگی از شبکه ارتباط داشته باشد. این سنجه یک معیار موضعی یا محلی است زیرا مقدار آن بر اساس تعداد پیوندهای مستقیم یک کنشگر با دیگر کنشگران مجاورش، محاسبه می‌شود. معمولاً کنشگران در شبکه‌های اجتماعی با مرکزیت بالا، بعنوان هابها و یا کانالهای اصلی اطلاعات عمل می‌کنند.

---

[1] Network Diameter

[2] Community

بعنوان معادل واژه community از کلمات "انجمن" و "اجتماع" استفاده شده و باید دقت نمود که "اجتماع" در اینجا با "جامعه" که معنای گسترده‌تری دارد، متفاوت است. در واقع انجمن یا اجتماع، بخشهایی کوچکتر اما متراکمتر از جامعهٔ بزرگ است.

[3] Edges Between Community

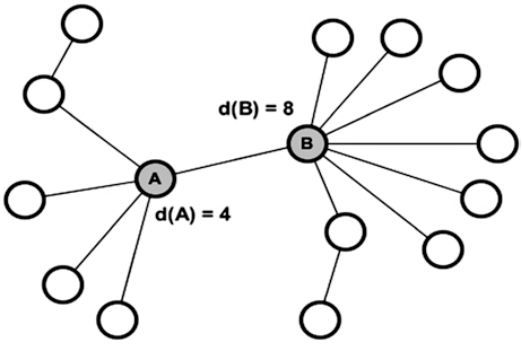
در شبکه‌های اجتماعی، فی‌المثل، توزیع درجه‌ای گره از "توزیع قانون قدرت"[1] پیروی می‌کند که به این معنی است که تعداد بسیار کمی از گره‌ها، تعداد بسیار زیادی از ارتباطات را دارا هستند. طبیعتاً، گره‌هایی با درجه بالا، عملاً تأثیر بیشتری در شبکه نسبت به سایر گره‌ها داشته و بنابراین اهمیت بیشتری دارند. مرکزیت درجه‌ای گره i را می‌توان به‌صورت رابطه ۲-۱ محاسبه کرد.

$$d(i) = \sum_j m_i \qquad ۲-۱$$

اگر یالی بین گره i و j وجود داشته باشد، مقدار m_{ij} آن برابر یک و در غیر این صورت معادل صفر خواهد شد. در شبکه‌های جهت‌دار مهم است که بین مرکزیت درجه‌ای ورودی و مرکزیت درجه‌ای خروجی تمایز قائل شویم.

در تحلیل شبکه‌ها، شناسایی افرادی با مرکزیت درجه‌ای بالا ضروری است، زیرا داشتن پیوندهای بسیار به معنی داشتن چندین راه برای برآورده کردن نیازها، وابستگی کمتر به بقیه و دسترسی بهتر به افراد دیگر و منابع شبکه است.

افرادی که بیشترین مرکزیت درجه‌ای را دارند، اغلب اشخاص ثالث و معامله‌ساز هستند و می‌توانند از این کارگزاری یا واسطه‌گری سود کسب کنند. برای شبکه‌های جهت‌دار، درجه ورودی[2] اغلب بعنوان قرینه‌ای برای محبوبیت و معروفیت به کار می‌رود (شکل ۵-۱).



شکل ۵-۱ مرکزیت درجه‌ای گره‌ها

[1] Power Law Distribution
[2] In-Degree

شکل ۵-۱ نشان می‌دهد که گره A و گره B دارای موقعیت ساختاری استثنایی (بیشترین مرکزیت درجه‌ای) هستند. همه خطوط ارتباطی باید از این دو گره عبور کنند. در نتیجه هر دو گره، صرفاً به خاطر موقعیت عالی خود، قدرتمند هستند. با این حال، استنتاج چنین یافته‌هایی تا حد زیادی وابسته به ماهیت پیوندها و ماهیت روابط موجود است.

۱-۳-۳ مرکزیت نزدیکی

مرکزیت نزدیکی را می‌توان بصورت میزان نزدیکی یک کنشگر معین به سایر کنشگران تعریف نمود. این مرکزیت از طریق مجموع فواصل ژئودزیک (کروی یا کمینه) یک گره به‌تمامی گره‌های دیگر در گراف بدست می‌آید. این معیار، طول مسیرها از یک گره به گره‌های دیگر در شبکه را محاسبه می‌کند.

طبق این تعریف، کنشگری مهم است که به بقیه کنشگران در شبکه نسبتاً نزدیک باشد. نمایش ریاضی مرکزیت نزدیکی، به‌صورت زیر است:

$$c(i) = \sum_j d_{ij} \qquad ۳-۱$$

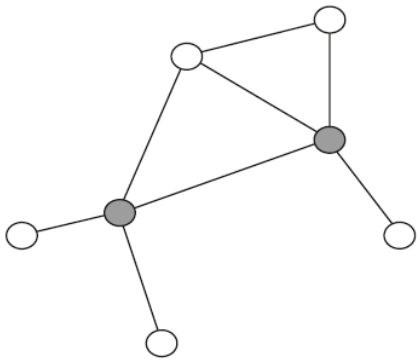
در اینجا d_{ij}، فاصله کروی میان گره i و j است (تعداد یالهای موجود در کوتاهترین مسیر بین گره i و j).

مرکزیت نزدیکی برای درک موضوع انتشار اطلاعات در شبکه‌ها مهم است زیرا فاصله بین یک گره خاص از سایر گره‌ها، بر نحوه دریافت یا ارسال اطلاعات آن گره اثرگذار است (بعنوان مثال، انتشار شایعات به همین صورت است). در شبکه‌های اجتماعی، این توانایی با پدیده‌ای که "افق قابل مشاهده"[1] نام دارد، محدود می‌شود. این پدیده بیان می‌کند که افراد بعد از دو مرحله (یعنی در فاصله دو گام از مرکز یک رویداد) تقریباً هیچ دید و اشرافی نسبت به رویداد مربوطه ندارند.

با تعریفی که انجام دادیم یعنی اینکه مرکزیت نزدیکی مبتنی بر فاصله بین گره‌های شبکه است، می‌توان آنرا معکوس مرکزیت دانست، زیرا مقادیر بالای این معیار به مرکزیت کمتری اشاره دارند و اندازه‌های

[1] Horizon of Observability

کوچک آن، نشان‌دهنده مرکزیت بالایی هستند. به لحاظ محاسباتی، مقدار $C(i)$ عددی بین صفر و یک خواهد شد، که هر چه مقدارش بیشتر باشد به نزدیکی بیشتری (یا متوسط مسافت کمتری) اشاره دارد و هر چه مقدارش کمتر باشد نمایشگر نزدیکی کمتر (یا متوسط مسافت بیشتری) است (شکل۶-۱).



شکل ٦-١ مرکزیت نزدیکی

در شکل ۶-۱، گره‌های خاکستری، بیشترین مرکزیت نزدیکی را دارا هستند، زیرا می‌توانند براحتی و بطور مساوی به سایر گره‌ها در شبکه دسترسی پیدا کنند. این گره‌ها همچنین قابلیت دسترسی به سایر گره‌ها در سریع‌ترین زمان ممکن را دارند. گره‌های دیگر عملاً فاقد چنین جایگاه ممتازی هستند.

با توجه به اینکه تعیین مرکزیت نزدیکی بر اساس محاسبات کوتاهترین مسیر است، بعضاً مشاهده شده که بکارگیری آن در شبکه‌های بزرگ، نتایج نسبتاً مشابهی برای برخی گره‌ها ببار می‌آورد که در نتیجه تمایز بین گره‌ها را سخت می‌کند.

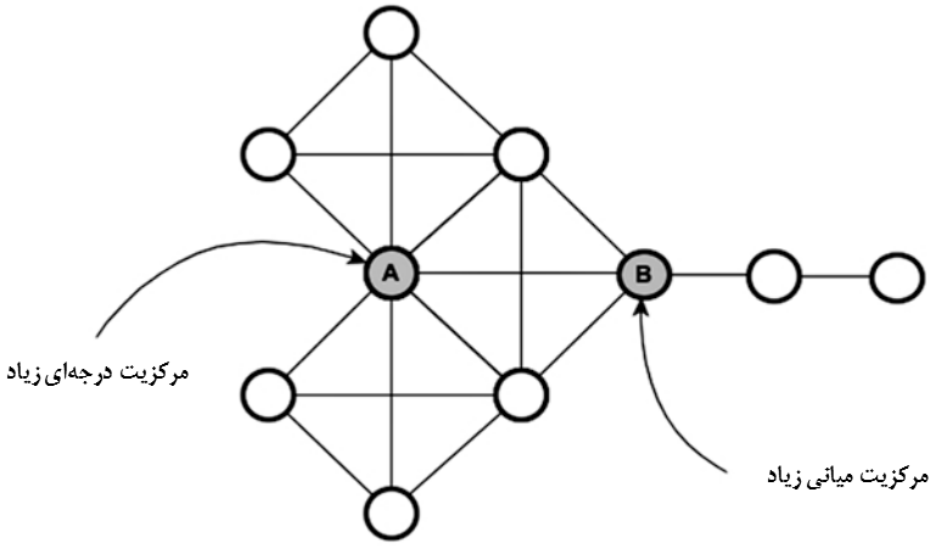
در شبکه‌های اطلاعاتی، مرکزیت نزدیکی نشان می‌دهد که چقدر طول می‌کشد تا یک بیت اطلاعات از یک گره به سایر گره‌ها در شبکه جریان پیدا کند. گره‌هایی با مرکزیت نزدیکی بالا معمولاً مسیرهای کوتاهتری برای دسترسی به بقیه گره‌ها در شبکه دارند.

۱-۳-٤ مرکزیت میانی یا بینابینی

مرکزیت میانی یا بینابینی را می‌توان بعنوان میزان اهمیت یک کنشگر برای ایفای نقش یک رابط یا یک پل، مابین شبکه‌های مختلف توصیف کرد. این مرکزیت نشاندهنده تعداد دفعاتی است که یک کنشگر باید از طریق گره پل-گونه گذر کند تا به یک کنشگر دیگر برسد. گره‌هایی با مرکزیت میانی بالا، جریان اطلاعات را کنترل می‌کنند چون آنها در نقش پلهای بحرانی مابین کنشگران یا گروهی از کنشگران ظاهر می‌شوند. مرکزیت میانی گره i به‌صورت رابطه ۴-۱ محاسبه می‌شود.

$$b(i) = \sum_{j.k} \frac{g_{ijk}}{g_{jk}} \qquad ۴-۱$$

در رابطه ۴-۱، عبارت g_{jk}، نشاندهنده تعداد کل کوتاهترین مسیرهای موجود بین دو گره j و k است (j و k مخالف i هستند) و عبارت g_{jik}، نشاندهنده تعداد کوتاهترین مسیرهای موجود بین دو گره j و k است که از گره i عبور می‌کنند.



شکل ۷-۱ مرکزیت میانی گره‌ها

مرکزیت میانی نشان می‌دهد که کدام گره‌ها احتمالاً گذرگاه و یا تنگه ارتباطات و اطلاعات هستند. اینها همان گره‌هایی هستند که فقدانشان، موجب جدایی بخش یا بخشهایی از گراف می‌شوند. به‌طور مشابه، این مرکزیت روشی برای شناسایی کسانی است که بعنوان پل عمل می‌کنند (مهره‌های اتصال مرزی[1] نیز نامیده می‌شوند) که اگر نباشند، دو یا چند بخش گراف قادر به ارتباط با یکدیگر نیستند.

این معیار بر اهمیت نقش یک کنشگر حاضر در میانه میدان تعاملات (و یا در صحنه ارتباطات) اجتماعی تأکید دارد و اینکه تا چه حد به وجود او بعنوان یک حلقه وصل در زنجیره اتصالات و تماسهای شبکه، نیاز است.

مرکزیت میانی (یا بینابینی) یک گره i بعنوان تعداد کوتاهترین مسیر بین جفت گره‌ها که از گره i عبور می‌کنند تعریف می‌شود. مرکزیت میانی یک گره اگر در کوتاه‌ترین مسیر بین هر دو راس دیگر در شبکه قرار نگیرد، برابر با صفر است که نشاندهنده نقش اجتماعی ناچیز او در حفظ ارتباطات شبکه است (شکل ۷-۱).

در تصویر ۷-۱، گره A دارای ارزش مرکزیت درجه‌ای خوبی است زیرا در وسط بخش بزرگی از نمودار قرار دارد. از طرف دیگر گره B ارزش بینابینی خوبی دارد زیرا دو بخش مختلف از شبکه را به هم متصل می‌کند. این قدرت بیشتر به این دلیل است که بر یک گلوگاه یا تنگنای ارتباطی نظارت یا حاکمیت می‌کند. تمام ارتباطات بین گره‌ها در بخش سمت راست گراف و قسمت چپ آن باید از گره B عبور کنند. فردی در آن موقعیت، یک نامزد خوب برای هر کار مربوط به تبلیغات مستقیم، عملیات اطلاعاتی و یا جمع‌آوری هوشمند اطلاعات است. باوجود اینکه موقعیت گلوگاه یا تنگنا می‌تواند موجب کسب قدرت باشد، اما در عین‌حال، یک منبع قابل‌توجه استرس و فشار نیز هست.

در شبکه‌های اعتماد، مفهوم کلیدی در شبکه، همانا مرکزیت میانی یا بینابینی است؛ زیرا این معیار عملاً پتانسیل فرد خاصی برای کنترل ارتباط بین افراد دیگر در یک شبکه را توصیف می‌کند. بکارگیری این مفهوم در محیط مدرسه، فی‌المثل، به یافتن افرادی کمک می‌کند که دوستانشان در گروههای اجتماعی متفاوت غیرهمپوشان (به لحاظ گویشی، زبانی، مذهبی، نژادی و امثال آن) هستند و لذا این افراد

[1] Boundary Spanners

"پل–گونه" گزینه ایده‌آلی برای ایفا کردن نقش رابط برای حفظ ارتباطات و تعاملات درون–اجتماعی این گروهها می‌باشند.

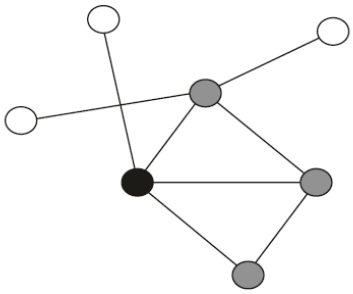
۱-۳-۵ مرکزیت بردار ویژه

معیار مرکزیت بردار ویژه، مرکزیت یک فرد را با توجه به ساختار کلی شبکه توصیف می‌کند. این معیار یک امتیاز نسبی به هر گره موجود در شبکه، نسبت می‌دهد. محاسبه این امتیاز نسبی بر اساس این مفهوم است که ارتباط با گره‌های دارای امتیاز بالا (در مقایسه با ارتباط با گره‌های امتیاز پایین) کمک بیشتری به کسب امتیاز نسبی می‌کند.

در واقع این روش میزان اتصال یک گره به گره‌های مهم یا نافذ را اندازه‌گیری می‌کند. این معیار با استفاده از بردار ویژه اصلی ماتریس مجاورت محاسبه می‌شود. محاسبه مرکزیت در این روش (که بردار ویژه را برای محاسبه مرکزیت اعمال می‌کند) عملاً با روش مرکزیت درجه‌ای (که تعداد ارتباطات هر گره را محاسبه می‌کند) متفاوت است. اگر ما تنها برخی از معیارهای شبکه مانند مرکزیت درجه‌ای، میانی یا نزدیکی را در نظر بگیریم، احتمالاً برخی افراد در شبکه نادیده خواهند ماند و اگر معلوم شود که آنها به برخی از افراد مهم و متنفذ متصل هستند، چه بسا از این بابت، آنها خودشان هم واقعاً افراد تأثیرگذار و نافذ باشند. این در حالی است که اگر این افرادِ نامشهود، بیشتر در سایه باقی بمانند، می‌توانند از قدرت افراد صاحب نفوذ و مهمی که با آنها ارتباط دارند استفاده کرده و به برنامه‌های خود دست یابند.

برای محاسبه مرکزیت بردار ویژه، باید مؤلفه‌های ویژه[1] ماتریس مجاورت زوجی گراف را محاسبه کرده، سپس بردار ویژه مرتبط با بیشترین مقدار ویژه انتخاب می‌شود. عنصر i در بردار ویژه، مرکزیت گره iام را مشخص می‌کند (شکل ۸–۱).

[1] Eigencomposition



شکل ۸-۱ مرکزیت برداروِیژه گره‌ها

۱-۳-٦ رتبه-صفحه

رتبه-صفحه ، که نوعی معیار مرکزیت بردار ویژه است، بر اساس اینکه یک کاربر با چه احتمالی از طریق ابرلینکها[1] صفحات وب را مشاهده می‌کند، اهمیت این صفحات را تعیین می‌کند. این الگوریتم یکی از الگوریتمهای محبوب "رتبه‌بندی مبتنی بر پیوند"[2] است که اشیا را در یک شبکه با توجه به اهمیت نسبی هر صفحه در بین مجموعه اشیا[3]، رتبه‌بندی می‌کند و برای رسیدن به این هدف، اطلاعات مربوط به لینکها را بکار می‌گیرد. این الگوریتم به‌طور موفقیت‌آمیزی در زمینهٔ جستجوی وب مورد استفاده قرار گرفته است. با توجه به الگوریتم رتبه-صفحه، اهمیت هر صفحه از طریق اهمیت صفحاتی که با آن لینک هستند تعیین می‌شود. اگر مجموع امتیازات رتبهٔ لینکهای بازگشتی یا بک-لینکهای[4] یک صفحه، بالا باشد، رتبه آن صفحه بالا می‌شود. در غیر این صورت، رتبه آن صفحه پایین تعیین می‌شود.

برای یک شبکه جهت‌دار G با ماتریس مجاورت A، امتیاز رتبه-صفحه یک صفحه u به صورت تکراری با امتیازات همسایه‌های ورودیش تعیین می‌شود.

$$PR(u) = \frac{1-\alpha}{N} + \alpha \sum_{v} A_{vu} \, PR(u) \Big/ d_{out}(v) \qquad ۵-۱$$

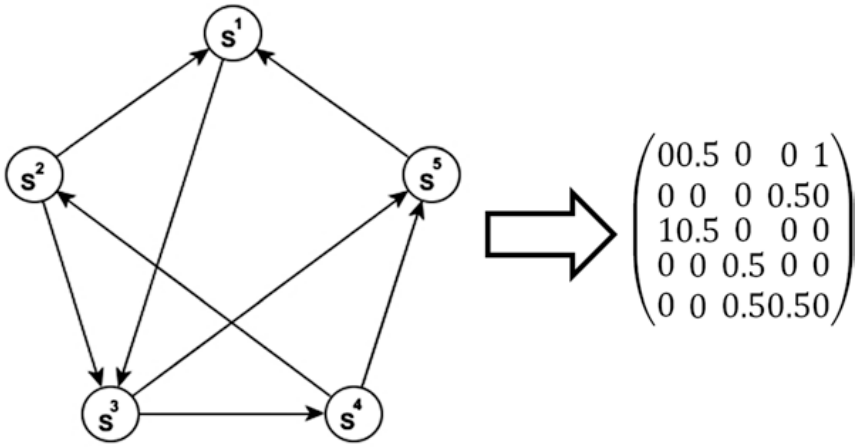
[1] Hyperlinks
[2] Link-Based Ranking
[3] Object Set
[4] Backlinks

در رابطه ۵-۱، پارامتر α، یک ضریب میرایی (یا فاکتور تعدیل[1]) است و مقداری بین صفر و یک دارد، N تعداد کل گره‌های شبکه هست و $d_{out}(v)$، نیز یالهای خروجی از گره v می‌باشد.

در تعیین رتبه-صفحه : گره‌ها عملاً صفحات وب هستند، یالها نیز لینکهای وب بوده و حالت[2]، اهمیت موقتی گره است. این معیار با تخصیص اجباری وزن غیر صفر مثبت به همه جفت گره‌ها محاسبه می‌شود که این کار کل شبکه را شدیداً همبند می‌کند. ماتریس ضرایب آن عملاً یک ماتریس احتمال انتقال یا احتمال گذار[3] (TPM) است، که می‌تواند از تقسیم هر ستون از ماتریس مجاورت، بر اولین عدد در آن ستون، به دست آید (شکل۹-۱).

محاسبه: همواره تنها یک مقدار ویژه غالب[4] برای ماتریس TPM در یک شبکهٔ به‌شدت همبند، با $\lambda = 1$ وجود دارد. این امر، توزیع متعادل جمعیت را بر روی وب WWW نشان می‌دهد.

با حل $x = Ax$، رتبه-صفحه را برای تمام صفحات وب در شبکه جهانی اینترنت WWW به دست خواهیم آورد.



شکل ۹-۱ تبدیل رتبه-صفحه به ماتریس

[1] Damping Factor
[2] State
[3] Transition Probability Matrix (TPM)
[4] Dominant Eigenvector

الگوریتم رتبه–صفحه، نتایج نابدیهی[1] (ناصفر یا نامعلوم) برای شبکه‌های نامتقارن می‌دهد. اگر پیوندها متقارن (بدون جهت) باشند، مقادیر رتبه–صفحه همانند درجه‌های گره هستند.

فرمول اولیهٔ محاسبه رتبه–صفحه، قابل سوءاستفاده و از این نظر آسیب‌پذیر است و بنابراین باید به‌طور منظم و به‌طور پنهانی تنظیم شود.

گرچه رتبه–صفحه برای فهرست صفحات وب پیشنهاد شده، با این‌حال، می‌توان از آن برای محاسبه مرکزیت گرافهای جهت‌دار در شبکه‌های اجتماعی استفاده کرد. توییتر می‌تواند مثال خوبی برای این منظور باشد!

۱-۳-۷ فاصله کروی (ژئودزیک) و کوتاهترین مسیر

برای گرافهای جهت‌دار، فاصله ژئودزیک d بین دو گره (u و v) به‌صورت تعداد یالهای بین این دو گره تعریف می‌شود، درحالی‌که "کوتاهترین مسیر بین دو گره" به‌صورت مسیری از u به v با حداقل تعداد یالها تعریف می‌شود.

برای گرافهای بدون جهت، فاصله d از u به v همان فاصله از v به u است (یعنی $d(u,v) = d(v,u)$)؛ زیرا تمام مسیرها می‌توانند معکوس شوند.

همچنین فاصله یک گره از خودش همیشه صفر است ($d(u,v) = 0$) و فاصله بین دو گره مجاور یک است (یعنی فقط یک یال میان u و v وجود دارد) و اگر مسیری بین دو گره u و v وجود نداشته باشد، $d(u,v) = \infty$ است.

بسیاری از اندازه‌گیریهای مهم شبکه بر اساس مفهوم فاصله ژئودزیک و کوتاهترین مسیر هستند. در شبکه‌های اجتماعی واقعی فاصله بین گره‌ها می‌تواند کوچک باشد درحالیکه تعداد یالها می‌تواند بسیار زیاد باشد.

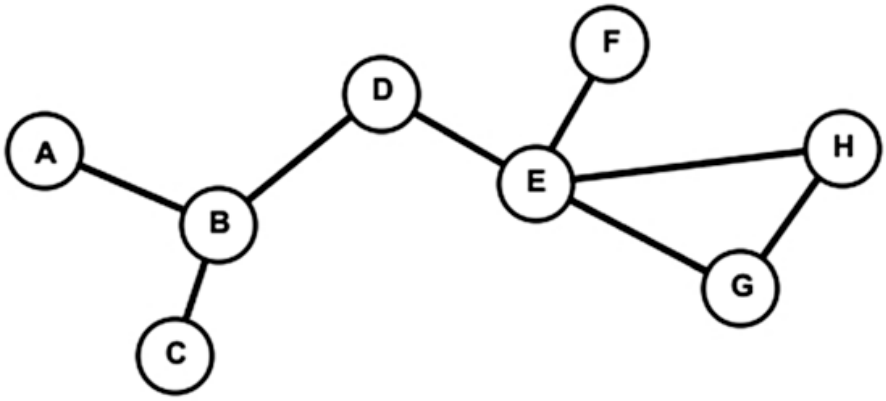
[1] Nontrivial

۱-۳-۸ گریز از مرکز

گریز از مرکز (یا برون‌مرکزی) حداکثر فاصله از یک گره معین به‌تمامی گره‌های دیگر در یک شبکه است. برخلاف سنجشهای مربوط به کل شبکه (مثلاً، قطر و چگالی) که مقداری برای همه شبکه (یا شبکه در کل) هستند، سنجش برون‌مرکزی یک معیار محبوب مبتنی–بر–مسیر است که به هر گره در یک شبکه بر اساس ارتباطات جهت‌دار و بی‌جهت آن گره، ارزش می‌دهد. اگر حداکثر مقدار گریز از مرکز را محاسبه کنیم، که عملاً حداکثر فاصله بین گره‌ها است، آنگاه نتیجه‌ای بدست می‌آید که به آن «قطر» می‌گویند. علاوه بر برون‌مرکزی، معیارهای معروف مبتنی–بر–مسیر دیگری همانند ضریب خوشه‌گی گره، مرکزیت نزدیکی و بینابینی وجود دارد.

به‌طور رسمی، گریز از مرکز E از یک گره u (که $E(u)$ نامیده می‌شود) بعنوان حداکثر فاصله بین u و سایر گره‌ها در شبکه تعریف می‌شود و این بدان معنی است که ما باید فاصله ژئودزیک را از u به تمام گره‌های دیگر محاسبه کرده و حداکثر مقدار را انتخاب کنیم.

$$E(u) = \max_{v \in V} d(u.v) \qquad ۶-۱$$



شکل ۱۰-۱ گراف ساده با هشت گره و هشت یال

محاسبه گریز از مرکز برای درک چگونگی نشر اطلاعات از یک گره به دیگر گره‌ها مهم است. اگر اطلاعات از یک گره مشخص به سمت برخی گره‌های خاص در گراف در تعداد گامها یا مراحل کمتری منتشر شوند، به این معنی است که این گره موقعیت انتشار بهتری دارد، اگرچه اطلاعات همیشه نباید از کوتاهترین مسیرهای شبکه عبور کنند (شکل ۱۰-۱).

شکل ۱۰-۱ یک گراف با هشت گره و هشت یال است. همانگونه که شکل نشان می‌دهد حداکثر مقدار گریز از مرکز (که همانا قطر شبکه است) ۴ می‌باشد. همچنین می‌توان دید که معیار برون‌مرکزی برای گره D عدد ۲ می‌باشد؛ که به این معنی است که دورترین گره‌ها از گره D در فاصله ۲ پرش یا گام، واقع شده‌اند. یعنی اطلاعات منتشرشده توسط گره D می‌تواند از میان گره‌های E و B (در کمترین تعداد مراحل) برای رسیدن به دورترین نقاط در گراف عبور کند. با این حال، برای مثال گره A دارای مقدار گریز از مرکز ۴ است که به این معنی است که اطلاعات باید حداقل از سه گره (B، D وE) و چهار یال عبور کند تا به دورترین نقاط برسد.

۱-۳-۹ چگالی

چگالی عملاً درجه یا میزانی است که گره‌های شبکه به یکدیگر متصل شده‌اند. از چگالی می‌توان بعنوان معیاری برای سنجش میزان نزدیک بودن یک شبکه به "شبکه کامل" استفاده کرد. چگالی یک شبکه یا گراف کامل (گرافی که در آن تمام یالهای ممکن، وجود داشته باشند) برابر با یک است. در زندگی واقعی، یک گروه متراکم یا چگال، گروهی از اشیا است که ارتباطات زیادی در میان موجودیتهای خود دارد (یعنی چگالی بالایی دارد)، در حالیکه یک گروه تُنُک یا پراکنده، ارتباطات کمتری را دارد (یعنی چگالی پایینی دارد).

به‌طور رسمی، چگالی $D(G)$ برای گراف G بعنوان نسبت تعداد یالهای موجود در G بر تعداد تمام یالهای ممکن تعریف می‌شود. مقادیر چگالی بین صفر و یک متغیر است.

برای گرافهای بدون جهت، تعداد یالهای ممکن، $n(n-1)/2$، است؛ بنابراین،

$$D(G) = \frac{2 \times m}{n(n-1)} \qquad ۷-۱$$

در رابطه ۷-۱، مقدار $m=|E|$، بوده یعنی برابر تعداد یالهای گراف G می‌باشد. اگر $D(G) = 1$ باشد، آنگاه گراف کامل است و هر جفت گره توسط یک یال به هم متصل هستند. اگر $D(G) = 0$ باشد، به این معنی است که گراف، هیچ یالی ندارد و همه گره‌ها مجزا هستند. با توجه به اینکه که هر گراف همبند[۱]، باید حداقل $n - 1$ یال داشته باشد، حداقل چگالی برای یک گراف همبند، $n/۲$ است.

اگر به محاسبهٔ چگالی شبکه‌های فردانه (یا شبکهٔ فردمحور)[۲] علاقه‌مند باشید، باید لینکهای اطراف گره موردنظر (گره فردانه) را در نظر بگیرید. هدف چنین تحلیلی، بررسی تأثیر گره موردنظر بر همسایگانش در زیرگروهی است که این گره به آن تعلق دارد. ولیکن اگر می‌خواهیم چگالی شبکه‌های جمعانه (یا جمع-محور)[۳] را محاسبه کنیم، باید گراف کامل و محدودیتهای شبکه را نیز در نظر بگیریم. شبکه کاملاً همبند (که در آن هر گره به تمام گره‌های دیگر در شبکه متصل است) دارای چگالی ۱ است، درحالیکه تعداد یالهای ممکن برای هر گراف $n(n-1)/2$ است.

شکل ۱۱-۱ یک گراف با $n=5$ گره را در سه حالت (یعنی با تعداد رو به افزایشی از یالها در هر حالت) نشان می‌دهد.

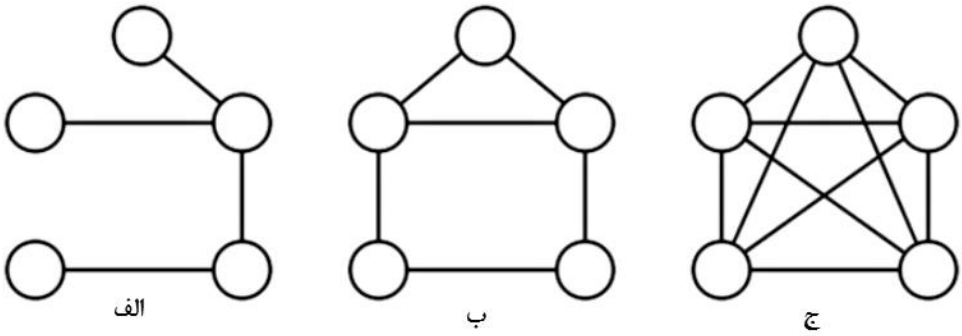
---

[1] Connected Graph

[2] Ego-centric Networks

منظور از Ego-centric Network شبکه یا زیرشبکه‌ای است که با محوریت Ego یعنی "من" ، "فرد" یا "خود" که از درون شبکه بزرگتر استخراج شده است. مرکز یا کانون چنین زیرشبکه‌ای، تنها یک گره خاص (یا گره کانونی) است. چنین زیرشبکه‌ای را با توجه به محوریت یک گره باید خود -محور یا من-محور نامید. نظر به اینکه در زبان فارسی این دو واژه نوعی رفتار اخلاقی تلقی می شود ما واژه "فردانه" یا "فرد-محور" را برای آن پیشنهاد داده‌ایم. برای حالتیکه هدف، بررسی کل شبکه باشد یعنی Socio-centric Networks ما واژه "جمعانه" یا "جمع-محور" را پیشنهاد داده‌ایم. مبحث اصلی در این زمینه، در فصل پنج ارائه شده است.

[3] Socio-centric Networks



شکل ۱۱-۱ گراف ساده با پنج گره و سه حالت

در قسمت الف، گراف دارای چهار یال است، بنابراین چگالی آن برابر با *D(G) = (2*4)/(5*4) = 0.4*، است. در گراف ب، چگالی برابر با *D(G) = (2*6)/(5*4) = 0.6*، و در نهایت چگالی قسمت ج برابر با *D(G) = (2*10)/(5*4)=1* است.

در بسیاری از شبکه‌های اجتماعی آنلاین (بعنوان مثال، فیسبوک)، چگالی معمولاً پایین است. این بدان دلیل است که هر کاربر معمولاً به دهها یا صدها کاربر دیگر متصل می‌شود درحالی‌که خود شبکه، تقریباً یک میلیارد کاربر دارد.

۱-٤ بخش‌بندی شبکه

معیارهای بخش‌بندی شبکه، مناطقی از شبکه را شناسایی می‌کنند که شبکه در آن مناطق متراکم‌تر، چگالتر و منسجم‌تر از دیگر نواحی هستند.

۱-٤-۱ گروههای منسجم

گروه منسجم، یک اجتماع در درون شبکه است که ارتباط داخلی گره‌ها بیشتر از ارتباط خارجی است.

گروه منسجم[1]، یک اجتماع یا انجمن در درون یک شبکه است که گره‌ها در آن گروه (یعنی اعضای گروه) با همدیگر (یعنی با اعضای داخل گروه) ارتباط بیشتری نسبت به اعضای خارج گروه دارند؛ چنین وضعیتی حاکی از آنست که همه اعضای آن گروه به یکدیگر دسترسی دارند.

در میان اعضای گروههای بسیار منسجم، معمولاً باورهای عمیقاً مشابهی وجود دارد (به بیان دیگر، افرادی با اعتقادات مشابه، پتانسیل خوبی برای تشکیل چنین اجتماعاتی را دارند). ارتباطات بین اعضای اجتماع می‌تواند از طریق تماسهای شخصی (یعنی مستقیم) یا هم–عضوی[2] (یعنی غیرمستقیم) شکل بگیرد. در چنین وضعیتی، هر چه افراد به یک اجتماع بیشتر گره خورده باشند، آنگاه بیشتر تحت‌تاثیر استانداردهای گروهی قرار می‌گیرند.

پیدا کردن مناطقی از یک شبکه که در آن مناطق گره‌ها به یکدیگر محکم‌تر از دیگر نواحی، پیوند خورده‌اند، "آنالیز انسجام گروهی"[3] نامیده می‌شوند.

یکی از نمونه گروههای منسجم که در نظریه گراف رایج است (و ریشه در تحلیل شبکه اجتماعی دارد) "محفل" نامیده می‌شود که یک زیرگراف کامل است. مثال دیگری از این‌دست، "سازه k–هسته"[4] است که تعریف آن در شبکه‌های بدون جهت، بدین صورت است: k–هسته یک زیرگراف است با درجه‌ای مساوی یا بزرگ‌تر از k. لازم به ذکر است که سازه‌های k–هسته‌ای الزاماً یک گروه منسجم نیستند، بلکه قسمتهایی از یک گراف را نشان می‌دهند که دارای "ساختارهای محفل–مانند" می‌باشند.

۱-٤-۲ محفل

محفل، یک گراف (یا زیرگراف) است که در آن هر گره به همه گره‌های دیگر متصل است. در ادبیات علوم اجتماعی، محفل یک گروه اجتماعی است که در آن، همه افراد یکدیگر را می‌شناسند (یعنی بین هر جفت گره، یالی وجود دارد). یک مثلث، مثالی از محفل با اندازه سه است چون دارای سه گره بوده و همه گره‌ها متصلند.

---

[1] Cohesive Groups

[2] در این کتاب بعنوان معادل واژه Co-Membership که به معنای عضویت دو نفر یا بیشتر در یک انجمن یا نهاد است از ترکیب "هم–عضوی" استفاده شده است. مبحث هم–عضوی و یا کلاً، هم–نهادی در فصل هفت تشریح شده است.

[3] Cohesive Group Analysis

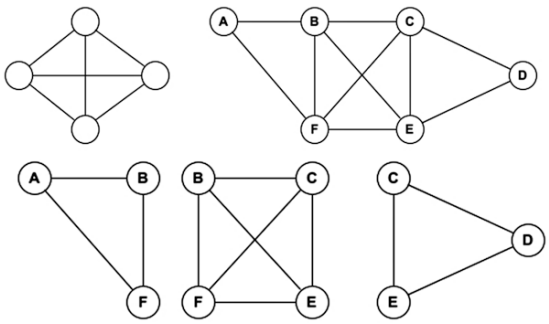
[4] K-Core Construct

یک محفل بیشین[1]، محفلی است که زیرمجموعه هیچ محفل دیگری در گراف نیست. یک محفل با اندازه‌ای برابر یا بزرگتر از سایر محافل دیگر در یک گراف را یک محفل بیشینه[2] نامند (شکل ۱۲-۱).

محفل بیشینه (بزرگترین محفل) در پیکره‌بندی گراف شکل ۱۲-۱، دارای اندازه چهار است. با این حال، تعدادی محفل بیشین به اندازه سه هم وجود دارند.

پیدا کردن محافل بیشینه در شبکه‌ها، یکی از مسائل بسیار جالب و کاربردی در آنالیز روابط اجتماعی است چرا که این کار امکان یافتن بزرگترین مجموعه دوستیهای متداول را فراهم می‌کند. با مقایسهٔ اندازهٔ محافل دوستی بیشینهٔ مربوط به دو گروه اجتماعی، می‌توان اطلاعات خوبی در مورد برخی از جنبه‌های پویایی آن گروهها مانند کار تیمی، اعتماد و بهره‌وری آنان به دست آوریم.

یکی از سؤالات متداول در شبکه‌های دوستی آن است که کدام یک از دوستان شما نیز با هم دوست هستند؟ چنین سؤالی که در مورد توانایی رسیدن به دوستیهای متقابل در شبکه اجتماعی (مثلاً فیسبوک) است همواره وجود دارد. پاسخ به این سؤال از طریق شناسایی محافل و یا دوستی متقابل در یک گراف فردانه (یا فرد-محور)[3] امکانپذیر است.



شکل ۱۲-۱ گراف با ساختارهای متفاوت

[1] Maximal Clique
[2] Maximum Clique
[3] Ego Graph

۱-٤-۳ k-هسته

در گرافهای بدون جهت، یک k-هسته‌ای، یک زیرگرافِ منتجهٔ بیشینِ همبند[1] است که مقدار بیشینه آن بزرگتر یا مساوی k است. این بدان معنی است که هر گره، با حداقل k گره دیگر ارتباط دارد. این زیرگرافها لزوماً گروه منسجم نیستند (و یا انسجام گروهی[2] ندارند)، اما می‌توانند مناطقی را مشخص کنند که دارای ساختارهای محفل-مانند هستند. تعیین k-هسته‌ایها را می‌توان بعنوان راهی برای یافتن زیرگروههای منسجم در گرافها به کار برد.

محققان از مفهوم k-هسته‌ای بعنوان یک روش نمونه‌گیری برای جمع‌آوری داده‌ها از شبکه‌های اجتماعی استفاده می‌کنند. در این راستا، این روش به‌ویژه آنگاه مؤثر است که محققی علاقه‌مند به استخراج گروه نسبتاً منسجمی از شرکت‌کنندگان (از میان یک گروه بزرگ اما پراکنده از شرکت‌کنندگان) در یک رویداد باشد. در اینصورت، مثلاً، اعضای متعلق به k-هسته‌ای باید در یک دوره مشخص در یک فعالیت معین مشارکت فعال داشته باشند، درحالیکه کسانی که به k-هسته‌ای تعلق ندارند، عضو ضعیف یا کم‌کار[3] هستند.

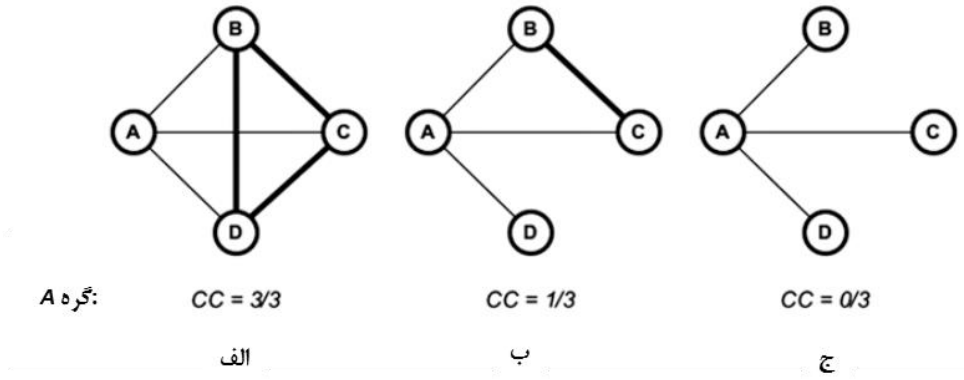
۱-٤-٤ ضریب خوشه‌گی

ضریب خوشه‌گی معیاری است که نشان می‌دهد تا چه حد گروهها، تمایل دارند تا زیرگرافهای چگال یا متراکم در یک شبکه تشکیل دهند. این زیرگرافهای چگال را بسته به کاربرد آنها، به نامهای محفل، اجتماع (انجمن) و یا خوشه می‌شناسند. در شبکه‌های اجتماعی، این مسئله را می‌توان به این صورت تفسیر کرد که: احتمال اینکه دو دوستِ یک فرد با هم دوست باشند تا چه حد است. به‌این‌ترتیب، مقدار ضریب خوشه‌گی عددی بین صفر و یک است.

[1] Connected Maximal Induced Subgraph
[2] Cohesive Groups
[3] Light Members

مقادیر بالای ضریب خوشه‌گی برای یک شبکه اجتماعی بدین معناست که این شبکه عملاً "پدیده جهان کوچک" را نشان می‌دهد (در شبکه‌های جهان کوچک، اغلب گره‌ها همگن بوده و با چند قدم یا چند یال می‌توان به هر گره‌ای دسترسی داشت).

معمولاً در شبکه‌های اجتماعی واقعی، ضریب خوشه‌گی بالاتری را نسبت به شبکه‌های تصادفی متناظرشان شاهد هستیم که به معنی تمایل بیشتر برای تشکیل مثلث است.



شکل ۱۳-۱ ضریب خوشه‌گی یک گراف با چهار گره (نگاه از منظر گره A)

به این ترتیب، با استفاده از ضریب خوشه‌گی می‌توان تمایزی بین شبکه‌های اجتماعی واقعی و تصادفی قائل شد و اینکار از طریق برجسته کردن گره‌هایی است که گره‌های مجاورشان به‌خوبی به هم متصل شده‌اند. از این معیار اغلب، بعنوان معیاری برای تراگذری یک گراف یاد می‌شود (تراگذری را گاهی ضریب خوشه‌گی کل شبکه می‌نامند) (شکل ۱۳-۱).

شکل ۱۳-۱ الف، نشاندهنده یک شبکه فردانه (فردمحور) فشرده یا چگال است (که بعنوان یک محفل نیز شناخته می‌شود). در واقع ضریب خوشه‌گی آن بالا و اعتماد متقابلش[1] نیز بالا است. شکل ۱۳-۱ ب از طرف دیگر، دارای یک گره مخابره کنندهٔ تنها[2] و چندین گره شنونده منفعل[3] است. این شبکه، ضریب خوشه‌گی پایینی دارد.

[1] Mutual Trust
[2] Single Broadcast Node
[3] Passive Listeners

یک ضریب خوشه‌گی به مقدار C برای یک گره i نشان می‌دهد که چگونه همسایگان آن گره باهم ارتباط دارند. به‌طور رسمی، این ضریب، نسبت بین تعداد کل پیوندهای ارتباط‌دهنده گره با همسایگانش به تعداد کل پیوندهای ممکن است. ضریب خوشه‌گی موضعی یا محلی (گره) به‌صورت رابطه ۸–۱، محاسبه می‌شود:

$$C_i = \binom{k_i}{2}^{-1} T(i) = \frac{2 \times T(i)}{k_i(k_i - 1)} \qquad ۸–۱$$

در رابطه بالا، $T(i)$ تعداد مثلث با گره i و $k_i(k_i - 1)$، حداکثر تعداد اتصالات ممکن در همسایگان گره i می‌باشد. یک C بزرگ نشان می‌دهد که شبکه به‌طور موضعی یا محلی متصل است و می‌تواند یک خوشه تشکیل دهد. میانگین ضریب خوشه‌گی به‌صورت رابطه ۹–۱ تعریف می‌شود:

$$C = \frac{1}{n} \sum_{i \in V} C_i \qquad ۹–۱$$

محاسبه ضریب خوشه‌گی گره C_i، در بدترین حالت ممکن (یعنی از طریق شمارش همه یالهای متصل به گره i) در زمان $O(n^2)$ قابل انجام است. ضریب خوشه‌گی شبکه را می‌توان با شمارش تمام سه‌تاییهای بسته[1] محاسبه کرد.

یک رویکرد بی‌قاعده برای تعیین ضریب خوشه‌گی شبکه، همانا بررسی ترکیب گره‌هاست که پیچیدگی زمانی آن برابر با $O(n^3)$ است. یک رویکرد دیگر، استفاده از عملیات ضرب ماتریسی سریع[2] بر روی ماتریس مجاورت گراف G است. هدف از این کار، یافتن و شمارش سه‌تاییها، و سپس شمارش گره‌هاست که در زمان $O(n^{2.376})$ و در فضای $O(n^2)$ امکانپذیر می‌باشد. در گرافهای دو–بخشی[3] که در آن مثلثها وجود ندارند، ضریب خوشه‌گی برابر با صفر است.

[1] Closed Triplets
[2] Fast Matrix Multiplication
[3] Bipartite Graphs

۱-٤-٥ هسته/حاشیه

مفهوم هسته و حاشیه بیان می‌دارد که در هر گراف جهت‌دار، گره‌ها باید به یکی از دو دسته زیر تعلق داشته باشند: (۱) **هسته** که در آن گره‌ها در یک زیرگراف منسجم قرارگرفته یا محاط شده‌اند و (۲) **حاشیه** که در آن گره‌ها در وضعیتی غیرمنسجم و پراکنده با هم اتصال دارند. به بیان دیگر، گره‌های هسته، بسیار بیشتر به هم متصل هستند تا گره‌های محیط اطراف یا پیرامون.

اگر ساختار k–هسته را در نظر بگیریم، همچنین می‌توانیم هسته و حاشیه را در یک اندازه کوچک‌تر تعریف کنیم. گره‌هایی که بخشی از ساختار k–هسته هستند، عملاً گره‌های هسته هستند، درحالی‌که گره‌های خارج از آن (یعنی گره‌های کم تحرک یا ضعیف) گره‌های حاشیه‌ای هستند.

شبکه‌ها به‌طورکلی، نوعی تغییرات ساختاری که شامل اضافه یا حذف گره‌ها و یا یالهاست را در طول زمان از خود نشان می‌دهند. بعنوان مثال، بعضاً شاهد آنیم که روابط اجتماعی با توجه به رویدادهای اجتماعی خاص در یک جامعه پدیدار و یا ناپدید می‌شوند. عوامل متعددی می‌توانند بر مسیر این تغییرات تأثیر بگذارند مانند ویژگیهای فردی و رفتار کنشگران و شباهت خصوصیات جفت گره‌ها.

مطالعه رفتار دینامیکی شبکه‌ها که نشان می‌دهد چگونه آنها در طول زمان تکامل می‌یابند، تحلیل شبکه پویا[1] (DNA) نامیده می‌شود و با تحلیل شبکه اجتماعی[2] (SNA) سنتی که فرض می‌کند شبکه موردنظر در طول زمان تحلیل، ایستاست، تفاوت دارد. درحالی‌که DNA، بیشتر به فعالیت فعالان شبکه و تعاملات آنها مربوط است، SNA، بر خواص ساختاری یک شبکه تمرکز می‌کند.

در شبکه‌های در حال تکامل، گره‌ها ممکن است همیشه در یک حالت باقی نمانند و از یک کلاس شبکه به کلاس دیگر حرکت کنند. برای مثال، گره‌های هسته، بسته به رفتارهایی که گره‌ها در شبکه نشان می‌دهند و همچنین رفتار شبکه بعنوان یک کل، می‌توانند گره‌های بلندمدت[3] هسته بوده (برای مدت طولانی در هسته بمانند) یا گره‌های کوتاه‌مدت هسته باشند (بعد از مدت کوتاهی که در هسته قرار گرفتند، یا به سمت حاشیه حرکت کنند یا کلاً از بین بروند).

[1] Dynamic Network Analysis
[2] Social Network Analysis
[3] Long-Term Core Nodes

در حوزه وبلاگ نویسی، نشانه‌های هسته و حاشیه می‌تواند به این صورت تفسیر شود که وبلاگهای هسته، آن وبلاگهایی هستند که بیشتر مورد مطالعه قرار گرفته، بیشتر مورد استناد رسانه‌ها بوده و بیشتر با سایر وبلاگها مرتبط هستند، اما وبلاگهای حاشیه‌ای، ارتباطات ضعیفی با دیگر بلاگها در این حوزه دارند. در شبکه‌های اجتماعی آنلاین، فردی که ارتباطات زیادی دارد (یعنی دوستان زیادی دارد) در هسته شبکه اجتماعی محلی می‌ماند و از مزایای این موقعیت منحصربه‌فرد سود می‌برد. شاید بشود گفت افرادی که در هسته هستند در مقایسه با افراد دیگری که در حاشیه قرار دارند، از سلامت اجتماعی بیشتری برخوردارند.

۱-٤-٦ مدلهای بلوک

مدل بلوکه‌ای یک روش تحلیلی است که از بخش‌بندی داده‌ها در یک شبکه اجتماعی استفاده کرده و کنشگران را بر اساس الگوهای ارتباطاتشان با دیگران، طبقه‌بندی می‌کند. این شیوه یکی از اولین روشهای تشخیص گروههای اجتماعی (یعنی اجتماعات یا انجمنها) است و بعنوان یک روش انقلابی در مستندسازی الگوهای ارتباط و شناسایی موقعیتهای اجتماعی مورد توجه قرار گرفت.

ایده کلی مدل بلوکه‌ای این است که گره‌ها (بر اساس برخی معیارهای هم‌ارز[۱]) به موقعیتهای زیرگروه گسسته (یعنی بلوکها)، بخش‌بندی می‌شوند. سپس هر جفت موقعیت (هر دو بلوک)، به منظور درک وجود یا عدم وجود یالهای ارتباطی بررسی می‌شود. از معیارهای هم‌ارزی متعددی از جمله هم ارزی ساختاری[۲]، هم ارزی منظم[۳] و هم ارزی تصادفی[۴] می‌توان برای این هدف بهره برد.

اولین مطالعه که از مدلهای بلوک برای کشف اجتماعات در شبکه‌های اجتماعی استفاده کرد، در سال ۱۹۴۱ انجام شد. داده‌های متقارن از فعالیتهای اجتماعی ۱۸ نفر زن فعال، در طی یک دوره ۹ ماهه جمع‌آوری شد که در خلال آن مدت، زیرمجموعه‌های مختلف این زنان، ۱۴ رویداد اجتماعی را محقق کرده بودند.

[1] Equivalence Measure
[2] Structural Equivalence
[3] Regular Equivalence
[4] Stochastic Equivalence

با استفاده از مدلهای بلوک می‌توان افرادی را شناسایی کرد که آنها قادرند نقش مشابهی را در یک جمعیت مشابه و یا در یک جمعیت متفاوت ایفا کنند. به همین دلیل از این مدلها (و به کمک داده‌های روابط اجتماعی) برای شناسایی ساختارهای نقش[۱] در میان مدیران کسب‌وکارها بهره برده شده است. از این مدلها همچنین می‌توان برای درک ساختار شبکه‌های هم–استنادی[۲] و ساختار گونه‌های شخصیتی در گروههای کوچک استفاده کرد.

۱-٤-۷ خوشه‌بندی سلسله مراتبی

خوشه‌بندی سلسله‌مراتبی[۳] یکی از روشهای اولیه برای خوشه‌بندی داده‌ها با استفاده از ماتریس تشابه است. نام این روش، در واقع عنوان یک دسته از مدلها در مقابل دسته دیگری از مدلها برای الگوریتمهای خوشه‌بندی است؛ یعنی: مدلهای سلسله‌مراتبی (بعنوان مثال، الگوریتم همجوشی) و مدلهای غیر سلسله‌مراتبی (بعنوان مثال، الگوریتم *k-means*). اعضای دسته اول، یک سلسله‌مراتب کامل از خوشه‌ها ایجاد می‌کنند، درحالی‌که اعضای دسته دوم، تعداد مشخصی از خوشه‌های داده را فراهم می‌کنند. در شبکه‌های اجتماعی، خوشه‌بندی سلسله‌مراتبی به این معنی است که ما گره‌های شبکه (کنشگران) را به گروههایی از گره‌های مشابه با استفاده از معیارهای فاصله‌ای بخش‌بندی می‌کنیم.

دو الگوریتم برای مدل خوشه‌بندی سلسله‌مراتبی توسعه داده شده است:

الگوریتم همجوشی[۴] با رویه پایین به بالا

الگوریتم جدایشی[۵] با رویه بالا به پایین

---

[1] Role Structures
[2] Co-Citation Networks
[3] Hierarchical Clustering
[4] Agglomerative
برای این کلمه، معادلهایی همچون همجوشی، ترکیبی، ترکیب‌گرایانه، پیوندی، ادغامی، تجمیعی، تلفیقی و یا امثال آن قابل استفاده است. ما برای داشتن یک معادل متمایز از واژه "همجوشی" استفاده می‌کنیم.
[5] Divisive

الگوریتم همجوشی (با رویه پایین به بالا) با خوشه‌بندی همه آیتمهای داده به خوشه‌ها (گروه‌ها) بر اساس یک سنجش فاصله‌ای شروع می‌شود، یعنی ابتدائاً هر آیتم را عملاً یک خوشه فرض می‌کند. سپس نزدیکترین جفت خوشه‌ها (در مسیر رفتن به سطح بعدی سلسله‌مراتب) برای شکل‌دهی یک خوشه واحد، به هم می‌پیوندند (و یا ملحق می‌شوند) و همانطور که ما از پایین (برگ) به بالا (ریشه) حرکت می‌کنیم، همبستگی بین اقلام داده‌ای خوشه‌های مختلف کاهش می‌یابد. این مراحل الحاق تا آنجا تکرار می‌شود که همه آیتمها به یک خوشه واحد خوشه‌بندی شوند.

از این الگوریتم معمولاً توسط گروههای اجتماعی برای انجمن‌یابی (یا شناسایی اجتماعات) استفاده می‌شود. با استفاده از این الگوریتم، ما می‌توانیم اندازه سنجش شباهت x_{ij} را بین یک جفت گره (i , j) بر اساس ساختار شبکه موردنظر تعریف کنیم. ما با یک جفت داده با بالاترین امتیاز شباهت (حداقل فاصله) شروع می‌کنیم و سپس جفتهای بیشتری اضافه می‌کنیم که شباهت را کاهش می‌دهد. تعیین تعداد مطلوب خوشه‌های (یا گروههای) موردنظر کاربران توسط تحلیلگر مشخص می‌شود.

انتخاب تعداد خوشه‌ها می‌تواند به روشهای مختلفی از قبیل محاسبه خطای طبقه‌بندی انجام شود؛ این روش متمرکز بر زانوی منحنی منتجه براساس تعداد خوشه‌هاست. (یعنی با افزایش تعداد خوشه‌ها، این خطا کاهش می‌یابد تا جاییکه منحنی خطا، کاهش بیشتری ندارد. شکل ظاهری این منحنی، باعث استفاده از واژه زانو یا آرنج برای آن شده است).

همچنین می‌توان از روشهای مختلفی برای یافتن خوشه‌ای که بهترین نامزد ادغام است، استفاده کرد. بعنوان مثال، روش پیوند–کامل[1]، فاصله بین دو خوشه را برابر با بزرگترین "فاصله پیوندی" می‌داند. فاصله پیوندی برابر است با فاصله هر آیتمِ داده در یک خوشه با هر آیتمِ داده در خوشه دیگر. در حالیکه روش پیوند–تکی[2] فاصله بین دو خوشه را برابر با کوتاهترین "فاصله پیوندی" می‌داند: که بازهم فاصله پیوندی برابر است فاصله یک آیتم داده در خوشه اول و هر آیتم داده در خوشه دیگر. روش پیوند–

برای این کلمه، معادلهایی همچون جدایشی، تجزیه‌ای، تجزیه‌گرایانه، انفصالی، تفکیکی، افرازی، جداگرایانه، و یا امثال آن قابل استفاده است که ما معمولاً از واژه "جدایشی" را استفاده کرده‌ایم.

[1] Complete-Linkage Method
[2] Single-Linkage Method

متوسط[1]، فاصله بین دو خوشه را برابر با میانگین "فاصله پیوندی" می‌سنجد. اگرچه روش خوشه‌بندی پیوند-تکی از "اثر زنجیره‌ای"[2] رنج می‌برد و کمتر برای تشخیص خوشه‌های کروی[3] مناسب است، اما هنوز هم معمولاً از آن استفاده می‌شود زیرا توانایی تشخیص خوشه‌های طولانی[4] و نامنظم[5] را دارد. نکته دیگر آنکه از روش خوشه‌بندی پیوند-کامل خیلی استفاده نمی‌شود زیرا به متغیرها بسیار حساس است.

الگوریتم جدایشی با رویه بالا به پایین، عملاً الگوریتمی معکوس است یعنی با یک خوشهٔ تکی شروع می‌شود و آن را بعداً به قطعات کوچکتر تقسیم می‌کند. از الگوریتم همجوشی (رویه پایین به بالا) نسبت به الگوریتم جدایشی (رویه بالا به پایین) بیشتر استفاده می‌شود.

۱-۵ تحولات اخیر در حوزه تحلیل شبکه

موضوعات متعددی در حوزه تحلیل شبکه وجود دارد که امروزه به دلیل اهمیت این موضوعات در حوزه‌های کاربردی، محققان زیادی به آنها علاقه‌مند شده‌اند. در اینجا در باره چهار مورد از آنها بحث می‌کنیم: شناسایی اجتماعات یا انجمن یابی[6]، پیش‌بینی پیوند[7]، شبکه‌های تعامل پروتئین به پروتئین[8] و سیستمهای توصیه‌گر[9].

[1] Average-Linkage Clustering Method
[2] Chain Effect
[3] Spherical Clusters
[4] Elongated Clusters
[5] Irregular Clusters
[6] Community Detection
[7] Link Prediction
[8] Protein-Protein Interaction Networks
[9] Recommendation Systems

۱-۵-۱ انجمن‌یابی

وظیفه شناسایی اجتماعات یا انجمن‌یابی (و یا خوشه‌بندی گراف[1]) عملاً کشف زیرمجموعه‌هایی از گره‌ها (یا خوشه‌ها) از اجتماعات همبند است که هر یک از گره‌ها در این زیرمجموعه‌ها، یالهای داخلی زیاد ولیکن یالهای خارجی کم دارند. شناسایی انجمنها در گرافها تنها در صورتی امکان‌پذیر است که گراف تُنُک یا پراکنده[2] باشد، به این معنی که تعداد ارتباطات (m) تقریباً برابر با تعداد گره‌ها (n) باشد. انجمن‌یابی دارای کاربردهای بسیاری ازجمله مدلسازی شبکه‌های آنلاین در مقیاس گسترده[3] و درک ساختار اجتماعی سازمانها است.

بسیاری از شبکه‌ها، مجموعه‌ای از گروههای متمایز[4] (یا انجمنها) را نمایش می‌دهند به‌گونه‌ای که تعداد یالهای موجود در بین این انجمنها (مابین هر دو انجمن) کم، ولیکن تعداد یالهای موجود در درون هر انجمن زیاد است. یک اجتماع یا انجمن با داشتن ارتباط بیشتر در داخل اجتماع و ارتباط کمتر با بخش باقی‌مانده شبکه، توصیف می‌شود.

تشخیص انجمنها در یک گراف، یک مسئله NP کامل[5] است. مسئله انجمن‌یابی به نحوی شبیه به خوشه‌بندی در علوم رایانه است. استخراج تمام اجتماعات از یک شبکه معین، به‌ویژه در مورد شبکه‌های بزرگ و پیچیده، کار ساده و سرراستی نیست، زیرا این امر مستلزم بررسی کل گراف است.

به‌طور رسمی، مسئله انجمن‌یابی را می‌توان به شرح زیر فرض کرد:

با توجه به گراف داده‌شده $G = (V,E)$ یک انجمن C می‌تواند بعنوان یک زیرگراف G تعریف شود به‌طوری‌که دربرگیرنده مجموعه‌ای همچون $V_C \in V$ از اشیاء مشابه هست.

انجمن‌یابی یک حوزهٔ تحقیقاتی فعال در رشته‌های مختلف مانند علوم رایانه، علوم اجتماعی، زیست‌شناسی و روانشناسی شده است. انجمن‌یابی (کشف اجتماعات) همچنین در حوزه شبکه‌های

[1] Graph clustering
[2] Sparse
[3] Modeling Large-Scale Online Networks
[4] Distinct Groups
[5] NP-Complete

اجتماعی، نقش مهمی ایفا می‌کند، زیرا به پیدا کردن بخشهایی از جمعیت کمک می‌کند که به‌خوبی با هم ارتباط دارند. هدف از انجمن‌یابی در حوزه جامعه‌شناسی، یافتن گروههای اجتماعی بر اساس شبکه‌های تعاملات اجتماعی است.

بدین منظور، الگوریتمهای بسیاری پیشنهاد شده است؛ که می‌توان آنها را به سه دسته تقسیم کرد: الگوریتمهای جدایشی (یا تجزیه‌ای)، همجوشی (یا ترکیبی) و بهینه‌ساز. برخی از روشهای محبوب در زمینه شناسایی اجتماعات عبارتند از k–محفلی، بهینه‌سازی مدولاسیون[۱] و انجمنهای پیوندی[۲] که مانند صدها الگوریتم دیگر ریشه در سیستمهای اطلاعاتی، علوم رایانه و استنتاج بیزی دارند.

در قلمرو علم کامپیوتر، شناسایی اجتماعات یا انجمن‌یابی ارتباط نزدیکی با بخش‌بندی گراف دارد و هدف آن یافتن بخشهایی در گراف است که دارای تعداد یالهای بین–بخشی[۳] کمینه (و البته یالهای درون–بخشی بیشینه است). دانشمندان جامعه‌شناسی، روشهای مختلفی برای استخراج اجتماعات ایجاد کرده‌اند که بر اساس مفهوم زیرگروههای منسجم استوار هستند. خوشه‌بندی سلسله مراتبی و ساختارهای محفل–مانند[۴] به تعریف زیرگروههای منسجم کمک می‌کنند و معمولاً توسط علوم اجتماعی برای انجمن‌یابی مورد استفاده قرار می‌گیرند.

شناسایی انجمنها در شبکه‌ها به این معنا نیست که این اجتماعات باید بخشهای مجزا یا غیرمرتبطی از گره‌ها در شبکه باشند. بلکه، از اجتماعات انتظار می‌رود اعضای مشترکی نیز داشته باشند که اصطلاحاً به آن اجتماعات همپوشان[۵] می‌گویند. لازم به ذکر است که انجمن‌یابی می‌تواند برای انجمنهای ایستا[۶] و یا انجمنهای پویا[۷] انجام شود که به مورد دوم، شناسایی زمانی (یا موقت) انجمن[۸] نیز می‌گویند و در آن، اجتماعات به مثابه خود افراد، تکامل می‌یابند.

[1] Modularity Optimization
[2] Link Communities
[3] Inter-Partition Edges
[4] Clique-Like Structures
[5] Overlapping Communities
[6] Static Communities
[7] Dynamic Communities
[8] Temporal Community Detection

معروف‌ترین الگوریتم برای شناسایی انجمنها در شبکه‌ها توسط گیروان و نیومن[۱] پیشنهاد شده است، همان افرادی که سه نوع معیار بینابینی را شناسایی کردند: بینابینی کروی یال[۲]، بینابینی گام–تصادفی یال[۳] و بینابینی جریان فعلی یال[۴]. این تحقیق، موجب افزایش علاقه و خیزش توجهات به انجمن‌یابی در علم شبکه گردید. آنها دریافتند که نیازی به تعریف تعداد گروهها از قبل و یا اعمال محدودیت در اندازه گروه‌ها نیست. بلکه، این گروهها می‌توانند از طریق تقسیمات طبیعی[۵] بین گره‌ها شناسایی شوند. آنها رویکرد جدایشی را اعمال کردند که در آن یالها به‌تدریج حذف شده و در پی حذف آنها، پیوندهایی نیز که بین انجمنها قرار دارند از شبکه حذف می‌شوند؛ این یالهای بین انجمنی در عمل بنوعی گلوگاه[۶] (تنگا) محسوب می‌شوند. روشی که توسط این دو نفر توسعه داده شد، بعداً در صدها کار دیگر و در بسیاری از رشته‌های مختلف دنبال شد.

در علم کامپیوتر، مسئله مذکور با تجزیه یک شبکه به تعدادی معین (از پیش تعیین‌شده) از "گروه–گره" حل می‌شود. این "گروه–گره‌ها" تقریباً هم‌اندازه هستند و تعداد یالهای بین دو گروه در حداقل مقدار ممکن است. در این روش، شبکه ابتدا به دو گروه تقسیم می‌شود. سپس تقسیمات بعدی انجام می‌گیرد تا تعداد مورد نیاز از گروهها به دست آید.

مسیرهای آینده انجمن‌یابی باید موضوعاتی همانند نحوه برخورد با داده‌ها در اندازه بزرگ، شبکه‌های پویا، پیچیدگی محاسباتی شبکه‌ها، از پیش نامعلوم بودن تعداد و اندازه اجتماعات و بالاخره نحوه مدیریت گره‌های متعلق به بیش از یک انجمن (انجمنهای همپوشان) را در نظر بگیرند.

[1] Girvan and Newman
[2] Geodesic Edge Betweenness
[3] Random-Walk Edge Betweenness
[4] Current-Flow Edge Betweenness
[5] Natural Divisions
[6] Bottlenecks

۱-۵-۲ پیش‌بینی پیوند

پیش‌بینی پیوند، به موضوع پیش‌بینی احتمال بروز و ظهور پیوند بین گره‌ها بر اساس ویژگیهای آنها و الگوهای کلی پیوندها در شبکه اجتماعی اشاره می‌کند. اهمیت این موضوع مربوط به شرایطی می‌شود که اطلاعاتی در مورد پیوندهای یک شبکه از قبل وجود ندارد. روشهای اولیه برای پیش‌بینی پیوند در حوزه گرافهای بدون جهت و بدون وزن توسعه یافته است و کارهای اخیر در این زمینه، به حل مسائل مربوط به شبکه‌های جهت‌دار، وزندار و در حال تکامل[۱] پرداخته است.

با اتکاء بر این ایده که پیوندها در یک شبکه به‌صورت تصادفی شکل نمی‌گیرند، می‌توانیم نتیجه بگیریم که دو گره که دارای اشتراکاتی هستند (بعنوان مثال، شباهت در اطلاعات موقعیت جغرافیایی آنها یا الگوی ارتباط آنها با گره‌های دیگر) احتمال بیشتری دارد که به یکدیگر متصل شوند.

مسئله پیش‌بینی پیوند می‌تواند به‌صورت رسمی به شرح زیر باشد:

فرض کنید یک گراف شبکه $G=(V,E)$ داریم. حال باید پیش‌بینی کنیم که آیا پیوندی بین دو گره u و v بوجود می‌آید یا خیر، درصورتیکه $u.v \in V$، و $e(u,v) \notin E$.

شباهت گره‌ها را می‌توان با استفاده از معیارهای تجزیه و تحلیل مجاورت گره‌ها در یک شبکه همانند توزیع درجه‌ای، همسایه‌های مشترک، پیوست ترجیحی[۲]، ضریب جاکارد[۳]، شاخص لایچ–هولم نیومن[۴] و شاخص کارت[۵] پیدا کرد.

علاوه بر روشهای مبتنی بر گره که روشی آسان برای پیش‌بینی پیوند است، روشهای دیگر نیز برای این منظور توسعه‌یافته‌اند. بعنوان مثال، روشهای مبتنی بر راستنمایی[۶] ابتدا یک فرض در مورد سازمان و ساختار شبکه ایجاد می‌کنند و سپس از این فرض برای پیش‌بینی پیوندهای مفقوده استفاده می‌کنند.

---

[1] Directed, Weighted, and Evolving Networks
[2] Preferential Attachment
[3] Jaccard Coefficient
[4] Leicht-Holme-Newman Index
[5] Kart Index
[6] Likelihood-Based Methods

روشهای دیگر برای پیش‌بینی پیوند عبارتند از تقسیم یا فاکتوریزه کردن ماتریس و دسته‌بندی مبتنی بر ویژگی[1] که روش دوم تلاش می‌کند تا تعدادی از ویژگیها را از مجموعه ویژگیهای توپولوژیک یا غیر توپولوژیک شبکه انتخاب کند، به‌طوری‌که ویژگیهای انتخاب‌شده باید به‌طور مؤثر و مناسبی راستنمایی پیوند را توصیف کنند. به‌طورکلی، روشهایی که از ویژگیها برای آموزش یک مدل دسته‌بندی باینری استفاده می‌کنند، مسئله پیش‌بینی پیوند را بعنوان یک روش طبقه‌بندی تحت نظارت[2]، در نظر می‌گیرند، درحالی‌که روشهای مبتنی بر شباهت گره، به پیش‌بینی پیوند بعنوان روش طبقه‌بندی بدون نظارت[3] نگاه می‌کنند.

کار پیش‌بینی پیوند را می‌توان به سه دسته تقسیم کرد: ۱- پیش‌بینی پیوندهای مفقوده یا نامشهود[4] در یک شبکه، ۲- پیش‌بینی پیوندهایی که احتمالاً در آینده نزدیک شکل می‌گیرد و ۳-پیش‌بینی تعامل جدید بین یک جفت گره که تعامل یا ملازمت آنها در گذشته مشاهده شده است. با این وجود، اگر ویژگی یا هویت گره‌ها[5] به‌طور کامل در دسترس نباشد، ممکن است این کار پیچیده‌تر شود.

پیش‌بینی پیوند در زمینه‌های بسیاری مانند تطور و تکامل شبکه کاربرد دارد؛ زیرا امکان مطالعه ساز و کارهایی را فراهم می‌کند که اضافه و حذف (تولد و مرگ) گره‌ها را در این شبکه‌ها کنترل می‌کند. پیش‌بینی پیوند همچنین توسط سیستمهای تجارت الکترونیک برای ایجاد زیرسیستمهای توصیه‌گر محصول به مشتری استفاده می‌شود. در بخش بهداشت و درمان، پیش‌بینی پیوند می‌تواند برای شناسایی تعاملات بین داروها و پروتئینها و یا بین بیماریها و پروتئینها استفاده شود.

در شبکه‌های اجتماعی، روشهای پیش‌بینی پیوند از شباهت کاربران برای پیش‌بینی یا پیشنهاد دوستیهای جدید استفاده می‌کنند. بعنوان مثال، شباهت زیاد بین دو کاربر، احتمال شکل‌گیری دوستی میان آن دو در آینده را افزایش می‌دهد. به همین ترتیب، مسئله پیش‌بینی پیوند را می‌توان به مسئله دیگری (یعنی همگونی) تعمیم داد که بیان می‌کند مردم تمایل دارند با فردی از افراد دیگر دوست شوند که ویژگیهای شخصی مشابه مانند جنس، نژاد، سن، تحصیلات، یا مذهب با او دارند. بر همین اساس می‌توان

[1] Feature-Based Classification
[2] Supervised Classification
[3] Unsupervise Task
[4] Missing or Unobserved Links
[5] Attributes or Identities of Nodes

پیش‌بینی کرد که پس از ملاقات چهره به چهره بین دو نفر، آیا تعاملی بین آن دو شکل خواهد گرفت یا خیر و یا اینکه آیا خطر افشای اطلاعات حیاتی و مهم تحت شرایط خاص توسط برخی از افراد وجود دارد یا خیر.

مفهوم و روشهای پیش‌بینی پیوند در پیش‌بینی ائتلاف‌های جدید بین ملتها نیز کاربرد دارد. در شبکه‌های هم-تألیفی[1]، پیش‌بینی پیوند می‌تواند برای اندازه‌گیری احتمال هم-تألیفی در مورد دو نویسنده استفاده شود.

۱-۵-۳ شبکه‌های فضایی

یک شبکه فضایی[2] گرافی است که در آن گره‌ها در یک فضای متریک جاسازی شده‌اند، به این معنی که گره‌ها در یک فضای مجهز به یک متریک خاص قرار دارند. داده‌های فضایی دارای یک مرجع فضایی، یعنی، مقادیر مختصات و یک سیستم خاص مرجع[3] برای این مختصات می‌باشند.

مطالعه شبکه‌های فضایی در بسیاری از حوزه‌ها اعم از شهرسازی تا شیوع یا همه‌گیری بیماریها (اپیدمیولوژی) بسیار مهم است. اینترنت نمونه‌ای واضح از شبکه‌های فضایی است زیرا شامل مجموعه‌ای از گره‌ها (روترها یا مسیرها) است که با یالها (یا کابلهای) مختلف (و البته با طول کابل و زمان تأخیر متفاوت) بهم پیوند داده شده‌اند. ترسیم نقشه‌ای از انواع مختلف شبکه‌های ترابری (مانند خودرو و قطار)، نمونه دیگری از شبکه‌های فضایی است.

در شبکه‌های فضایی شهری (شبکه‌های حمل‌ونقل)، یالها عملاً خیابانها را نشان می‌دهند، درحالی‌که گره‌ها، تقاطعی هستند که در آن دو یا چند یال (مسیر) با هم برخورد می‌کنند. بااین‌حال، چنین شبکه‌هایی می‌توانند به‌صورت شبکه‌های سه-حالته[4] به‌جای شبکه‌های دو-حالته نمایان شوند. در چنین وضعیتی عملاً سه عنصر با این شبکه‌ها مرتبط هستند: یالها، مسیرهایی که عابران می‌توانند حرکت

[1] Coauthorship Networks
[2] Spatial Network
[3] Special System of Reference
[4] Tripartite Networks

کنند؛ گره‌ها، تقاطعهایی که بیش از دو یال از آنها عبور می‌کنند و ساختمانها، مکانهایی که ترافیک از آنها شروع و یا به آنها ختم می‌شود (یعنی ترافیک به درون آنها می‌رود و یا از آنجا به سمت خیابان حرکت می‌کند).

اگر می‌خواهیم یک شبکه فضایی ایجاد کنیم، ابتدا باید گراف یک سیستم (یعنی گره‌ها و یالها) را بکشیم و سپس نمایه‌های فیزیکی[1] را به شبکه اضافه کنیم؛ این کار به درک بهتر اهمیت فواصل فیزیکی بین عناصر بر روی عملکرد شبکه‌ها کمک می‌کند.

١-٥-٤ شبکه‌های تعامل پروتئین به پروتئین

شبکه‌های تعامل پروتئین–پروتئین *(PPI)* نوعی از شبکه‌های بیولوژیکی هستند. گره‌ها در این شبکه‌ها پروتئینهای انفرادی را نشان داده و تعامل بین پروتئینها با یالها نمایش داده می‌شوند. چنین تعاملاتی نمایشگر پیوند دو یا چند پروتئین برای انجام یک عملکرد بیولوژیکی خاص است. تجزیه و تحلیل شبکه‌های *PPI* برای درک کامل مبنای مولکولی بسیاری از بیماریها ضروری است.

روشهای متعددی برای مطالعه تعاملات بین پروتئینها وجود دارد که از آن میان، سیستم دوگانه مخمر[2] روش متداول برای کشف تعاملات پروتئین–پروتئین و تعاملات *DNA* به پروتئین است. روش کار بدین صورت است که تعاملات فیزیکی بین یک جفت پروتئین یا یک پروتئین واحد و یک مولکول *DNA* را آزمایش می‌کند.

از آنجا که تعاملات بین پروتئینها بسیار مهم است، آنها به شبکه‌های عمیقاً تجزیه و تحلیل شده در زیست‌شناسی تبدیل شده‌اند. این شبکه‌ها به وسیله رخدادهای زیست پزشکی[3] یا بر اثر نیروهای الکتریکی (یا هر دو مکانیزم) بوجود می‌آیند. آنها در زمینه‌های مختلف مانند شبکه‌های بیوشیمی، دینامیک مولکولی، انتقال سیگنال و دیگر موارد تحت مطالعه قرارگرفته‌اند. دیگر انواع شبکه‌های

[1] Physical Representations
[2] Yeast Two-Hybrid System
[3] Biomedical Events

بیولوژیکی عبارتند از هم-گونی ژن[۱]، متابولیسم[۲]، سیگنالینگ[۳]، عصبی، تعامل بین‌گونه‌ای[۴] و تعامل درون‌گونه‌ای[۵] و همچنین شبکه‌های غذایی[۶].

اتصال یا همبندی در شبکه‌های تعامل پروتئین-به-پروتئین برای زنده ماندن موجودات، حیاتی است، به‌طوری‌که برای بقای موجود زنده، نقش پروتئینهای با درجه بالایی از همبندی، بسیار مهم‌تر از پروتئینهای با درجه پایین‌تر از همبندی است.

۱-۵-۵ سیستمهای توصیه‌گر

سیستمهای توصیه (یا توصیه‌گرها)[۷] ابزارهای نرم‌افزاری هستند که پیشنهادهایی در رابطه با اقلامی که احتمالاً موردعلاقه یک کاربر خاص باشند، به او ارائه می‌دهند. آنها از افزایش تعداد اقلام فروخته‌شده، فروش اقلام متنوع‌تر، افزایش رضایت کاربر، افزایش وفاداری کاربران و ایجاد درک بهتر از تقاضای کاربر، سود می‌برند.

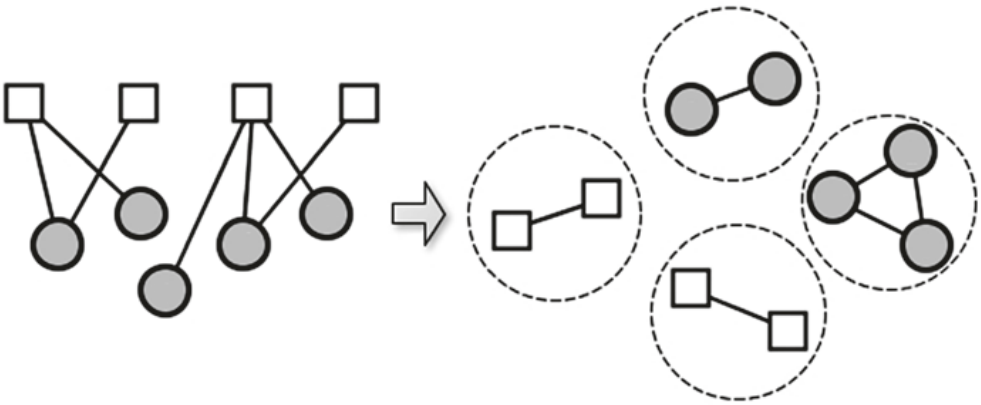
به دلیل محبوبیت خرده‌فروشان آنلاین در زمینه‌های مختلف مانند انواع کتاب، موسیقی، لوح سی‌دی، فیلم، لباس، اخبار، مقالات پژوهشی و غیره، سیستمهای توصیه‌گر در سالهای اخیر بسیار رایج شده‌اند. سامانه‌های خدمات آنلاین معروف که توصیه‌هایی را برای بازدیدکنندگان ارائه می‌دهند، عبارتند از *Amazon.com* و *eBay.com* و چندین موتور توصیه‌گر موسیقی آنلاین مانند *Lastfm.com* و *Pandora.com*، که فهرستهای پخش موسیقی را بر اساس انتخاب کاربر و ویژگیهای آهنگها مانند سبک، آهنگ‌ساز (نوازنده) و خواننده تنظیم می‌کنند.

---

[1] Gene Co-Expression
[2] Metabolic
[3] Signaling
[4] Between-Species Interaction
[5] Within-Species Interaction
[6] Food Webs
[7] Recommenders

ارائه توصیه به کاربران می‌تواند به دو صورت انجام شود: (۱) پالایش مبتنی بر محتوا[1] و (۲) پالایش ائتلافی یا مشارکتی[2]. سامانه توصیه‌گر در روشهای پالایش مبتنی بر محتوا *(CBF)* موارد مشابهی که در گذشته مورد ترجیح کاربر واقع‌شده است، را به او توصیه می‌کند. روشهای فیلتر یا پالایش مشارکتی *(CF)* عملکرد گذشته کاربر (بعنوان مثال، اقلامی که قبلاً خریداری‌شده یا رتبه‌بندی شده‌اند) را در کنار تصمیمهای مشابه دیگر کاربران، در نظر می‌گیرند تا علاقه‌های کاربر را پیش‌بینی کنند. این روشها بر این فرض تأکید دارند که کاربران مشابه، اقلام مشابه را انتخاب می‌کنند.

یک شبکه اقلام-کاربر[3] را می‌توان بر اساس کالاهایی که کاربر در گذشته به آنها ابراز علاقه نموده و یا خریداری کرده است، ایجاد کرد. رابطه بین دو کاربر، درصورتیکه موارد مشترک بیشتری داشته باشند، قوی‌تر خواهد شد. این شبکه، شبکه دو–حالته[4] نامیده می‌شود، زیرا دارای دو نوع گره، یعنی کاربران و اقلام است. یک فرآیند تجزیه و تحلیل ساده سپس برای تقسیم آن شبکه به دو شبکه فرعی می‌تواند با کنار هم قرار دادن اقلام مشابه در یک گروه انجام شود. توصیه‌های بعدی بر اساس اینکه روابط تا چه حد قوی هستند مطرح می‌شود. از این سامانه‌های توصیه‌گر همچنین بعنوان سامانه‌های توصیه‌گر اجتماعی یاد می‌شود؛ بعلاوه این روشها، ریشه‌هایی در نظریه همگونی یا همریختی دارند (شکل ۱۴–۱).



شکل ۱۴–۱ تبدیل یک شبکه دو حالته به شبکه‌های فرعی

[1] Content-Based Filtering (CBF)
[2] Collaborative Filtering (CF)
[3] User-Item Network
[4] Two-Mode Network

شکل ۱۴-۱ نشان می‌دهد چگونه می‌توان چندین زیرشبکه کاربر-کاربر و آیتم -آیتم را از شبکه اصلی کاربر-آیتم انتخاب کرد که در آنها کاربران با آیتمهایی که دوست دارند یا خریداری کرده‌اند پیوند داده می‌شوند.

به غیر از کسب‌وکارهای آنلاین که سامانه‌های توصیه‌گر را بکار گرفته‌اند، برخی از وب‌سایتهای شبکه‌های اجتماعی (مانند فیسبوک) هم توصیه‌هایی به کاربران ثبت‌شده خود می‌کنند تا به آنان در جهت یافتن دوستان جدید و گسترش شبکه‌های خود، بر اساس ارتباطات اجتماعی موجود و علائق مشابه کمک کنند. هدف از این توصیه‌ها، پیش‌بینی ایجاد روابط با کسانی است که قبلاً با آنها در ارتباط نبوده‌اند. چنین شبکه‌هایی بعضی اوقات سامانه‌های توصیه دوستی[۱] یا (*FRSs*) نامیده می‌شوند. این امر نیز می‌تواند بعنوان یک مسئله پیش‌بینی پیوند در نظر گرفته شود.

۱-۶ بسته *iGraph*

ابزارهای مختلفی برای تجزیه و تحلیل شبکه‌های اجتماعی مانند *MuxViz* ، *NetworkX* (استفاده‌شده در این کتاب) و *iGraph* وجود دارد. علاوه بر آنها، تعدادی از روشهای استاندارد برای تجسم اطلاعات[۲] مانند جعبه نمودارها[۳]، نمودارهای پراکنده[۴]، ابرِ کلمه[۵] و درخت تصمیم‌گیری، نیز توسعه داده شده‌اند. از آنجا که *iGraph* یک ابزار بصری بسیار قدرتمند است، قصد داریم از آن در این بخش برای ساخت نمودار از طریق *RStudio* (محیطی برای توسعه نرم‌افزار *R*) استفاده نماییم.

بسته *iGraph* یک نرم‌افزار برای تحلیل و تجسم شبکه‌های بزرگ است. این نرم‌افزار امکان انجام امور مربوط به تجزیه و تحلیل شبکه، مانند انجمن‌یابی، گروههای منسجم، حفره‌های ساختاری، دووند، سه‌وند و شمارش موتیف[۶] را به تحلیلگر می‌دهد. این برنامه همچنین دارای کاربردهای متعددی در

[1] Friend Recommender Systems
[2] Visualize Data
[3] Box Plots
[4] Scatter Plots
[5] Word Clouds
[6] Motif Count

مسائل خاص‌تر گراف است، مسائلی همانند درخت پوشا (یا فراگیر) کمینه[۱] و جریانهای شبکه. این بسته، یک نرم‌افزار منبع باز[۲] *(OSS)* است که می‌تواند بر روی سیستم عاملهای مختلف مانند ویندوز، لینوکس و اُ اس ایکس[۳] اجرا شود. *iGraph* رابطهایی برای *C*، *R*، *Python* و زبانهای برنامه‌نویسی روبی[۴] دارد. ما استفاده از *iGraph* را در زبان برنامه‌نویسی *R* و رابط *RStudio* در نظرخواهیم گرفت. این به این معنی است که ابتدا *R* را نصب می‌کنیم و سپس *RStudio* را نصب می‌کنیم.

ما از تابع *layout ()* برای تعیین مکان راسها در شبکه و تابع *plot ()* برای ترسیم شبکه استفاده می‌کنیم. بررسی ویژگیهای مختلف از طریق *RStudio* با استفاده از هر یک از این توابع با اجرای کد زیر امکان‌پذیر است.

?<function name> function

باید بسته *iGraph* را از مسیر *packages → install → iGraph* نصب کنیم. پس از نصب *iGraph*، باید بسته *iGraph* را در *RStudio* با دستور زیر بارگذاری کنیم.

```
require(igraph)
```

بیایید یک نمودار ساده بسازیم تا ببینیم این بسته چگونه کار می‌کند. از یک بردار (*c*) استفاده می‌کنیم که در آن هر جفت عنصر، مشخص‌کننده یک یال است. به‌طور مثال عنصر اول به عنصر دوم اشاره دارد و یا عنصر سوم به چهارم و به همین ترتیب؛ که با استفاده از کلید فاصله می‌توان جفت گره‌ها را جدا نمود. همچنین تعداد گره‌ها (اختیاری) را می‌توان مشخص نمود.

```
g <- graph(c(1,2, 1,3, 2,3, 3,4, 4,5, 4,6, 6,7, 6,8),n=9)
```

برای نشان دادن این‌که چگونه یالهای گراف به هم متصل شده‌اند، فرمان زیر را اجرا کنید:

```
Print(g)
```

[1] Minimum Spanning Trees
[2] Open Source Software
[3] OSX
[4] Ruby Programming Languages

نتیجه دستور فوق:

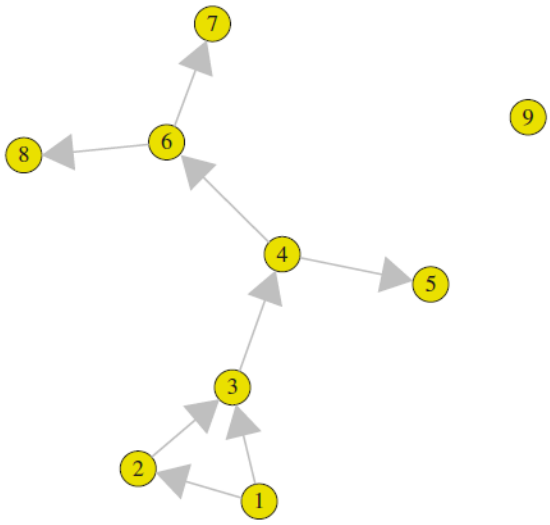
```
IGRAPH D--- 9 8 --
+ edges:
 [1] 1->2 1->3 2->3 3->4 4->5 4->6 6->7 6->8(
```

برای رسم گراف از تابع *plot()* استفاده نمایید. در شکل ۱۵-۱ رنگ گره‌ها نیز، با استفاده از دستور زیر، مشخص‌شده است.

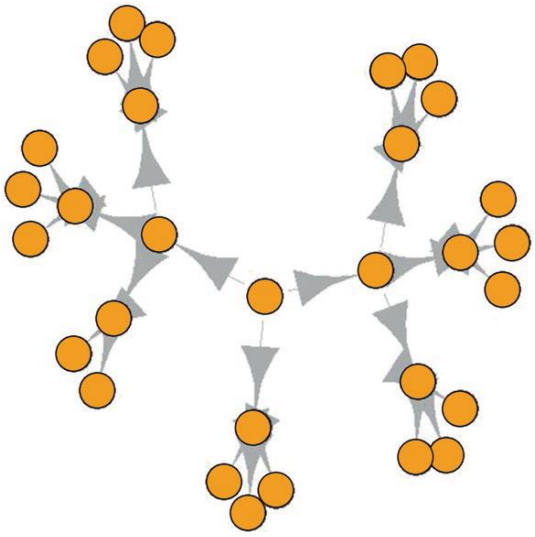
```
plot(g, vertex.color="yellow")
```

نتیجه کار، یک گراف غیرمتصل و جدا خواهد بود که دو بخش دارد: بزرگ‌ترین قسمت شامل هشت گره است، کوچک‌ترین قسمت نیز تنها یک گره (مجزا) دارد. جهت رسم گرافهای تصادفی نیز می‌توان از *iGraph* استفاده نمود. ما یک درخت با ۳۰ راس (ریشه) و سه فرزند (برگ) برای هر راس، با استفاده از دستور زیر، ایجاد خواهیم کرد (شکل۱۶-۱).

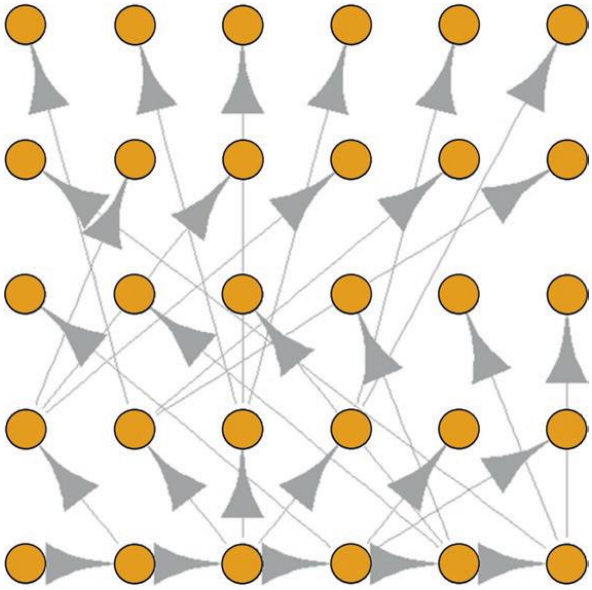
```
g <- graph.tree(30, 3)

plot(g)
```



شکل ۱۵-۱ نمایش گراف ساده در *iGraph*



شکل ۱۶-۱ نمایش گراف تصادفی در *iGraph*



شکل ۱۷-۱ نمایش گراف تصادفی با چیدمان *davidson.harel*

چیدمانهای مختلفی برای ترسیم این گراف وجود دارد، به‌طور مثال *fruchterman.reingold*، *graphopt*، *kamada.kawai*، *gem* و غیره. در شکل ۱۷-۱ چیدمان *davidson.harel* وجود دارد.

ما همچنین می‌توانیم از تابع *tkplot* برای انتقال نقاط یک گراف استفاده کنیم که به ما یک چیدمان صفحه تعاملی[1] می‌دهد.

```
Tkplot(g, layout.kamada.kawai, vertex.color="yellow")
```

نتیجه دستور بالا یک صفحه تعاملی است که به ما اجازه می‌دهد تا گره‌ها را بکشیم و موقعیت آنها را تغییر دهیم که یک ویژگی عالی است. تابع دیگری به نام *rglplot* وجود دارد که امکان ترسیم نمودار سه‌بعدی را ایجاد می‌نماید و بینش عمیق‌تری از ساختار گراف به دست می‌دهد. قبل از اینکه بتوانیم از این تابع استفاده کنیم، باید بسته *rgl* در *RStudio* را با استفاده از مسیر زیر، نصب کنیم.

choose packages → install → rgl

```
rglplot(g, lyout=layout.kamada.kawai, vertex.color="cyan")
```

[1] Interactive Plotting Pane

فصل دوم

مبانی شبکه

این فصل "مفهوم شبکه" را معرفی می‌کند؛ موضوعی که عملاً هسته اصلی مبحث "تحلیل شبکه" است. ما در اینجا موضوعاتی همانند انواع شبکه‌ها، سنجه‌ها یا معیارهای شبکه[۱]، نصب و استفاده از کتابخانه *NetworkX*، نمایش داده‌های شبکه[۲]، عملیات پایه‌ای ماتریسی[۳] و تجسم بخشی یا تصویرسازی داده‌ها[۴] را مورد بحث قرار خواهیم داد.

۲-۱ شبکه چیست؟

شبکه را می‌توان نوع خاصی از گراف ریاضی[۵] یا سیستمهای بهم متصل (و یا همبند)[۶] دانست. ازاین‌رو مفهوم شبکه از تعریف گراف که مستلزم نمایش بصری مجموعه‌ای از گره‌ها و یالهاست، دور نیست.

---

۱ Network Measures

۲ Network Data Representation

۳ Basic Matrix Operations

۴ Data Visualization

۵ Mathematical Graph

۶ Interconnected Systems

گره‌های شبکه ممکن است صفحات وب، افراد، سازمانها، مقالات، مکانها و بسیاری چیزهای دیگر را نشان دهند.

۲-۲ انواع شبکه‌ها

شبکه‌ها در دنیای واقعی به یکی از چهار گونه اصلی زیر تقسیم می‌شوند:

- **شبکه‌های اطلاعاتی[۱]:** این شبکه‌ها، ساخته (یا دست‌ساز) انسان[۲] هستند که در آن اقلام داده‌ای به نحوی با یکدیگر پیوند دارند. وب گسترده جهانی *(WWW)* بهترین مثال از این نوع است، شبکه‌ای که در آن راسها صفحات وب بوده و این صفحات نشاندهنده متنها، تصاویر و سایر اطلاعات هستند. جابجایی یا اصطلاحاً ناوبری[۳] از یک صفحه به صفحه دیگر با استفاده از ابرلینکها انجام می‌شود که نمایشگر یالهای شبکه هستند. نمونه‌های دیگری از شبکه‌های اطلاعاتی – علیرغم آنکه دارای جنبه‌های اجتماعی نیز می‌باشند– شامل ارتباطات ایمیلی[۴]، وب‌سایتهای شبکه‌های اجتماعی مانند فیسبوک و توییتر، وبلاگها و ارجاعات یا استنادات[۵] هستند.

- **شبکه‌های فناوری[۶]:** این نوع شبکه‌ها نیز ساخت بشر هستند که برای توزیع کالاها یا منابع، طراحی می‌شوند. این شبکه‌ها عمدتاً در طول قرن گذشته رشد کرده و از پیشرفتهای نوپدید در فنآوری استفاده می‌کنند. یکی از بهترین نمونه‌ها، اینترنت است، که یک شبکه فنآوری جهانی[۷] بوده و شامل دستگاههای ارتباطی برای اتصال کامپیوتر و سایر تجهیزات اطلاعاتی می‌شود. مثالهای

---

۱ Information Networks
۲ Man-Made Networks
۳ Navigation
۴ Email Communications
۵ Citation
۶ Technological Networks
۷ Global Technological Network

دیگر شامل شبکه‌های نیروی برق یا قدرت[۱]، شبکه‌های حمل‌ونقل، شبکه‌های تحویل و توزیع و شبکه‌های مخابراتی یا تلفن است.

- **شبکه‌های بیولوژیکی[۲]:** این نوع شبکه‌ها الگوهای تعامل بین عناصر بیولوژیکی را نشان می‌دهند. نمونه‌هایی از شبکه‌های بیولوژیکی عبارتند از: (۱) شبکه‌های بیوشیمیایی: شبکه‌هایی که نشاندهنده الگوهای تعامل و مکانیسمهای کنترل در سطح مولکولی در سلولهای بیولوژیکی هستند. (۲) شبکه‌های عصبی: شبکه‌هایی که نورونها را در مغز انسان و نیز در دستگاه‌های عصبی مرکزی[۳] حیوانات بهم متصل می‌کنند. (۳) شبکه‌های زیست محیطی: شبکه‌هایی که تعامل اکولوژیکی[۴] بین گونه‌های مختلف محیط زیست را نشان می‌دهند.

- **شبکه‌های اجتماعی:** شبکه‌هایی که در آن راسها، نمایانگر مردم و یالها، برخی از انواع تعاملات اجتماعی[۵] بین راسها شامل دوستی، خویشاوندی، جریان اطلاعات، شیوع بیماری و غیره را نشان می‌دهند.

- **شبکه‌های پیچیده[۶]:** ویکی‌پدیا[۷] شبکه‌های پیچیده را بعنوان شبکه‌هایی با ویژگیهای توپولوژیکی غیرمعمول تعریف می‌کند؛ ویژگیهایی که در شبکه‌های ساده مانند گرافهای مشبک[۸] یا گرافهای تصادفی[۹] رخ نمی‌دهند، اما اغلب در گرافهای واقعی اتفاق می‌افتند. نمونه‌هایی از شبکه‌های پیچیده شامل شبکه‌های بیولوژیکی (مثلاً پروتئین، مولکولها و واکنشهای شیمیایی)، شبکه‌های مغزی (مانند شبکه‌های فیزیولوژیکی و مسیرهای عصبی[۱۰])، شبکه‌های اجتماعی و اینترنت می‌باشند.

---

[۱] Power Grids
[۲] Biological Networks
[۳] Central Nervous Systems
[۴] Ecological Nnteractions
[۵] Social Interaction
[۶] Complex Networks
[۷] Wikipedia
[۸] Lattices Graphs
[۹] Random Graphs
[۱۰] Neurological Pathways

شبکه‌های پیچیده، نوعاً دارای ویژگیهای نوپدید یا نوظهوری[۱] هستند که از طریق واحدهای انفرادی آنها نگهداری شده و یا بروز می‌کند. شبکه‌های اجتماعی هم نوعی شبکه پیچیده هستند زیرا آنها ویژگیهای توپولوژیکی غیرمعمول[۲] را نشان می‌دهند، یعنی ویژگیهایی که در شبکه‌های ساده رخ نمی‌دهند.

به‌طور معمول یک شبکه مجموعه‌ای از گره‌ها است که از طریق پیوندها بهم متصل می‌شوند. گره‌ها و پیوندها می‌توانند با توجه به حوزه مورد نظر هر چیزی باشند. برای مثال، در اینترنت، گره‌ها به روترها اشاره دارند و پیوندها همان خطوط فیبر نوری هستند که آنها را به یکدیگر مرتبط می‌کنند. در *WWW*، گره‌ها عملاً فایلهای اسناد را نشان داده و پیوندها نشانگر ابرلینکها هستند. در شبکه استناد علمی، گره‌ها نشانگر مقالات و پیوندها نشاندهنده استناد به آنها، هستند. در شبکه اجتماعی، گره‌ها، عملاً افراد و پیوندها هم روابط را نشان می‌دهند. برخی از تولیدکننده‌های شبکه‌های پیچیده (نمونه‌هایی برای کارهای تحقیقاتی) عبارتند از گراف اردوش رینی[۳]، گراف واتس–استروگاتز[۴]، گراف باراباسی–آلبرت[۵] و گراف تصادفی–لوبستر[۶].

۲-۳ ویژگیهای شبکه‌ها

شبکه‌ها با دو نوع ویژگی مشخص می‌شوند: ایستایی و پویایی

۱) "خواص ایستای شبکه" همواره ثابت هستند. آنها ویژگیهای گره‌ها و یالها هستند و پس از زمانی که شبکه ایجاد می‌شود، تغییر نمی‌کنند (بعنوان مثال، تراکم، کوتاهترین مسیرها و مرکزیتها).

---

[۱] Emergent Properties

[۲] Nontrivial Topological Features

[۳] Erdos-Renyi Graph

[۴] Watts-Strogatz Graph

[۵] Barabasi-Albert Graph

[۶] Random-Lobster Graph

۲) "خواص پویای شبکه" ثابت نیستند. آنها می‌توانند فرایندهای در حال پیشرفت بر روی لایه‌زیرین یا زیرساخت شبکه‌ها باشند (بعنوان مثال، پویایی جمعیت در شبکه‌های زیست‌محیطی و یا سرایت بیماری در شبکه‌های اجتماعی). این خصوصیات همچنین می‌توانند فرایندهای دینامیکیِ شبکه‌های پیچیده و خواص ساختاری/آماری[۱] آنها باشند؛ مانند شکل‌گیری شبکه‌های اجتماعی، تشکیل شبکه غذایی در یک مقیاس زمانی زیست‌محیطی/تکاملی[۲]، رشد اینترنت و *WWW*، رشد شبکه‌های علمی استنادی و اثرات حذف یا افزودن گره/پیوند.

با وجود برخی اختلافات، انواع شبکه‌های مختلف دارای ویژگیهای مشترک زیر هستند:

- **تراگذری** (خوشه‌گی) به این معنی است که اگر گره A به گره B متصل شده و گره B به گره C متصل شود، احتمال زیادی وجود دارد که گره A نیز به گره C متصل شود.

- **استواری و آسیب‌پذیری شبکه**[۳]، به این معنی است که شبکه‌ها به تغییرات پویای توپولوژیکی ناشی از نیروهای خارجی حساس هستند (بعنوان مثال، حذف گره). حذف گره می‌تواند نتیجه یکی از عوامل زیر باشد: (۱) خطا، که حذف تصادفی گره‌ها است و (۲) حمله، که حذف انتخابی گره‌های متصل‌تر است. حذف گره منجر به تغییراتی در اتصالات و طول مسیر مشخصی می‌شود. اگر حذف گره ادامه یابد، شبکه درنهایت فرو خواهد ریخت (تکه‌تکه می‌شود)[۴]. با این وجود، تمام شبکه‌ها در پاسخ به حذف گره، رفتار مشابهی را نشان نمی‌دهند.

- **الگوهای امتزاج یا اختلاط**[۵] که اشاره می‌کند به گرایش راسها برای اتصال به دیگر راسهایی که با آنها از جهاتی شباهت دارند (یا ندارند). شبکه‌های اجتماعی به الگوی اختلاط هماهنگ یا همسان[۶] تمایل دارند. بعنوان مثال، افرادی که هم‌سن هستند، احتمال بیشتری دارد که با یکدیگر دوست باشند و یا افرادی که قومیت یا نژاد یکسانی دارند تمایل بیشتری برای معاشرت با هم دارند. از

---

۱ Structural Statistical Properties/

۲ Food Web Formation over Ecological/Evolutionary Time Scales

۳ Network Robustness and Vulnerability

۴ Fragmentation

۵ Mixing Patterns

۶ Assortative Mixing Pattern

سوی دیگر، شبکه‌های فناورانه و بیولوژیکی بیشتر به الگوی اختلاط ناهماهنگ یا ناهمسان[۱] تمایل دارند.

- **ساختار اجتماع**[۲]، به این معنی است که در برخی از شبکه‌ها ممکن است چگالیِ یال مابین گره‌ها در درون یک گروه خاص خیلی بالا بوده، درحالیکه چگالیِ یال بین گره‌های گروههای دیگر، پایین باشد. (یعنی چگالی یالها در انجمنها، از چگالی یالها در غیرانجمنها بیشتر است).

علاوه بر این، دانشمندان ویژگیهای بیشتری یافته‌اند که بیشتر مربوط به شبکه‌های بزرگ هستند (شبکه‌هایی با هزاران، صدها هزار یا حتی میلیونها گره). ویژگیهایی همچون:

- تمایل به ایجاد گروههایی از آشنایان و دوستان و نهایتاً تشکیل محفل (یعنی حداکثر مولفه‌های متصل) که همه گره‌های آن محفل یکدیگر را می‌شناسند.

- پیروی درجه گره از توزیع قانون قدرت، که این قانون، احتمال داشتن تعداد مشخصی یال برای یک گره تصادفی را نشان می‌دهد.

۲-٤ سنجه‌های شبکه

ما در اینجا ضمن نام بردن از هر یک از سنجه‌ها یا معیارهایی که به‌طور گسترده در تحقیقات تحلیل شبکه استفاده می‌شوند، تعریف مختصری نیز برای آنها ارائه می‌دهیم. اکثر این معیارها از نظریه گراف، اقتباس شده‌اند:

۱) **محدودیت تجمعی**[۳]**:** مجموع محدودیتهای دووندی یا دوگاهی[۱] بر روی تمام یالها در یک راس خاص. (واژه محدودیت در اینجا، به وجود مانع برای واسطه‌گری اشاره می‌کند؛ یعنی در صورت ارتباطات بیشتر، کار میانجیگری هم سخت‌تر می‌شود. این موضوع در فصل شش ارائه می‌گردد).

---

[۱] Disassortative Mixing Pattern

[۲] Community Structure

[۳] Aggregate Constraint

۲) **درجه متوسط[۲]:** معیاری که نشاندهنده انسجام ساختاری[۳] یک شبکه است. برای محاسبه آن، تمام درجه‌ها ابتدا جمع شده و سپس تقسیم بر تعداد کل گره‌ها در شبکه می‌شود.

۳) **توزیع درجه:** فراوانی درجه گره‌ها است. گرافهای با "توزیع درجه‌ای" پایین برای "قانون قدرت"، گرافهای بدون مقیاس[۴] نامیده می‌شوند. این سنجه، مشخصات کلی و نادقیقی از نحوه توزیع اتصالات در شبکه می‌دهد.

۴) **میانگین کوتاهترین مسیر[۵]:** میانگین طول کوتاهترین مسیر برای تمام جفت گره‌ها مشخص می‌کند که جامعه یا جهانی که توسط شبکه نمایش داده می‌شود تا چه حد بزرگ است. با توجه به این سنجه، یک طول کوچک مسیر، نشان می‌دهد که شبکه در کل به‌خوبی همبند شده است.

۵) **گریز از مرکز یا برون‌مرکزی:** حداکثر طول کوتاهترین مسیر از هر گره. (یعنی آیا گره‌ای از مرکزیت دور است و یا به آن نزدیک است).

۶) **قطر:** طولانی‌ترین مسیر در میان کوتاهترین مسیرها در تمام جفت گره‌ها در شبکه است (حداکثر برون‌مرکزی در شبکه). هدف از اندازه‌گیری قطر، نشان دادن گستردگی شبکه است و به این معنی است که دورترین گره‌ها در شبکه، تا چه مقدار از یکدیگر فاصله دارند. شعاع[۶] عملاً کمترین دورمرکزی در شبکه است.

۷) **دووَند (دوگاه):** یک جفت گره هستند که از طریق یک یا چند پیوند، بهم متصل هستند.

---

[۱] Dyadic Constraint
[۲] Average Degree
[۳] Structural Cohesion
[۴] Scale-Free
[۵] Average Shortest Path
[۶] Radius

۸) **محدودیت دووَندی:** محدودیت دووَندی در راس u (که از طریق ارتباط بین راس u و راس v تصویر می‌شود) نشان می‌دهد تا چه اندازه، راس u روابط بیشتر و قوی‌تری دارد با همسایگانی که به‌شدت با راس v مرتبط هستند.

۹) **فاصله کروی (ژئودزیک):** یک فاصله کروی عملاً کوتاهترین مسیر بین دو راس است.

۱۰) **میانگین مسافت ژئودزیک:** میانگین طولانی‌ترین مسیر در میان تمام جفتهای متصل در شبکه فردانه (یا خود–محور)[۱].

۱۱) **چندگانگی:** تعداد دفعاتی که یک خط خاص (بین یک جفت مرتب یا نامرتب از راسها) در یک شبکه ترسیم شده و یا رخ می‌دهد. (در عمل، گره‌هایی که تعداد خطوط بیشتری بین آنهاست، روابط قویتری دارند).

۱۲) **معروفیت یا محبوبیت[۲]:** شهرت یک راس در یک شبکه جهتدار، تعداد کمانهایی[۳] است که دریافت می‌کند.

۱۳) **سه‌وَند (سه‌گاه):** زیر شبکه‌ای که متشکل از سه گره است.

نوع تجزیه و تحلیلهایی که در گرافها اعمال می‌شود (مانند کاردینالیتی[۴]، گذرگاههای شبکه[۵] و اجتماع یا انجمن‌یابی) دارای پایه‌های ریاضی مفصلی است؛ بنابراین، از یکی از بزرگترین کتابخانه‌های پایتون یعنی کتابخانه *NetworkX* برای تحلیل شبکه استفاده خواهیم کرد.

---

[۱] Ego Network

[۲] Popularity

[۳] Arcs

[۴] Cardinality

[۵] Network Traversals

۲-۵ کتابخانه NetworkX

کتابخانه *NetworkX* یک بسته نرم‌افزاری زبان پایتون و یک ابزار منبع باز برای ایجاد، دستکاری و مطالعه ساختار، پویایی و عملکرد شبکه‌های پیچیده است. این کتابخانه، یک ابزار مدل‌سازی و محاسباتی شبکه[۱] است و نه توسعه ابزار نرم‌افزار[۲]. این بسته می‌تواند شبکه‌ها را بارگیری، ذخیره و تجزیه و تحلیل نموده، شبکه‌های جدیدی را تولید کرده، مدلهای شبکه را ساخته، و شبکه‌ها را رسم کند. اولین انتشار عمومی این کتابخانه که همه مبتنی بر پایتون بود، در آوریل ۲۰۰۵ انجام شد. البته این کتابخانه می‌تواند با زبانهای غیر از پایتون مانند *C*، *C ++* و *FORTRAN* کار کند.

کتابخانه *NetworkX*، یعنی مجموعه ابزار گرافیکی مبتنی بر پایتون[۳]، نوعی از پایگاه داده‌های گرافیکی حافظه است. مقدار کاری که می‌توان انجام داد، مستقیماً با مقدار حافظه کاری دستگاهی که بر روی آن سیستم خود را اجرا می‌کنیم، متناسب است. از طریق کاهش اندازه داده‌ها و یا استفاده از یک دستگاه با حافظه بزرگتر می‌توان عملکرد نهایی را بهبود داد.

اگرچه کتابخانه *NetworkX* برای مسائل در مقیاس بزرگ[۴] با نیازهای پردازشی سریع، مناسب و ایده‌آل نیست، اما گزینهٔ بسیار خوبی است برای تجزیه و تحلیل شبکه‌های دنیای واقعی، زیرا:

- اکثر الگوریتمهای اصلی با استفاده از کدهای قدیمی[۵] اما بسیار سریع تهیه شده‌اند.
- از الگوریتمهای گراف استاندارد[۶] استفاده می‌کند.
- مجموعه گسترده‌ای از فرمتهای قابل‌خواندن و نوشتن بومی یا محلی دارد.
- به‌آسانی بر روی پلتفرمهای بزرگ[۷] که دارای قدرت بالایی در بهنگام‌سازی برخط مستندات هستند قابل نصب و استفاده می‌باشد.

---

[۱] Computational Network Modeling Tool
[۲] Software Tool Development
[۳] Python-Based Graph Toolkit
[۴] Large-Scale Problems
[۵] Legacy Code
[۶] Standard Graph
[۷] Main Platforms

- برای نشان دادن شبکه‌های مختلف مانند گرافهای کلاسیک، گرافهای تصادفی و شبکه‌های مصنوعی، ایده آل است.
- از مزیت پایتون برای واردکردن داده‌ها[۱] از منابع خارجی[۲] استفاده می‌کند.
- کتابخانه *NetworkX* شامل انواع توابع تولیدکننده گراف[۳] و امکاناتی برای خواندن و نوشتن گرافها در بسیاری از فرمتها همانند *edgelist*. ، *adjlist*. ، *gml*. ، *graphml*. ، *pajek*. و غیره هست.

با این حال، بسته‌ها و ابزارهای دیگری نیز برای تجزیه و تحلیل و مصورسازی شبکه در پایتون در دسترس هستند؛ ازجمله:

- *PyCX Project: http://pycx.sf.net/*
- *ComplexNetworkSim*
- *SimPy*
- *graph-tool [http://graph-tool.skewed.de/]*
- *pyGraphViz (GraphViz) [http://www.graphviz.org/]*
- *igraph [http://igraph.org/]*
- *python-graph*
- *gephi [http://gephi.org/]*

٦-٢ نصب

- مثالهای این کتاب با استفاده از *Anaconda* که در وب‌سایت *Continuum Analytics* موجود است، توسعه داده شده است. این نرم‌افزار از جمله موارد توزیع رایگان پایتون است و شامل تمام بسته‌هایی است که برای اجرای کد (و چیزهای بیشتر) نیازمند هستید.

---

[۱] Import Data

[۲] Outside Sources

[۳] Graph Generator Functions

- نرم‌افزار*Anaconda* نصب آسانی دارد. به‌طور پیش‌فرض، نصب در سطح کاربر[۱] انجام می‌شود، نه در سطح سیستم، بنابراین شما به امتیازات ادمین[۲] نیاز ندارید. علاوه بر این، از پایتون ۲ و پایتون ۳ پشتیبانی می‌کند. شما میتوانید *Anaconda* را از *Continuum* دانلود کنید.

- اگر شما نمی‌خواهید از *Anaconda* استفاده کنید، به بسته‌های زیر نیاز خواهید داشت:
  - ✓ *NetworkX* برای تحلیل شبکه
  - ✓ *Numpy* برای محاسبات عددی پایه
  - ✓ *SciPy* برای محاسبات علمی از جمله آماری
  - ✓ *Matplotlib* برای تصویرسازی

- اگرچه اینها، بسته‌های نرم‌افزاری معمولی مورداستفاده هستند، اما در نصب پایتون گنجانده نشده‌اند و ممکن است نصب آنها در برخی از محیطها سخت باشد. اگر شما در نصب آنها مشکل دارید، توصیه می‌شود از *Anaconda* یا یکی دیگر از توزیعهای پایتون که شامل این بسته‌ها هستند، استفاده کنید.

برای چاپ فهرست انواع گراف موجود در بسته *NetworkX*، به‌سادگی دستور[۳] زیر را وارد کنید:

```
In:  print[s for s in dir(nx) if s.endswith('graph')]
Out:
     ['LCF_graph', 'barabasi_albert_graph', 'barbell_graph',
     'binomial_graph', 'bull_graph', 'caveman_graph',
     'chordal_cycle_graph', 'chvatal_graph',
     'circulant_graph', 'circular_ladder_graph',
     'complete_bipartite_graph', 'complete_graph',
     'complete_multipartite_graph', 'connected_caveman_graph',
     ……..]
```

[۱] User-Level Installation

[۲] Administrative Privileges

[۳] Command

در *IPython* می‌توانید کلاس شی را بصورت *class obgect?* تایپ کنید تا جزئیات مربوط به آن شی را دریافت کنید. اجازه دهید اطلاعاتی در مورد شیء گراف نردبان[1] بدست آوریم:

```
In:  nx.ladder_graph?
Out:
     Signature:
     nx.ladder_graph(n, create_using=None)
     Docstring:
     Return the Ladder graph of length n.
     This is two rows of n nodes, with each pair connected by a
single
     edge.
     Node labels are the integers 0 to 2*n - 1.
     File:
     c:\users\zuhair\anaconda2\lib\sitepackages\
     networkx\generators\classic.py
     Type:
     function
```

بیایید دو مثال زیر را بررسی کنیم تا نشان دهیم چگونه می‌توان کتابخانه *NetworkX* را بکار گرفت. این مثالها به درک و شهود بهتر گرافها کمک می‌کند.

مثال ۱

در این مثال، کتابخانه *NetworkX* را تحت نام *nx* وارد خواهیم کرد. با کمک این کتابخانه، ما می‌توانیم فی‌المثل یک گراف بدون جهت را با شروع از یک نمونه از کلاس *nx.Graph ()* ایجاد کنیم.

ابتدا یک گراف ساده جهتدار را با استفاده از *nx.Graph ()* ایجاد کرده و سپس یک گره را با استفاده از روش *nx.add_node ()* به آن اضافه کنید و در ادامه مجموعه‌ای از گره‌ها را با استفاده از روش *nx.add_nodes_from()* به آن بیفزایید که فهرستی از آیتمها را باز می‌گرداند:

```
In [1]:  import networkx as nx
     %matplotlib inline
     import matplotlib.pyplot as plt
```

[1] Ladder Graph

```
In:  g = nx.Graph()
     g.add_node(1)
In:  g.nodes()
Out: [1]
In:  g.add_nodes_from([2, 3, 4])
     g.nodes()
Out: [1, 2, 3, 4]
```

اکنون، این گره‌ها باید متصل شوند که اینکار با استفاده از روش *nx.add_edge ()* انجام می‌شود. برای اضافه‌کردن فهرستی از یالها، به‌جای یک یال تکی در هر زمان، می‌توانیم از روش *nx.add_edges_from ()* استفاده کنیم و یک فهرست چندتایی از گره‌ها[۱] را برای اتصال به آن اضافه کنیم.

```
In [7]: g. add_edge(1, 2)
In [8]: g.edges()
Out [8]: [(1, 2)]
In [9]: g.add_edges_from([(3, 4), (5, 6)])
In [10]: g.edges()
In [10]: [(1, 2), (3, 4), (5, 6)]
```

روش *ad_weighted_edges_from ()* برای اضافه کردن یالهای وزن‌دار استفاده می‌شود. وزنها، در این مورد، می‌توانند فاصله، فرکانس، نوع رابطه یا غیره را نشان دهند. در این روش، آن دو گره‌ای که قرار است متصل شوند و همچنین وزن آنها را مشخص می‌کنیم. بطور اختیاری، می‌توان از گزاره *data = Ture* به هنگام فراخوانی[۲] *edges method* استفاده کرد تا داده‌های یال را در کنار فهرست نتایج، نمایش دهد. توجه داشته باشید که اضافه کردن یالها بین گره‌هایی که هنوز وجود ندارد، موجب اخطار یا پیام خطا نمی‌شود، بلکه این گره‌ها را به‌طور خودکار ایجاد می‌کند.

```
In [11]: g.add_weighted_edges_from([(7, 8, 1.5), (9, 10,
         3.5)])
In [12]: g.edges(data = True)
```

[۱] List of Tuples of Nodes

[۲] Call

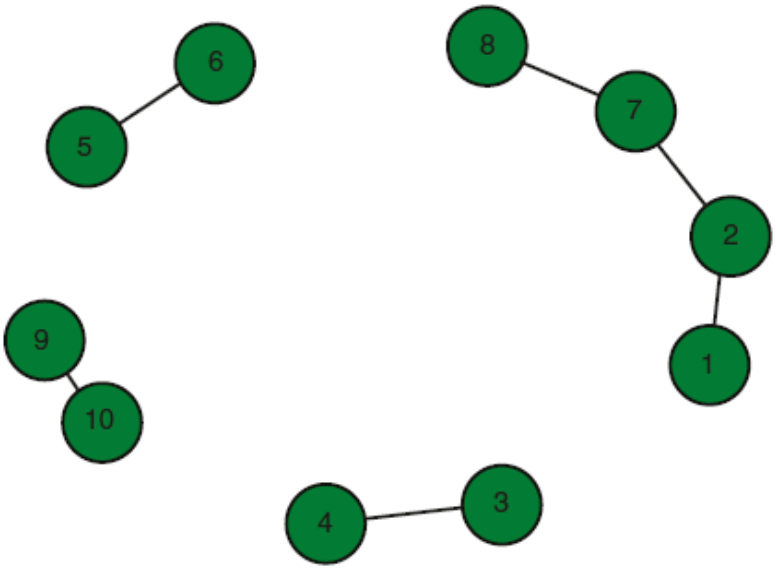
```
Out [12]: [(1, 2, {}),
          (3, 4, {}),
          (5, 6, {}),
          (7, 8, {'weight': 1.5}),
          (9, 10, {'weight': 3.5})]
```

برای اضافه کردن یالها با استفاده از امکان "بازکردن چندتایی یالها"[۱]، می‌توانیم دستور زیر را وارد کنیم:

```
In [13]: e = (2, 7)
In [14]: g.add_edge(*e)
```

همچنین می‌توانیم رنگ و اندازه گره‌ها را تغییر دهیم. برای رسم شبکه، باید دستور زیر را وارد کنیم:

In [15]: nx.draw_networkx(g, node_color='green', node_size=700)) (شکل ۱-۲)



شکل ۱-۲ گراف غیرجهتدار در *NetworkX*

[۱] Unpack Edge Tuples

۲-۷ ماتریسها

استفاده از ماتریس، راهی جایگزین برای نمایش و تلخیص داده‌های شبکه است. ماتریس عملاً آرایه‌هایی از عناصر است که حاوی اطلاعاتی مشابه اطلاعات گراف است. عملیات ماتریسی[۱] گسترده‌ای برای تعریف و محاسبه درتحلیل شبکه‌های اجتماعی (*SNA*) استفاده شده است.

عملیات مربوط به گراف و ماتریس بعنوان پایه بسیاری از مفاهیم تجزیه و تحلیل شبکه‌های اجتماعی استفاده شده‌اند. با این حال، ماتریسها برای محاسبات و تجزیه و تحلیل کامپیوتری مناسب‌ترند.

به‌طورکلی یک ماتریس مجموعه‌ای از سطرها و ستونها است. سلولهای ناشی از تقاطع سطری–ستونی[۲] ماتریس نشان می‌دهند چگونه عناصر در سطرها به آیتمها در ستونها نگاه می‌کنند. انواع معمول ماتریسها بشرح زیرند:

- **ماتریس ستونی[۳]:** ماتریسی که فقط یک ستون دارد یا ماتریسی تنها با مقادیر ورودی عمودی.
- **ماتریس سطری[۴]:** ماتریسی که فقط یک ردیف دارد یا ماتریسی تنها با مقادیر ورودی افقی.
- **ماتریس مربعی[۵]:** یک ماتریس که در آن تعداد سطرها برابر با تعداد ستونها است.
- **ماتریس همانی[۶]:** یک ماتریس مربعی است که در آن مقادیر قطر اصلی یک و مقادیر تمام عناصر دیگر صفر است.
- **ماتریس قطری[۷]:** یک ماتریس مربعی است که در آن مقادیر همه عناصر (به غیر از قطر اصلی) صفر است و مقادیر قطر اصلی لزوماً یک نیستند.
- **ماتریس متقارن[۸]:** یک ماتریس مربعی که با ترانهاده خود برابر است.

---

[۱] Matrix Operations
[۲] Column-Row Intersection
[۳] Column Matrix
[۴] Row Matrix
[۵] Square Matrix
[۶] Identity Matrix
[۷] Diagonal Matrix
[۸] Symmetric Matrix

- **ماتریس پادمتقارن**[۱]: یک ماتریس مربعی که برابر با منفی ترانهاده‌اش است.
- **ماتریس مثلثی**[۲]: یک ماتریس مربعی است که در آن تمام ضرایب زیر قطر اصلی صفر است.
- **ماتریس صفر**[۳]: یک ماتریس که تمام عناصر آن صفر است.

۲-۸ انواع ماتریس در شبکه‌های اجتماعی

سه نوع ماتریس برای نشان دادن داده‌ها در شبکه‌های اجتماعی استفاده می‌شود که عبارتند از:

۲-۸-۱ ماتریس مجاورت

ماتریس مجاورت ماتریسی است که سطرها و ستونهای آن گره‌های شبکه می‌باشند، بگونه‌ای که عنصر A_{ij} تعداد ارتباطات از سوی گره i به سمت گره j را نشان می‌دهد (این ماتریس برای گراف بدون جهت، یک ماتریس متقارن می‌شود). ماتریس مجاورت برای گرافهای غیر جهت‌دار، خواهد بود:

	A	*B*	*C*	*D*	*E*
A	۰	۱	۰	۰	۰
B	۱	۰	۰	۱	۰
C	۰	۰	۰	۱	۱
D	۱	۰	۱	۰	۰
E	۱	۰	۱	۰	۰

مقدار (۱) وجود ارتباط (یال) بین گره A و گره B را نشان می‌دهد درحالی‌که مقدار (۰) به این معنی است که چنین رابطه‌ای وجود ندارد.

ماتریس مجاورت برای گرافهای وزندار جهتدار:

	A	*B*	*C*	*D*	*E*

[۱] Skew-Symmetric Matrix

[۲] Triangle Matrix

[۳] Null Matrix

A	۰	۲	۰	۵	۴
B	۲	۰	۰	۱	۰
C	۰	۰	۰	۳	۴
D	۶	۲	۳	۰	۰
E	۵	۳	۴	۰	۰

از معایب اصلی ماتریسهای مجاورت این است که سلولهایی با مقدار صفر، به اندازه سایر سلولهای دیگر، حافظه اشغال می‌کنند. در حقیقت، بیش از ۹۰ درصد از سلولهای واقعی شبکه اجتماعی، ارزشی برابر با صفر دارند؛ بنابراین، استفاده از ماتریس مجاورت در عمل، به‌خصوص در مورد شبکه‌های بزرگ، بسیار دشوار است.

مثال

خوشبختانه، کتابخانه *NetworkX* ایجاد، اداره و مطالعه بسیاری از جنبه‌های شبکه‌های پیچیده و گرافها را آسان می‌کند. تأکید اصلی آن بر اجتناب از پیچیدگیهای بسیار در کار با الگوریتمهای گراف است. در مثال زیر نشان خواهیم داد که چگونه می‌توان ماتریس مجاورت پایه را در *NetworkX* ایجاد کرد:

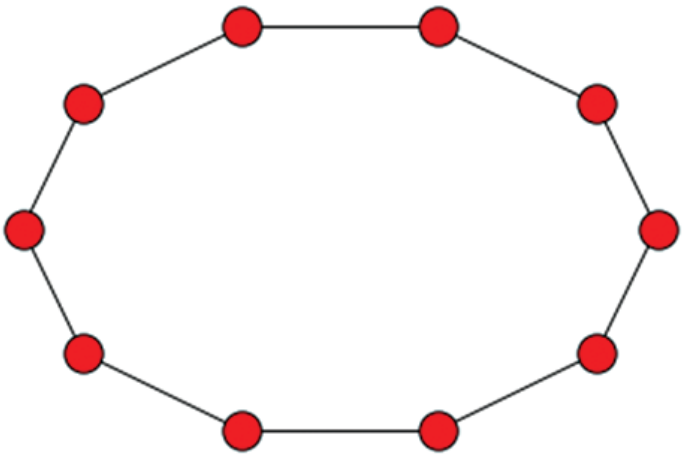
```
In:  import networkx as nx
     import 51atplotlib.pyplot as plt
     g = nx.cycle_graph(10)
     x = nx.adj_matrix(g)
     print(x.todense())
out:
     [[0 1 0 0 0 0 0 0 0 1]
     [1 0 1 0 0 0 0 0 0 0]
     [0 1 0 1 0 0 0 0 0 0]
     [0 0 1 0 1 0 0 0 0 0]
     [0 0 0 1 0 1 0 0 0 0]
     [0 0 0 0 1 0 1 0 0 0]
     [0 0 0 0 0 1 0 1 0 0]
     [0 0 0 0 0 0 1 0 1 0]
     [0 0 0 0 0 0 0 1 0 1]
     [1 0 0 0 0 0 0 0 1 0]]
```

راه دیگری نیز برای ساختن ماتریس مجاورت با ۱۰ گره وجود دارد که ایجاد فهرستی از یالها (زوج گره‌ها) است:

```
In:  n = 10
     adj_mat = [(i, (i+1)%n) for i in range(n)]
     adj_mat += [(i,(i+2)%n) for i in range(n)]
     g = nx.Graph(adj_mat)
```

می‌توانیم یا به‌طور صریح موقعیت گره‌ها را مشخص کنیم و یا الگوریتمی را تعریف کنیم که می‌تواند چیدمان موردنظر را به‌طور خودکار محاسبه کند. در مثال زیر، روش *draw_circular()* را فراخوانی می‌کنیم که گره‌ها را به‌صورت خطی بر روی یک دایره قرار می‌دهد. (شکل ۲-۲):

```
In:  nx.draw_networkx(g)
```



شکل ۲-۲ نمایش دایره‌ای ماتریس مجاورت

۲-۸-۲ ماتریس فهرست یال

فهرست یال[۱] بعنوان راه‌حلی برای مشکل کم‌پشتی یا تُنُک بودن ماتریس مجاورت ایجاد شده است. در این نوع گرافها، چیزی که مورد نیاز است فهرستی از رئوسی است که یالی به آنها برخورد دارد. با پیمایش ماتریس، رئوس گراف نیز مشخص می‌شوند، زیرا هر راس با حداقل یک یال برخورد می‌کند. بااین حال، در این نوع از نمایش داده، اندازه ماتریس به‌صورت خطی نسبت به تعداد یالها رشد می‌کند.

از	به	ارزش
A	*B*	۱
A	*D*	۴
A	*E*	۵
B	*A*	۳
C	*E*	۳
D	*A*	۵
E	*C*	۱

ماتریس فهرست یالها (یا به‌طور ساده: فهرست یالها) را در پایتون می‌توان با استفاده از دستور زیر بدست آورد.

```
edge_list = {{'A','B', 1}, {'A','D', 4}, {'A','E',5}...}
```

در ادامه مثال قبل، به‌جای توصیف گراف بعنوان مجموعه‌ای از گره‌ها، می‌خواهیم گراف را به‌صورت مجموعه‌ای از یالها تعریف نماییم که در آن یالها دارای ویژگیهای خاصی می‌باشند.

مثال

```
In:  nx.to_edgelist(g)
Out: [(1, 2, {}), (1, 4, {}), (2, 3, {}), (3, 4, {})]
```

[۱] Edge Lists

از آنجا که تاکنون برای یالها، در گرافی که ایجاد کرده‌ایم هیچ نوع ویژگی را مشخص نکرده‌ایم، لذا با آکُلادهای[۱] خالی (بصورت {}) در خروجی دستور مواجه شدیم. وجود آکُلادهای خالی در خروجی فوق، در واقع گویای فقدان ویژگی یالها (مثلاً مسافت، فرکانس و غیره) است.

۲-۸-۳ فهرست مجاورت

اگرچه "فهرست یالها" تا حدود زیادی مشکلات مربوط به کوچک بودن حافظه و پراکندگی داده‌ها را برطرف می‌نماید، اما به اندازه کافی برای جستجو و پیمایش گراف، سریع نیست. یک راه‌حل بهتر، استفاده از "فهرست مجاورت"[۲] است که شامل فهرستی از ارتباطات یا پیوندها می‌باشد. وجود عنصری همچون $i \to j$ در این فهرست، نشاندهنده یالی است که از گره i به گره j کشیده شده است. همچنین آنرا بصورت $i \to \{j_1.j_2.j_3 \ldots.\}$ نشان می‌دهند. با چنین نمایشی از داده‌ها، جستجو سریعتر می‌شود. همچنین اضافه یا حذف یالها/گره‌ها نیز آسان می‌گردد. مثالی از این دست:

از	یالها
A	*(B,1),(D,4),(E,5)*
...	

برای مشخص کردن فهرست مجاورت فوق در پایتون، می‌توان از دستور زیر استفاده نمود.

```
edge_list = {'A':{'B':1, 'D':4, 'E':5}, 'B':{'A':3},...}
```

مثال ۱

```
In: g = nx.Graph()
    g.add_edge(۱,۲)
    g.add_nodes_from([۳,۴])
    g.add_edge(۳,۴)
    g.add_edges_from([(2,3), (4,1)])
```

[۱] Curly Brackets

[۲] Adjacency List

حال بیایید گراف را رسم کنیم (شکل ۳-۲).

```
In:  nx.draw_networkx(g, node_color='orange', node_size=400)
Out:

In:  g.nodes()
Out: [1, 2, 3, 4]
In:  g.edges()
Out: [(1, 2), (1, 4), (2, 3), (3, 4)]
```

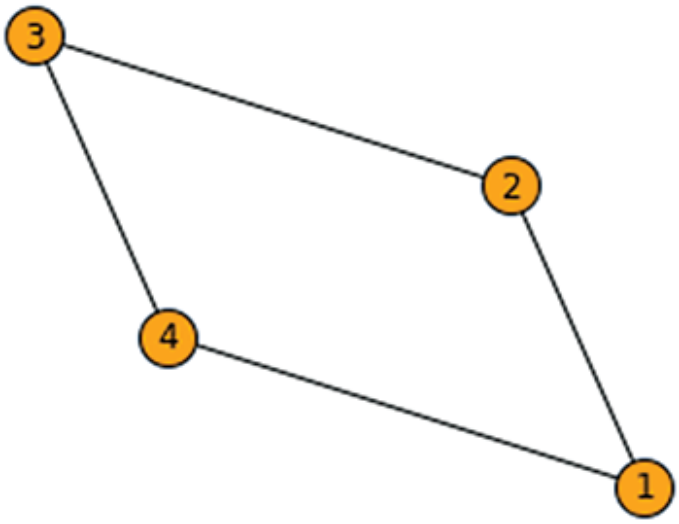
می‌توانیم این "گراف" را از طریق "فهرست مجاورت" آن تشریح کنیم؛ این کار با فهرست کردن همسایه‌های هر گره انجام می‌شود. به‌طور مثال همسایه‌های گره ۱، گره‌های [۲ و ۴] هستند و همسایه‌های گره ۲، گره‌های [۱ و ۳] و غیره.

```
In:  g.adjacency_list()
Out: [[2, 4], [1, 3], [2, 4], [1, 3]]
```

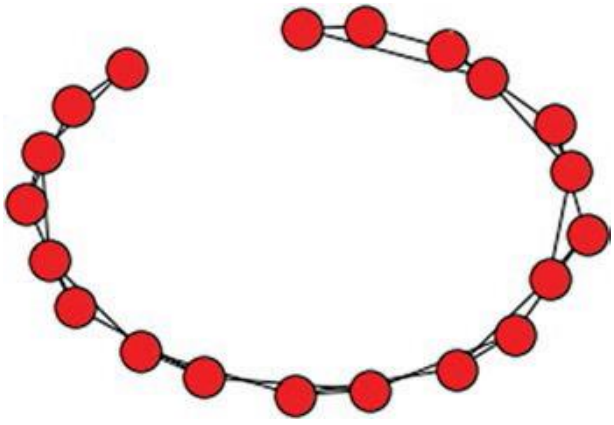
همچنین این "گراف" را می‌توان بعنوان "دیکشنری فهرستها" تعریف نمود، به‌طوری که **کلید دیکشنری**[۱]، در اینجا نام گره‌ها بوده و **ارزش دیکشنری**، عملاً فهرست مجاوران گره‌ها[۲] است. این روش نمایش، موجب آشکار شدن همسایگان گره‌ها نیز می‌شود.

[۱] Dictionary's Keys

[۲] Nodes' Adjacency Lists



شکل ۳-۲ گراف ساده با چهار گره



شکل ٤-۲ گراف نردبانی در *NetworkX*

```
In: nx.to_dict_of_lists(g)
Out: {1: [2, 4], 2: [1, 3], 3: [2, 4], 4: [1, 3]}
```

مثال ۲

می‌توانیم همچنین "فهرست مجاورت" یکی از شبکه‌های ایجادشده در *NetworkX* را با استفاده از کدهای زیر به دست آوریم (شکل ۴-۲).

```
In:  g = nx.ladder_graph(10)
     nx.draw_networkx(g)
Out:

In:  print(g.adjacency_list())
Out:
     [[1, 10], [0, 2, 11], [1, 3, 12], [2, 4, 13], [3, 5, 14],
     [4, 6, 15], [16, 5, 7], [8, 17, 6], [9, 18, 7], [8, 19],
     [0, 11], [1, 10, 12], [2, 11, 13], [3, 12, 14], [4, 13,
     15], [16, 5, 14], [17, 6, 15], [16, 18, 7], [8, 17, 19],
     [9, 18]]
```

یا اگر بخواهیم آنرا بعنوان "دیکشنری فهرستها" چاپ کنیم:

```
In:  print(nx.to_dict_of_lists(g))
Out:
     {0: [1, 10], 1: [0, 2, 11], 2: [1, 3, 12], 3: [2, 4, 13], 4: [3, 5,
     14], 5: [4, 6, 15], 6: [16, 5, 7], 7: [8, 17, 6], 8: [9, 18, 7], 9:
     [8, 19], 10: [0, 11], 11: [1, 10, 12], 12: [2, 11, 13], 13: [3, 12,
     14], 14: [4, 13, 15], 15: [16, 5, 14], 16: [17, 6, 15], 17: [16, 18,
     7], 18: [8, 17, 19], 19: [9, 18]}
```

۲-۸-٤ ماتریس *Numpy*

این نوع ماتریس مختص زبان برنامه‌نویسی پایتون هست. اگر رابطه‌ای بین یال i و j وجود داشته باشد، درایه مربوطه در این ماتریس یک و در غیر این صورت صفر است.

```
In:  nx.to_numpy_matrix(g)
Out: matrix([[ 0. 1. 0. 1.],
     [1. 0. 1. 0.],
```

```
      [0. 1. 0. 1.],
      [1. 0. 1. 0.]])
```

۲-۸-۵ ماتریس تُنُک

با توجه به اینکه معمولاً ماتریسها تُنُک هستند؛ یعنی تعداد یکهای موجود در آنها در مقایسه با تعداد صفرها اندک شمار است، می‌توان آنها را به شکل ماتریس *SciPy* نشان داد.

```
In:  print(nx.to_scipy_sparse_matrix(g))
Out: (0, 1) 1
     (0, 3) 1
     (1, 0) 1
     (1, 2) 1
     (2, 1) 1
     (2, 3) 1
     (3, 0) 1
     (3, 2) 1
```

۲-۹ عملیات پایه‌ای ماتریسها

بیایید نگاهی به عملیات پایه‌ای ماتریسها بیاندازیم.

- **دایره واژگان (یا بیان‌افزار)**[۱]: اندازه یک ماتریس با تعداد سطرها و ستونهای آن تعریف می‌شود. اگر تعداد سطرها و ستونهای یک ماتریس برابر باشد، آنگاه آن ماتریس مربعی و در غیر اینصورت مستطیلی خواهد بود. هر ورودی ماتریس یک سلول نامیده می‌شود. دریک ماتریس مربعی، عناصر یا درایه‌های موجود بر روی قطر اصلی، دارای شمارنده سطری و ستونی یکسانی هستند.

[۱] Vocabulary

- **ماتریس جایگشت**[۱]**:** ماتریس جایگشت، تنظیم مجدد سطرها و ستونها در یک ماتریس است. این کار در مواقعی انجام می‌شود که الگوهای ارتباطات[۲] شبکه مشخص نیستند لذا ما برای فهم بهتر الگوها، عملاً سطرها و ستونهای ماتریس را تغییر می‌دهیم. ماتریس جایگشت برای مطالعه گروههای منسجم، ساخت مدلهای بلوکی و ارزیابی تناسب مدلهای بلوک[۳] لازم است.

- **ترانهاده ماتریس**[۴]**:** ترانهاده یک ماتریس با تعویض سطرها و ستونها در ماتریس اصلی انجام می‌شود. "ماتریس‌گراف جهتدار"[۵] همیشه با ترانهاده خود یکسان نیست. چون "ماتریس-اجتماعی"[۶] مربوط به یک رابطه جهت‌دار، معمولاً متقارن نیست.

- **جمع ماتریسی**[۷]**:** منظور همان جمع دو ماتریس با اندازه (تعداد سطر و ستون) یکسان است. این کار با جمع عناصر در سلولهای متناظر دو ماتریس انجام می‌شود.

- **تفریق ماتریسی**[۸]**:** تفریق دو ماتریس با اندازه یکسان می‌تواند با محاسبه اختلاف بین عناصر در سلولهای متناظر دو ماتریس انجام شود.

- **ضرب ماتریسی**[۹]: منظور همان ضرب دو ماتریس هست. در مطالعه تحلیل شبکه‌های اجتماعی ضرب ماتریسی حیاتی و مهم است، زیرا جهت بررسی قدمها[۱۰] و قابلیت دسترسی[۱۱] مورداستفاده قرار می‌گیرد و همچنین پایه روابط مرکب[۱۲] در تحلیل جبر روابط است.

---

[۱] Matrix Permutation
[۲] Tie Patterns
[۳] Fitness of Block Models
[۴] Matrix Transpose
[۵] Digraph Matrix
[۶] Sociomatrix
[۷] Matrix Addition
[۸] Matrix Subtraction
[۹] Matrix Multiplication
[۱۰] Walks
[۱۱] Reachability
[۱۲] Compounding Relations

اگرچه ماتریسها در طول سالها به‌طور گسترده برای تجسم شبکه‌ها مورد استفاده قرار گرفتند، اما با توجه به توان محدود ماتریسها برای پذیرش ورودیها، در مقایسه با تعداد زیاد ورودیهای شبکه‌های آنلاین، امروزه به‌ندرت برای این منظور به کار می‌روند.

۲-۱۰ تجسم داده‌ها

تجسم یا مصورسازی داده برای آنالیز شبکه بسیار مفید است چون به کاربران کمک می‌کند ویژگیهای ساختاری مهم شبکه را به‌طور بصری تشخیص دهند.

گراف بهترین راه برای تجسم بخشیدن به داده‌های شبکه به‌خصوص در زمانی است که اندازه شبکه بسیار بزرگ است. همچنین درختان (یعنی گرافهای بدون جهت که در آن، هر دو راس از طریق یک مسیر ساده بهم متصل می‌شوند) برای تجسم داده‌ها و به‌خصوص زمانی که مجموعه داده‌ها کوچک است، مورد استفاده قرار می‌گیرند. درختان می‌توانند بصورت چندگانه[۱]، ریشه‌دار[۲]، برچسبدار[۳]، بازگشتی[۴]، جهتدار[۵]، آزاد[۶]، دودویی[۷] یا سه‌تایی[۸] باشند.

ماتریسها عملاً آرایه‌هایی از عناصر (شامل سطرها و ستونها) هستند که در آن، تقاطع بین یک سطر و یک ستون، سلول نامیده می‌شود. انواع متداول ماتریسها شامل ماتریسهای ستونی، ماتریسهای مربعی، ماتریسهای همانی، ماتریسهای قطری، ماتریسهای متقارن، ماتریسهای پادمتقارن، ماتریسهای مثلثی و ماتریس صفر هستند. انواع متداول ماتریس در مطالعه شبکه‌های اجتماعی، همانا ماتریس مجاورت، ماتریسهای فهرست یال و فهرست مجاورت هستند.

---

[۱] Polytree
[۲] Rooted
[۳] Labeled
[۴] Recursive
[۵] Directed
[۶] Free
[۷] Binary
[۸] Ternary

دیگر ابزارهای نمایشی یا تصویرسازی شامل نقشه‌ها و رویکردهای ترکیبی هستند؛ از جمله آنها، فنون تجسم چندبعدی[۱] بوده که در کنار گرافها و نقشه‌ها استفاده می‌شوند.

[۱] Multidimensional Visualization Techniques

فصل سوم

تئوری گراف

در این فصل ویژگیهای اصلی "نظریه گراف"، مبانی ریاضی کاربردها و خواص آنها را بحث می‌کنیم. موضوع گرافها در ابتدا با مطالعه بازیهای شانسی[1] شروع شد و سپس گسترش فراوانی یافت. بنابراین موضوعاتی مانند ریشه‌های نظریه گراف، مبانی گراف، گونه‌های مختلف گراف، پیمایشهای گراف و انواع عملیات در گراف را بررسی می‌کنیم.

[1] Games of Chance

۳-۱ ریشه‌های نظریه گراف

اولین مقاله در مورد نظریه گراف در سال ۱۷۳۶ توسط لئونارد اویلر[۱] منتشر شد. این مقاله، نتیجه کار اویلر بود، وقتی او مسئله "هفت پل" درشهر کونیگسبرگ (اکنون کالینینگراد، روسیه) را مطرح کرد. اویلر مسئله چگونگی پیاده‌روی در شهر، بطوری که از هر منطقهٔ شهری و از هر پل، تنها یک بار بتوان گذر کرد را بررسی نمود. کونیگسبرگ یک شهر در پروس[۲] بود که در آن زمان رودخانه پریگل از میان آن عبور می‌کرد و دو جزیره را ایجاد کرده بود. همانطور که در شکل ۱-۳ مشاهده می‌شود، شهر و جزایر مذکور توسط هفت پل، با هم ارتباط داشتند.

شکل ۱-۳ سرآغاز طرح نظریه گراف

اویلر یک برداشت انتزاعی از این مسئله را با حذف حقایق غیرضروری و تمرکز بر روی زمینها و پلهای ارتباطی آنها، فرمول‌بندی کرد. او متوجه شد که انتخاب مسیر در داخل هر منطقه بی‌اهمیت است و تنها چیزی که مهم است ترتیبی است که بر اساس آن از پلها عبور می‌شود. او کشف کرد که این مسئله هیچ راه حلی نداشته و باید پل هشتم ساخته شود. گرافهای اویلری[۳] به همین دلیل به نام او نامگذاری شده‌اند. کار او، که تئوری گراف را بوجود آورد، رابطه زیادی با تحلیل شبکه‌های اجتماعی دارد.

---

[۱] Leonhard Euler

[۲] پروس (Prussia) نام سرزمینی است که امروزه بخش شرقی آلمان و بخش غربی لهستان را در بر دارد.

[۳] Eulerian

فریمن[۱] (۲۰۰۴) با پژوهش و رصد، اولین استفاده از تصاویر بصری را به جاکوب مورنو[۲] (۱۹۳۲-۱۹۳۴) نسبت داد، روانشناسی که از جامعه‌نما بعنوان روشی برای نشان دادن ساختار بین فردی (در میان یک گروه از مردم) استفاده کرد. (از جامعه‌نما گاهی بعنوان دیاگرام گره-پیوند یاد می‌کنند).

استفاده از نظریه گراف در مطالعه روابط اجتماعی توسط گروهی از ریاضیدانان و روانشناسان در دهه ۱۹۴۰ آغاز شد. نظریه گراف به محققان این توانایی را می‌دهد تا قضیه‌ها را اثبات کرده و گزاره‌های قابل آزمایش را استنتاج کنند. براساس این نظریه، شبکه اجتماعی را می‌توان بعنوان مدلی از یک سیستم اجتماعی متشکل از کنشگران و روابط میان آنها نشان داد.

عملاً یک گراف، نوعی ابزار ریاضی است که رابطه بین اجزا را توصیف می‌کند. در واقع گراف، ساختاری برای مدلسازی اطلاعات است. این ساختار شامل گره و یال است؛ گره‌هایی که اشیا را نشان داده و یالهایی که یک گره را به دیگری مرتبط می‌کنند. "تئوری گراف" ابزار مورد نیاز برای توصیف و تجسم آن دسته از سازه‌های اجتماعی[۳]را فراهم می‌کند که شامل سه کنشگر یا بیشتر می‌شوند. این نظریه منجر به شیوه جدیدی برای عینیت بخشیدن به سازه‌های اجتماعی شده، که در آن "یالها " هم نمایانگر اقدامات انسانی بوده و هم نمایشگر زمینه (بافتار)[۴] اقدامات انسانی هستند.

گرافها تبدیل به یک روشِ فراگیر در دنیای محاسبات شده‌اند. اینکه اصطلاحاً سر و کله آنها در همه جا پیدا می‌شود بدین دلیل است که آنها می‌توانند انواع مختلف روابط دنیای واقعی را نشان دهند. روابطی همچون: دوستان در یک شبکه اجتماعی، صفحات وب سایت، سلولهای یک شبکه عصبی و غیره. آنها همچنین ساختارهای داده استاندارد[۵] در مطالعه علم کامپیوتر هستند.

نظریه گراف در مطالعه شبکه‌های اجتماعی به دلایل مختلف مفید بوده است:

- اولاً، این نظریه، واژگانی ارائه می‌دهد که قابل استفاده برای نامگذاری و نمایش بسیاری از خواص ساختارهای اجتماعی است.

---

[۱] Freeman

[۲] Jacob Moreno

[۳] Social Compositions

[۴] Context

[۵] Standard Data Structures

- ثانیاً، امکان استفاده بسیاری از عملیات و ایده‌های ریاضی را فراهم می‌کند که به کمی‌سازی و اندازه‌گیری شماری از خواص ساختارهای اجتماعی کمک می‌کند.

- ثالثاً، نظریه گراف، راهی برای اثبات قضیه‌های گرافها و نیز نمایشِ ساختارهای اجتماعی ارائه می‌دهد.

گرافها در تحقیقات مهم هستند، زیرا اولاً آنها یک نمایش انتزاعی از زندگی واقعی ارائه می‌دهند، بعلاوه جریان اطلاعاتی موجود را نمایش داده و بموازات آن، روابط را به صراحت نشان می‌دهند. گرافها همچنین محاسبات را برای مجموعه داده‌های بزرگ امکانپذیر می‌کنند و در همین راستا، امکان محاسبات را به صورت محلی برای بخش خاصی از داده‌ها، از طریق پیمایش[۱] کوچکی فراهم می‌آورند.
بسیاری از مسائل عملی را می‌توان به صورت گراف نشان داد. بعنوان مثال، از گرافها برای مدلسازی مسائل یا موقعیتهایی در حوزه فیزیک، زیست‌شناسی، روانشناسی و علوم رایانه استفاده شده است. در علوم کامپیوتر، گرافها برای نشان دادن شبکه‌های ارتباطات، سازماندهی داده‌ها، دستگاههای محاسباتی و جریان محاسبات استفاده می‌شوند. ساختار پیوند وب‌سایتها را همچنین می‌توان بعنوان گراف در نظرگرفت.
الگوریتمهای گراف (بعنوان مثال، بلمن فورد[۲]، دایکسترا[۳]، فورد- فالکرسون[۴]، کروسکال[۵]، نزدیکترین مجاورت[۶]، پیمایش عمقی یا اول-عمق[۷] و پیمایش سطحی یا اول-سطح[۸]) برای حل مسائل مربوط به پیمایش گراف[۹]، رنگ‌آمیزی گراف[۱۰]، مولفه‌های همبندی[۱]، کوتاهترین مسیرها، مسیرهای همیلتون[۲]، مسیرهای اویلری[۳] و مسئله فروشندۀ دوره‌گرد[۴] طراحی شده‌اند.

---

[۱] Traversals
[۲] Bellman-Ford
[۳] Dijkstra
[۴] Ford-Fulkerson
[۵] Kruskai
[۶] Nearest Neighbor
[۷] Depth-First Traversal
[۸] Breadth-First Traversal
[۹] Graph Traversals
[۱۰] Graph Coloring

با در نظر داشتن این موضوع، گروه مسائل مربوط به مطالعه گراف، به چهار دسته طبقه‌بندی می‌شوند:

۱) **موجودیت:** مسائلی که بدنبال آنند تا مشخص کنند که آیا یک مسیر، یک راس یا یک مجموعه وجود دارد. حالت ویژه این مسائل، زمانی است که محدودیتی نیز وجود داشته باشد.

۲) **ساخت و ساز:** مسائلی که درصدد جواب به این سوالند که با توجه به مجموعه‌ای از مسیرها و راسها و نیز محدودیتهای موجود، چگونه باید یک گراف را ساخت؟

۳) **شمارش:** مسائلی که سعی دارند با توجه به مجموعه‌ای از محدودیتها، تعیین کنند کلاً چند راس و یال وجود دارد.

۴) **بهینه‌سازی:** مسائلی که بدنبال یافتن کوتاهترین مسیر بین دو گره هستند.

۳-۲ مبانی گراف

یک گراف مجموعه‌ای از نقاط و خطوطی است که آنها را متصل می‌کند. در واقع راهی برای نشان دادن روابط بین مجموعه‌ای از اشیاست.

هر نقطه را یک راس و هر خط را یک یال می‌نامند. یک گراف G با مجموعه‌ای از راسها X و مجموعه‌ای از یالها E، به صورت $G=(X,E)$ نشان داده می‌شود (شکل ۲-۳).

فرض کنیم داریم که

$$G = (X,E) \quad x,y \in X$$

فاصله از x تا y که بصورت $d(x, y)$ نمایش داده می‌شود عملاً طول کوتاهترین مسیر (x, y) است. اگر چنین مسیری در G وجود نداشته باشد، $d(x.y) = \infty$ است. در این مورد، G منفصل است و x و y در دو قسمت مختلف شبکه قرار دارند. قطر G که بصورت $diam(G)$ نشان داده می‌شود، بیشترین فاصله بین رئوس یا $max_{x,y \in X}\, d(x,y)$، است.

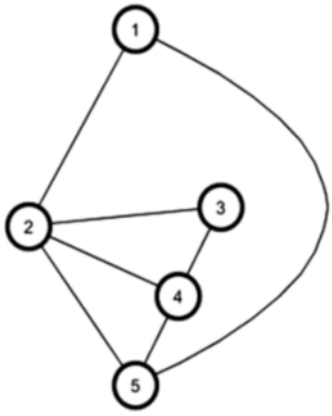
---

[۱] Connected Components,

[۲] Hamiltonian Paths

[۳] Eulerian Paths

[۴] Traveling Salesman Problem



گره‌ها: ۱، ۲، ۳، ٤، ٥
پیوندها: ۱<->۲،
۱<->۳، ۱<->٥،
۲<->۳، ۲<->٤،
۲<->٥، ۳<->٤،
۳<->٥، ٤<->٥

شکل ۲-۳ گراف ساده با پنج گره و هفت یال

مفهوم سیر یا گشت[۱] به یک دنباله متناوب از راسها و یالها اشاره دارد که در آن هر یال، راس قبلی و راس بعدی این توالی را بهم متصل می‌کند. به عبارت ساده؛ گشت، فهرستی از پیوندها است که به صورت متوالی به هم وصل شده‌اند تا تشکیل یک مسیر پیوسته بدهند. سیر یا گشت با یک راس شروع می‌شود و به یک راس خاتمه می‌یابد. یک گشت می‌تواند یک دنباله[۲]، مسیر[۳] یا چرخه (دور)[۴] باشد:

- یک دنباله، گشتی است که از هر یال بیش از یکبار عبور نمی‌کند.
- یک مسیر، گشتی است که از هر گره بیش از یکبار عبور نشود.
- یک چرخه یا دور، گشتی است که از یک گره شروع شده و سپس به همان گره پایان یافته و در مسیر خود از هر گره فقط یکبار عبور می‌کند.

اصطلاحات متعدد دیگری نیز، وجود دارند که در بحثها به ما کمک می‌کنند. حلقه[۵] یک یال است که یک راس را به خودش متصل می‌کند. راسهای بدون همسایه (درجه = ۰) راسهای مجزا یا ایزوله[۶] نامیده می‌شوند. دو راس را می‌توان با بیش از یک یال بهم متصل کرد. چنین یالهایی بعنوان یالهای موازی یا

---

۱ Walk
۲ Trail
۳ Path
۴ Cycle
۵ Loop
۶ Isolated

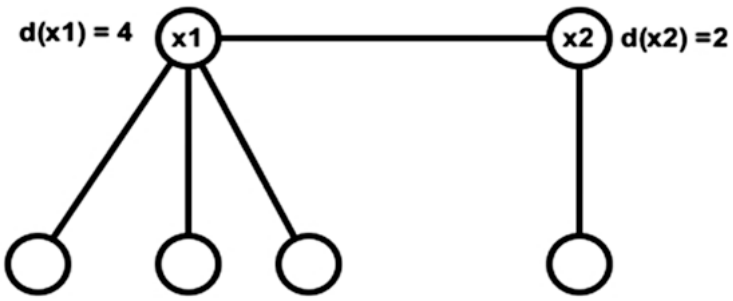
چندگانه نامیده می‌شود. یک زوج مرتب از راسها بعنوان قوس یا کمان شناخته می‌شود. اگر (x,y) یک قوس باشد، x بعنوان راس آغازین و y بعنوان راس پایانی شناخته می‌شود.

۳-۳ راسها

رئوس (یا گره‌ها) عناصر اصلی همه گرافها هستند. آنها ویژگیهای زیر را دارند:

- یک گراف $G=(X.E)$ و یک راس $x\epsilon X$ را در نظر بگیرید: حذف x از G به معنای حذف x از مجموعه X و حذف همه یالهایی است که حاوی x هستند. با این حال، حذف یک یال آسانتر از حذف یک راس است زیرا برای حذف یک یال تنها نیاز است که آن یال از فهرست یالها حذف شود.

- برای یک راس مشخص x، جمع تعداد تمام راسهای مجاور آن "درجه" نامیده می‌شود و با $d(x)$ مشخص می‌شود. حداکثر درجه در میان راسها بعنوان بالاترین درجهٔ G شناخته می‌شود. بعضاً راسهای مجاور را همسایه نامند و از مجموعه همسایگانِ یک راس فرضی x بعنوان همسایگی x یاد کرده که بصورت $N(x)$ نشان داده می‌شود. همچنین مجموعه‌ای از یالهای مربوط به یک راس x با $E(x)$ مشخص می‌شود. (شکل ۳-۳)



شکل ۳-۳ درجه رئوس

- درجه یک راس، تعداد یالهایی است که از آن راس عبور می‌کند. یک راس ایزوله، درجه صفر دارد، یعنی راسی است که نقطه پایانی هیچ یالی نیست. یک راس برگ[۱] عملاً راسی با درجه یک است.

- به دو راس که توسط یک یال بهم متصل می‌شوند، نقاط پایانی[۲] آن یال می‌گویند. یال بعنوان تلاقی میان[۳] رئوس توصیف شده است. یک مجاورت ساده[۴] بین دو راس بدین معنی است که دقیقاً یک یال بین آنها وجود دارد.

- در یک گراف جهتدار، "معیار درجه خروجی"[۵] تعداد یالهای خروجی است در حالیکه "معیار درجه ورودی"[۶] تعداد یالهای ورودی است. یک راس منبع، راسی با درجه ورودی صفر است در حالیکه یک راس فروچاله[۷]، یک راس با درجه خروجی صفر است.

- یک راس برشی[۸]، راسی است که اگر حذف شود، تعداد مولفه‌های شبکه افزایش می‌یابد. جداکننده رئوس[۹] مجموعه‌ای از راسها هستند که اگر حذف شوند، گراف به مولفه‌های کوچکتر تجزیه می‌شود.

- یک راس ارزشدار یا راس برچسبدار[۱۰]، راسی است که با یک مقدار همراه شده است بگونه‌ای که عملاً اطلاعات بیشتری را به آن راس اضافه می‌کند.

---

[۱] Leaf Vertex

[۲] Endpoint

[۳] Incident

[۴] Simple Adjacency

[۵] Out-Degree

[۶] In-Degree

[۷] Sink

برای واژه Sink معادلهایی همچون چاه، حفره، چاهک، فروچاله، فروکش، فرونشستن، گودال، گره پایانی و امثال آن وجود دارد که نشانه عمل "دریافت" در مقابل عمل "ارسال" است.

[۸] Cut-Vertex

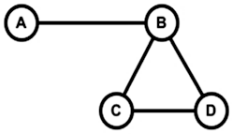
[۹] Vertex-Separator

[۱۰] Labeled Vertex

۳-٤ انواع گراف

در نظریه گراف چندین نوع مختلف گراف وجود دارد که به منظور نمایش ارتباط بین گره‌ها مطرح شده‌اند. شایعترین آنها عبارتند از:

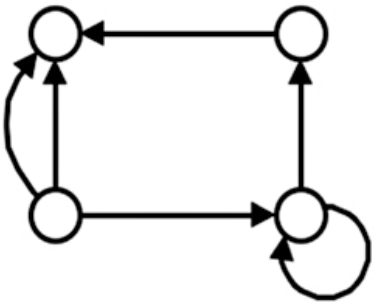
۱) **گراف بدون جهت:** گراف بدون جهت، یک گراف با یالهای بدون جهت است. این نوع گراف برای نشان دادن پیوندهای متقارن بکار می‌رود. بعنوان مثال، دو نفر (بمثابه دو گره در یک شبکه) را در نظر بگیرید که در جلسه‌ای با یکدیگر دست بدهند (بزبان گراف، به یکدیگر متصل شوند) آنگاه چنین رابطه‌ای قاعدتاً متقابل است (بزبان گراف، رابطه بدون جهت است). دوستان فیسبوکی و ارتباطات لینکدینی نیازمند تایید متقابلند، که براساس مقررات این دو سیستم، لازم الاجراست. برای مشابه‌سازی این الزام با نحوه ارتباطات دنیای واقعی در نظر داشته باشید که به هنگام دست دادن دو نفر، وقتی فرد A با فرد B دست می‌دهد (یعنی دستش را می‌گیرد و تکان می‌دهد)، عملاً شخص Bنیز با شخص A دست می‌دهد (دستش را گرفته و تکان می‌دهد). گرافهای بدون‌جهت را می‌توان به آسانی به گرافهای جهتدار، با جهتدار کردن یالها، تبدیل کرد (شکل ۴-۳).



شکل ٤-۳ گراف بدون جهت

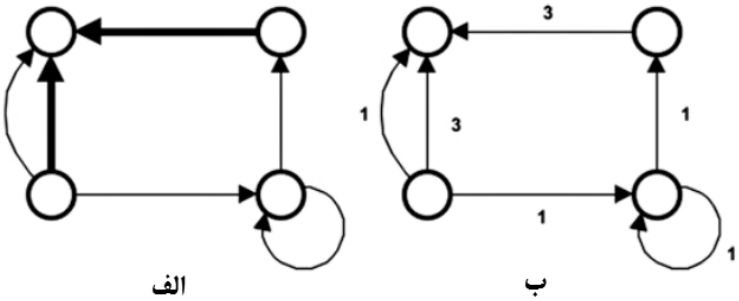
۲) **گراف جهتدار:** گراف جهتدار که دایگراف[۱] نیز نامیده می‌شود، گرافی است که حاوی یالهای جهتدار است. با یک مثال بهتر می‌توانیم این مطلب را درک کنیم: آلیس، باب را می‌شناسد، اما عکس آن درست نیست. در مورد روابطی همچون رابطه معلم-دانشجو و یا رئیس-کارمند چنین فرض می‌شود که این روابط نامتقارن است. نمایش رابطه آلیس و باب با استفاده از نظریه گراف، بدینگونه است که باید یک یال جهتدار از باب تا آلیس ترسیم کرد. بعنوان مثال دیگری از این دست می‌توان به شبکه‌های بی‌سیم اشاره کرد که اگر ارسال پیامها تنها در یک جهت ممکن باشد، آنگاه پیوند میان دو گره متفاوت، عملاً نامتقارن است. (شکل ۵-۳)

[۱] Digraph



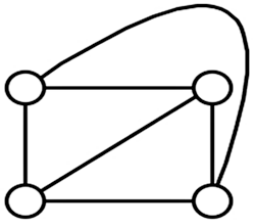
شکل ۵-۳ گراف جهتدار

۳) **گراف وزندار:** گراف وزندار، گرافی است که بر روی یالهای آن مقادیر و یا ارزشهایی بعنوان وزن وجود دارد. اوزان روی یالها بیان کننده یک مفهوم مرتبط با پیوند مانند هزینه، طول، ظرفیت، شباهت، فاصله و غیره هستند، که بستگی به استفاده‌ای دارد که از آن گراف می‌شود. بعنوان مثال، هنگام مدلسازی یک شبکه راه‌آهن بعنوان یک گراف، ایستگاههای راه‌آهن بعنوان راس نمایش داده می‌شوند، در حالی که یالها، ایستگاههای مجاور را بهم متصل می‌کنند. وزنها را می‌توان در اینجا به یالها اضافه کرد تا فاصله بین هر دو ایستگاه را بیان کنند. وزن گراف از طریق جمع اوزان تمامی یالها محاسبه می‌شود. یکی دیگر از نمونه‌های مفید، فراوانی یا فرکانس ارتباط در یک رابطه است که به طور دقیقی بیانگر محتوای هیجانی و میزان نفوذ رابطه است. فرکانس پایین ارتباطات بین دو نفر (که همچنین بعنوان روابط ضعیف شناخته می‌شود) نیاز به زمان یا انرژی کمی برای نگهداری دارد. این بدان معنی است که آنها در آنچه دریافت می‌کنند و زمانی که دریافت می‌کنند، ناهمسو و ناهماهنگ هستند. اگر چه چنین مدل ارتباطی می‌تواند در پاره‌ای از اوقات قابل توجه باشد؛ زیرا اطلاعاتی که بین دو گره عبور می‌کند، معمولاً جدید است و از نقطه نظر دیگری مطرح شده است (شکل ۶-۳).



شکل ٦-٣ دو راه برای نشان دادن قدرت یک رابطه

۴) **گراف مسطح[۱]:** گراف مسطح یا تخت، گرافی در یک سطح دو بعدی است بدون هیچ تقاطعی میان اتصالات (یالها یکدیگر را فقط در راسها قطع می‌کنند). مسطحی یا تختی[۲] گاه یک موضوع قابل توجه است. بعنوان مثال، هنگام طراحی شبکه‌های حمل و نقل، یک گراف مسطح به این معنی است که نیازی نیست که پلها یا تونلها را در طراحی قرار دهیم. مثال دیگری از این دست، طراحی مدارهای الکتریکی است، که تراشه‌ها را متصل می‌کنند. چنین کاری نیازمند سیمهایی است که اجزای مختلف را متصل کنند اما نباید از روی یکدیگر عبور کنند. یکی از راههای رسیدن به این هدف، قرار دادن تراشه‌ها در لایه‌های مختلف است که هر لایه، در این مورد، یک گراف مسطح است (شکل ۷-۳).



شکل ۷-۳ گراف مسطح

[۱] Planar Graph

[۲] Planarity

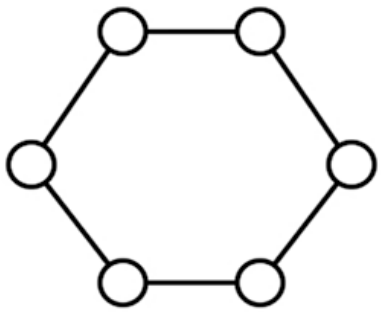
۵) **گراف متعامد[۱]:** گراف متعامد گرافی است که دارای خطوط افقی و عمودی می‌باشد.

۶) **گراف مشبک[۲]:** گرافی است که در آن راسها و یالها بر روی یک توری دو بعدی[۳] قرار می‌گیرند.

چند نوع گراف دیگر نیز وجود دارد که داده‌ها را به صورت متفاوتی نمایش می‌دهند:

۱) **گراف ساده:** یک گراف ساده یک گراف بدون جهت است که بدون حلقه و بدون وجود چند یال بین هر دو راس می‌باشد (بدون جهت، بدون حلقه و بدون چند یال). هر یال یک جفت راس متمایز را متصل می‌کند.

۲) **گراف منظم[۴]:** یک گراف منظم (یا یکنواخت) گرافی است که در آن تمام راسها به تعداد برابر همسایه داشته باشند، که همان درجه هر راس است. (درجه رئوس با هم برابر است) (شکل ۸-۳).



شکل ۸-۳ گراف منظم

۳) **گراف کامل[۵]:** یک گراف که در آن هر جفت گره، بهم متصل می‌باشند (شکل ۹-۳) و اغلب بدین صورت نوشته می‌شود:

K1, K2, K3, ...

---

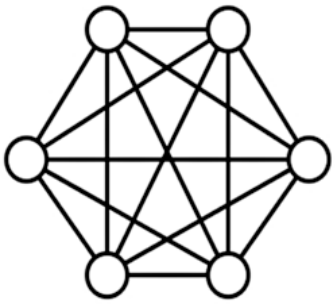
۱ Orthogonal Graph

۲ Grid-Based Graph

۳ Two-Dimensional Grid

۴ Regular Graph

۵ Complete Graph



شکل ۹-۳ گراف کامل

۴) **گراف مختلط[۱]:** یک گراف مختلط G گرافی است که در آن برخی از یالها جهت داشته و برخی دیگر بدون جهت هستند. گراف جهتدار و گراف بدون جهت حالت خاص این نوع گراف هستند.

۵) **گراف چندگانه[۲]:** اصطلاح گراف چندگانه یا چندگرافی برای اشاره به گرافهایی است که در آنها از یالهای چندگانه (و گاهی اوقات حلقه‌ها) استفاده می‌شود.

۶) **گراف نیمه یال یا سست یال[۳]**: نوعی گراف است که تنها یک انتها دارد که به آن گراف نیمه یال یا بدون انتها (یالهای سست) می‌گویند. مثالهایی از این دست، گرافهای علامت‌دار[۴] و گرافهای سوگیرانه[۵] هستند.

۷) **گرافهای متناهی و نامتناهی[۶]:** یک گراف متناهی، گرافی همانند $G= (V,E)$ است که در آن V و E مجموعه‌های متناهی هستند. یک گراف نامتناهی گرافی با مجموعه بی‌نهایت یا نامتناهی از راسها، یالها یا هر دو است.

---

۱ Mixed Graph

۲ Multigraph

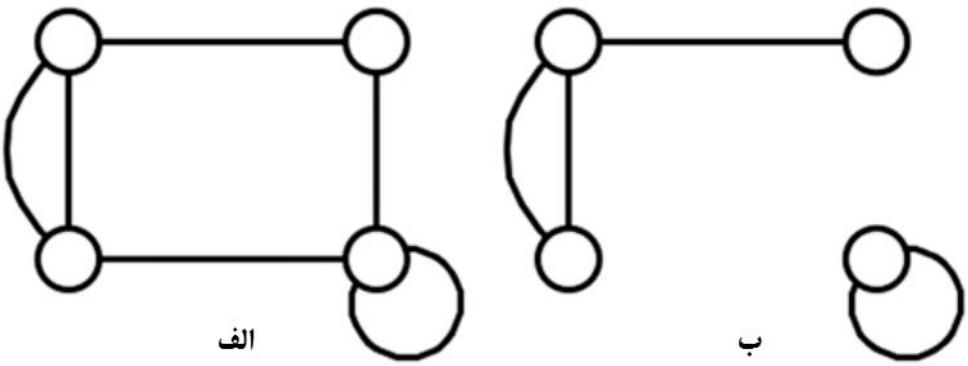
۳ Half-Edges, Loose Edges Graph

۴ Signed Graph

۵ Biased Graph

۶ Finite and Infinite Graphs

۸) **گراف متصل و منفصل[۱]:** اگر هر جفت راس مشخص در یک گراف متصل باشند، گراف متصل (همبند) نامیده می‌شود؛ در غیر این صورت، منفصل (ناهمبند) نامیده می‌شود (شکل ۱۰-۳).



شکل ۱۰-۳ الف: گراف متصل ب: گراف منفصل

۹) **گراف با k راس متصل[۲]**: یک گراف k-راس متصل یا k-یال متصل نامیده می‌شود، در صورتی که هیچ (k-1) راسی (یا یالی) وجود نداشته باشد که گراف را قطع کند. یک گراف با k-راس متصل اغلب، k-اتصالی[۳] یا k-همبند نامیده می‌شود.

۱۰) **گراف همبند ضعیف و قوی[۴]:** یک گراف جهتدار یک گراف همبند ضعیف است اگر با جایگزینی تمام یالهای جهتدار آن با یالهای بدون جهت، یک گراف همبند (بدون جهت) تولید شود. یک گراف به شدت همبند (یا قوی) است، اگر برای هر جفت راس u و v، یک مسیر جهتدار از u به v و یک مسیر جهتدار از v به u وجود داشته باشد.

---

[۱] Connected and Disconnected Graph

[۲] K-Vertex-Connected Graph

[۳] K-Connected

[۴] Weakly and Strongly Connected Graph

۳-۵ پیمایشهای گراف

اغلب ابزارهای تجزیه و تحلیل شبکه، کار خود را از طریق تکرار یک رویه محاسباتی بر روی گراف (یعنی بر روی گره‌ها و یالهای آن) و محاسبه معیارهای مشخصی انجام می‌دهند. هدف این رویه محاسباتی ممکن است پیدا کردن کوتاهترین مسیر بین دو نقطه، یافتن خوشه‌های مرتبط و یا محاسبه گره و یالهای بینابینی باشد. در برخی موارد، هدف این است که کل شبکه پیمایش شود تا یا شبکه بطورکلی شناسایی شده و یا داده‌های خاصی از روی شبکه نمونه‌برداری شود. در هر حال، تمام این تکنیکها نیاز به اجرای یکی از الگوریتمهای پیمایشی دارند که یا برای یافتن کوتاهترین مسیر بین نقاط گراف طراحی شده و یا برای گذر بر روی کل گراف بوده و سعی در درک ساختار آن دارند.

۳-۵-۱ پیمایش اول-عمق

پیمایش اول-عمق یا الگوریتم جستجوی عمقی (*DFS*)، یک تکنیک جستجوی ناآگاهانه است که تا زمانی که هدف خود را پیدا کند، به طور سیستماتیک از روی گره‌ها عبور می‌کند (شکل ۱۱-۳). این الگوریتم یک فرزندِ فرزند در گراف را می‌پیماید و سپس به هر یک از خواهران و برادران آن گره (گره‌های همسایه یا مجاورش) برمی‌گردد که نهایتاً یک درخت فراگیر یا پوشا[۱] از گره‌هایی که بازدید شده‌اند، تولید می‌شود. یالهایی که به گره‌های تازه کشف‌شده منتهی می‌شوند، بعنوان یالهای پیشروی یا اکتشافی[۲] نگهداری شده؛ و یالهایی که برای عقب نشینی یا عقبگرد مورد استفاده قرار می‌گیرند، بعنوان یالهای پسروی یا بازگشتی[۳] نگهداری می‌شوند. این دو نوع یال به همراه گره‌های بازدید شده، درختی را شکل می‌دهند که به "درخت پوشا" موسوم است. همچنین نسخهٔ دیگری از الگوریتم استاندارد وجود دارد که دارای یک ذخیره یا پشته[۴] برای نگهداری گره‌های بازدید شده است.

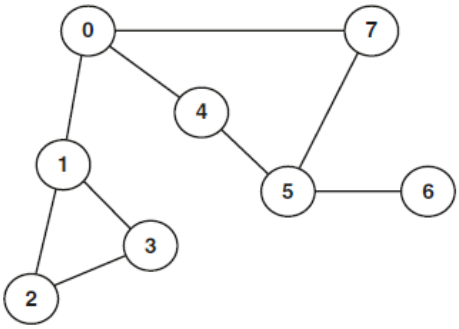
---

[۱] Spanning Tree

[۲] Discovery Edges

[۳] Back Edges

[۴] Stack



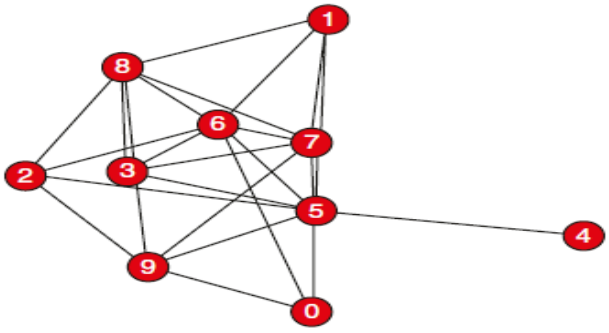
شکل ۱۱-۳ پیمایش جستجوی اول-عمق

مثال

در این مثال الگوریتم *DFS* با استفاده از یک نمودار ساده در *NetworkX* بررسی می‌شود. این مثال بر اساس کار *D. Eppstein* (جولای ۲۰۰۴) است (شکل ۱۲-۳).

```
In:  g = nx.barabasi_albert_graph(10, 5)

     print(nx.info(g))
In:  edges = nx.dfs_edges(g, 0)
```

در *DFS* باید گره منبع را که در واقع نقطه شروع کار است، مشخص نمود. در این مثال گره شماره صفر برای شروع در نظر گرفته شده است.



شکل ۱۲-۳ گراف باراباشی- آلبرت[۱] در NetworkX

[۱] Barabasi-Albert

```
In:  print(edges)
Out: <generator object dfs_edges at 0x000000000A742318>
```

درخواست پرینت یالها (*edge object*) به تنهایی، موجب فعال شدن "تولیدکننده شی"[۱] می‌شود، که یک سازه پایتون برای یک کار دم‌دستی و اصطلاحاً ارزیابی کاهلانه[۲] است و لیکن، می‌توانیم با استفاده از تابع فهرست (*list function*) از پایتون به صورت زیر استفاده کنیم:

```
In:  print(list(edges))
Out: [(0, 9), (9, 8), (8, 1), (1, 5), (5, 2), (2, 6), (6, 3),
     (3, 7), (5, 4)]
```

حال فهرستی از گره‌های تعبیه‌شده در گراف *(list on nodes)* را به صورت یک فهرست می‌توان تولید کرد:

```
In:  print(list(nx.dfs_tree(g,0)))
Out: [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```

همچنین با استفاده از دستور تابع درخت (*tree function*) می‌توانیم درخت جهتداری را که از گره منبع و براساس *DFS* گرفته شده است، تولید کنیم:

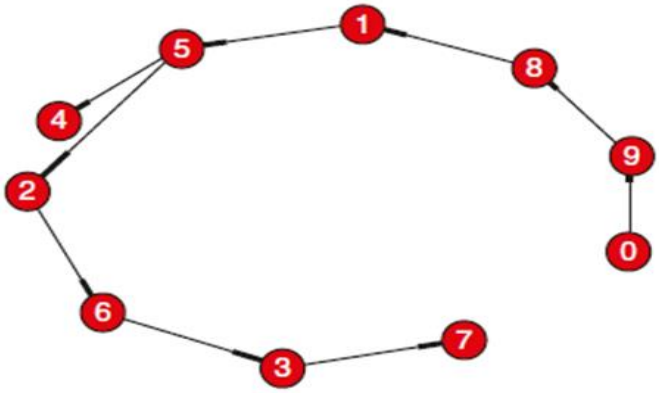
```
In:  tree = nx.dfs_tree(g, 0)
     print(list(tree.edges()))
Out: [(0, 9), (1, 5), (2, 6), (3, 7), (5, 2), (5, 4), (6, 3),
     (8, 1), (9, 8)]
```

[۱] Object Generator

[۲] Lazy Evaluation

در شکل ۱۳-۳ درخت *DFS* نشان داده شده است. قسمتهای ضخیم و پررنگ در این شکل نشان دهنده سر فلش (پیکان) می‌باشد.

```
In:  nx.draw_NetworkX(tree)

Out:
```



شکل ۱۳-۳ الگوریتم جستجوی اول-عمق در ساختار درخت مانند

تابع *dfs_successors* یک فرهنگ لغت (یا دیکشنری) را بازخواهد گرداند که در آن، گره‌ها بعنوان **کلید** و فهرستی از گره‌های لاحق یا پسین (گره‌های بعدی برای پیمایش گراف) بعنوان **مقدار** آن هستند:

```
In:  successors = nx.dfs_successors(g, 0)\
     print(successors)
Out:
     {0: [9], 1: [5], 2: [6], 3: [7], 5: [2, 4], 6: [3], 8:
     [1], 9: [8]}
```

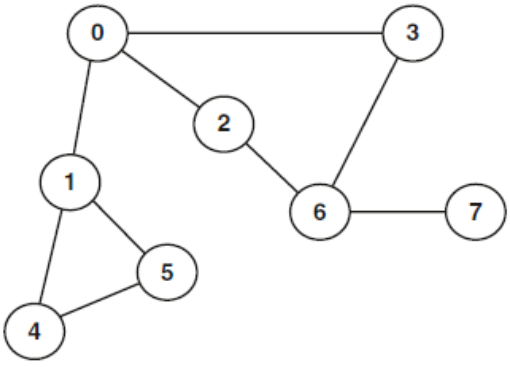
تابع *dfs_predecessors* یک فرهنگ لغت (یا دیکشنری) را ایجاد می‌کند که در آن گره‌ها بعنوان **کلید** و گره‌های سابق یا پیشین (گره‌های قبلی در پیمایش گراف) بعنوان **مقدار** است:

```
In:  predecessors = nx.dfs_predecessors(g, 0)
     print(predecessors)
Out: {1: 8, 2: 5, 3: 6, 4: 5, 5: 1, 6: 2, 7: 3, 8:  9, 9: 0}
```

۳-۵-۲ پیمایش اول-سطح

پیمایش اول-سطح یا الگوریتم جستجوی سطحی (*BFS*)، با دنبال کردن کوتاه‌ترین مسیر از یک راس منبع (یا راسهای منبع) به هر راس دیگر در هر بخش از شبکه، آن شبکه را می‌پیماید. روش کار *BFS* شامل جستجوی همه همسایگان گره‌ها در سطح فعلی و سپس جستجوی همه همسایگان بازدیده نشده در سطح یا پوسته بعدی است. به این ترتیب، الگوریتم می‌تواند کوتاهترین فاصله بین دو گره را محاسبه کند. همچنین اگر بیش از یک مسیر هندسی یا ژئودزیک بین دو گره وجود داشته باشد، الگوریتم جستجوی اول-سطح با انجام تغییرات جزئی می‌تواند آنرا پیدا کند.

روشی که الگوریتم جستجوی اول-سطح برای یافتن کوتاه‌ترین مسیر بکار می‌برد به شرح زیر است: ما فقط می‌دانیم که فاصله گره "s" از خودش صفر است و فاصله‌اش تا گره‌های دیگر نامعلوم است. سپس، تمامی همسایگان "s" را که براساس تعریف، دارای فاصله واحد با "s" هستند، پیدا کرده و به دنبال آن، همه همسایگان این رئوس را انتخاب می‌کنیم. صرف‌نظر از آنهایی که قبلاً بازدید شده‌اند، این گره‌ها باید از "s" دارای فاصله دو واحد بوده و گره‌های مجاور آن، البته بجز آن همسایگانی که قبلاً بازدید شده‌اند، دارای فاصله ۳ واحد بوده و همینطور تا آخر. (شکل ۱۴-۳).



شکل ۱٤-۳ الگوریتم جستجوی اول-سطح

در این قسمت با استفاده از گراف باراباشی- آلبرت در *NetworkX* به دنبال همان گامهایی هستیم که در بخش *DFS* به آنها اشاره شد.

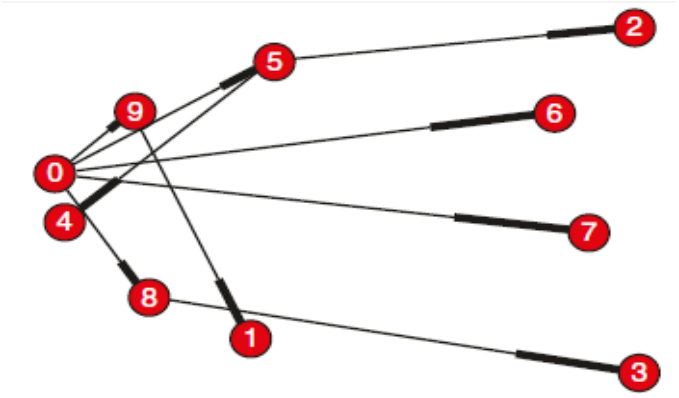
```
In:  edges = nx.bfs_edges(g, 0)
     print(list(edges))
Out: [(0, 8), (0, 9), (0, 5), (0, 6), (0, 7), (8, 3), (9, 1),
     (5, 2), (5, 4)]


In:  print(list(nx.bfs_tree(g,0)))
Out: [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```

بیایید درخت *BFS* را تعریف کنیم:

```
In:  tree = nx.bfs_tree(g, 0)
     print(list(tree.edges()))
Out: [(0, 8), (0, 9), (0, 5), (0, 6), (0, 7), (5, 2), (5, 4),
     (8, 3), (9, 1)]
```

همانطور که می‌بینیم، درخت جستجو در *BFS* با درخت جستجوی *DFS* متفاوت است. بیایید درخت *BFS* را رسم کنیم (شکل ۱۵-۳):

```
In:  nx.draw_NetworkX(tree)
Out:
```



شکل ۱۵-۳ الگوریتم جستجوی اول-سطح در ساختار درخت-مانند

سپس، گره‌های پسین و پیشین را حساب می‌کنیم:

```
In:  successors = nx.bfs_successors(g,0)
     print(successors)
Out: {0: [8, 9, 5, 6, 7], 8: [3], 5: [2, 4], 9: [1]}


In:  predecessors = nx.bfs_predecessors(g,0)
     print(predecessors)
Out: {1: 9, 2: 5, 3: 8, 4: 5, 5: 0, 6: 0, 7: 0, 8: 0, 9: 0}
```

همانطور که انتظار داشتیم، فهرست ایجاد شده در دیکشنری *BFS* برای گره‌های پسین و پیشین با فهرست ایجاد شده در دیکشنری *DFS* متفاوت است، که این امر گویای آن است که هر الگوریتم به دنبال راه خودش برای پیمایش گراف باراباشی- آلبرت (و همچنین هر گراف دیگری) است.

۳-۵-۳ الگوریتم دایکسترا

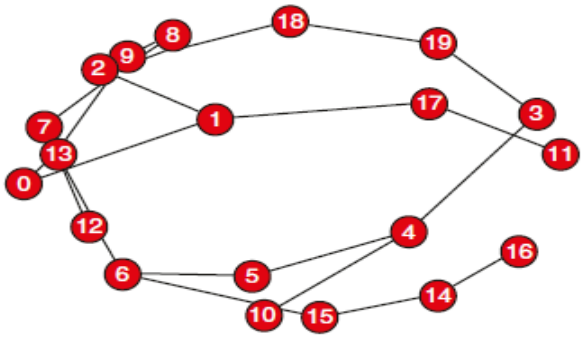
این الگوریتم توسط اِدسخِر دایکسترا[۱] در سال ۱۹۵۹ منتشر شد و یکی از مهم‌ترین الگوریتمها در شبکه‌های ارتباطی مدرن و مبنای بسیاری از الگوریتمهای مسیریابی است که در اینترنت استفاده می‌شوند. الگوریتم دایکسترا کوتاه‌ترین مسیر را از یک گره داده شده به هر گره دیگر در همان شبکه با در نظر گرفتن طول یالها پیدا می‌کند. این کار با حفظ سابقه (یا گزارش) کوتاه‌ترین مسیری که الگوریتم تاکنون پیدا کرده انجام می‌شود. این در حالی است که سابقه یا گزارش مربوطه هنگامی که یک مسیر کوتاه‌ترِ جدید، پیدا شود به روز می‌شود. در پایان اجرا، کوتاه‌ترین فاصله برای هر راس تعیین می‌شود.

مثال

NetworkX پیاده‌سازی خوبی برای الگوریتم دایکسترا دارد.

می‌خواهیم یک گراف ساده را رسم کرده و عملکرد برخی از توابع آنرا ببینیم (شکل ۱۶-۳).

```
In:  g = nx.watts_strogatz_graph(0.5 ,2 ,20)

     nx.draw_NetworkX(g)

Out:
```



شکل ۱۶-۳ گراف وات استروگاتس با ۲۰ گره

[۱] Edsger Dijkstra

نام این الگوریتم بر اساس نام ارائه‌دهنده هلندی آن، یعنی اِدسخِر دایکسترا انتخاب شده‌است. در منابع فارسی، آن را به شکلهای دایکسترا، دِیکسترا، دکسترا، دایجسترا، دیجسترا، دایجکسترا و دیجکسترا هم نوشته‌اند، ولی حرف جیمِ فامیل او، در تلفظ هلندی ادا نمی‌شود، لذا سه مورد اول، صحیح‌تر هستند.

```
In:  print(nx.info(g))

Out:

     Name: watts_strogatz_graph(20,2,0.5)

     Type: Graph

     Number of nodes: 20

     Number of edges: 20

     Average degree: 2.0000
```

دریافت جزئیات بیشتر از گراف وات استروگاتس[۱]:

```
In:   nx.watts_strogatz_graph?

Out:

Signature:

nx.watts_strogatz_graph(n, K, p, seed=None)

Docstring:

Return a Watts-Strogatz small-world graph.

Parameters

----------

n : int

    The number of nodes

K : int

    Each node is joinedwith its ``k`` nearest neighbors in a ring

    topology.
```

Watts-Strogatz [۱]

```
p : float
    The probability of rewiring each edge
seed : int, optional
    Seed for random number generator (default=None)
…
```

تابع *nx.dijkstra_path* نشاندهنده فاصله بین هر دو نقطه در یک گراف است. این تابع عملاً نقطه شروع، نقطه پایان، و وزن (اختیاری) را بعنوان ورودی دریافت کرده و فهرستی از گره‌ها در کوتاه‌ترین مسیر را بعنوان خروجی نمایش می‌دهد:

```
In:  nx.dijkstra_path(g, 0, 15)
Out: [0, 2, 1, 17, 18, 15]
```

همچنین می‌توانیم طول این مسیر را با استفاده از تابع *nx.dijkstra_path_length* بیابیم. این تابع بعنوان ورودی، یک گره منبع و یک گره مقصد و وزنی که اختیاری است را دریافت کرده و یک عدد صحیح را که بعنوان طول کوتاه‌ترین مسیر است، در خروجی نمایش می‌دهد:

```
In:  nx.dijkstra_path_length(g, 0, 15)
Out: 5
```

بیایید نگاهی به فهرست گره‌هایی بیاندازیم که در گراف وات استروگاتس هستند:

```
In:  print(g.nodes())
Out: [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,
     16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29,
     30, 31, 32, 33]
```

تابع *itertools.combinations()* از بسته *sitertool*، این امکان را ایجاد می‌کند تا مسیرهای موجود در گراف را ببینیم. آرگومان دوم، طول مسیر برگشت توسط تابع را مشخص می‌کند:

```
In:  import itertools
In:  print(list(itertools.combinations(g.nodes(), 3)))
Out: [(0, 1, 2), (0, 1, 3), (0, 1, 4), (0, 1, 5), (0, 1, 6), (0, 1, 7),
     (0, 1, 8), (0, 1, 9), (0, 1, 10), (0, 1, 11), (0, 1, 12), (0, 1, 13),
     (0, 1, 14),…]
```

می توانیم به طور متناوب نتایجی که از تابع فوق انتظار داریم را با مشخص کردن تنها تعداد کمی از گره‌ها (مثلاً ۵) محدودکنیم:

```
In:  print(list(itertools.combinations(g.nodes()[:5], 3)))
Out: [(0, 1, 2), (0, 1, 3), (0, 1, 4), (0, 2, 3), (0, 2, 4),
      (0, 3, 4), (1, 2, 3), (1, 2, 4), (1, 3, 4), (2, 3, 4)]
```

۶-۳ عملیات بر روی گرافها

ایجاد گرافهای جدید از گرافهای قدیمی با انجام یکی از عملیات یا روشهای زیر امکانپذیر است:

- **عملیات اولیه[۱]:** این روش، عملیات ویرایش نیز نامیده می‌شود. عملیات اولیه، گرافهای جدید را از طریق ایجاد تغییرات محلی در ساختار شبکه، مانند اضافه یا حذف گره‌ها (یا یالها)، ادغام یا تقسیم گره‌ها و غیره ایجاد می‌کند.

- **عملیات بازنویسی گراف[۲]:** یک نمونه از گراف جانشین جدید (و متناظر با گراف میزبان) را بجای وقوع یا رخداد برخی از گرافهای الگو در گراف میزبان قرار داده یا جایگذاری می‌کند.

- **عملیات یکتایی یا یگانی[۱]:** گراف کاملاً جدیدی از یک گراف قدیمی ایجاد می‌کند. مثالهایی از این دست، شامل گراف خطی، گراف دوگان[۲] و گراف مکمل می‌باشند.

---

[۱] Elementary Operations
[۲] Graph Rewrite Operations

- **عملیات دودویی[۳]:** یک گراف جدید از دو گراف والد ایجاد می‌کند. مثالهایی از عملیات دودویی عبارتند از: اجتماع گرافهای مجزا[۴]، ضرب دکارتی گرافها[۵]، ضرب تانسور گرافها[۶]، ضرب قوی گرافها[۷]، ضرب ترتیبی یا لکسیکوگرافیک گرافها [۸]

منبع:

Freeman LC (2004) *The development of social network analysis: a study in the sociology of science.* BooKSurge, LLC, North Charleston

[۱] Unary Operations
[۲] Dual Graph
Binary Operations [۳]
Disjoint Union of Graphs [٤]
Cartesian Product of Graphs [٥]
Tensor Product of Graphs [٦]
Strong Product of Graphs [۷]
[۸] Lexicographic Product of Graphs

فصل چهارم

شبکه‌های اجتماعی

این فصل، مفاهیم اصلی و برخی دیگر از موضوعات مرتبط با شبکه‌های اجتماعی مانند خواص شبکه‌های اجتماعی[۱]، جمع‌آوری داده‌ها در شبکه‌های اجتماعی[۲]، نمونه‌گیری داده‌ها[۳] و تجزیه و تحلیل شبکه‌های اجتماعی[۴] را معرفی می‌کند.

---

[۱] Properties of Social Networks
[۲] Data Collection in Social Networks
[۳] Data Sampling
[٤] Social Network Analysis

٤-١ شبکه‌های اجتماعی

"شبکه اجتماعی" یک سازه یا آمیزهٔ اجتماعی[۱] از کنشگران و روابط میان آنها می‌باشد. "کنشگران" می‌توانند افراد، مکانها، سازمانها، نقشها و غیره باشند، در حالیکه "روابط" می‌تواند از نوع خویشاوندی، دوستی، آشنایی، پژوهشی، نمایندگی، کارگزاری[۲] و غیره باشند. هر کنشگر در شبکه ممکن است به یک یا چند کنشگر دیگر مرتبط شود تا نهایتاً یک ساختار را ایجاد کند؛ چنین ارتباطاتی در عین حال بافتار اجتماعی[۳] آن فرد را نشان می‌دهد. شبکه اجتماعی را گاه "نقشه‌ای از پیوند بین گره‌ها" می‌دانند که توسط تحلیلگران بررسی می‌شود. معمولاً هر سیستمی که انسانها در آن مشارکت کرده و با هم ارتباط برقرار می‌کنند را می‌توان بعنوان یک "شبکه اجتماعی" در نظر گرفت.

گره‌های شبکه اغلب نشان‌دهنده افراد یا سازمانها هستند. با این وجود، گره‌ها می‌توانند همچنین صفحات وب، مقالات مجلات، ادارات، محله‌ها و یا حتی کشورها باشند (مارین و ولمن[٤] ۲۰۱۰). در سالهای اخیر علاقه بسیار زیادی به مطالعه شبکه‌های اجتماعی بوجود آمده که ناشی از دو نکته است. نکته اول، منافع و مزایایی است که انتظار می‌رود در پژوهش پیرامون شبکه‌های اجتماعی حاصل شود و نکته دوم، چالشهایی است که در مورد جمع‌آوری داده‌ها و تجزیه و تحلیل شبکه وجود دارد.

پیوندهایی که عاملان یا کنشگران را در یک شبکه بهم متصل می‌کنند می‌توانند جهتدار (برای نشاندادن منبع و مقصد یک رابطه[٥]) یا بدون‌جهت (برای نشان دادن یک رابطه متقابل[٦]) باشند. همچنین این پیوندها می‌توانند دوبخشی[۷] (برای نشان دادن وجود یا عدم وجود رابطه) یا دارای ارزش[۸] (اندازه‌گیری شده در یک مقیاس خاص) باشند (برگاتی و فاستر[۹] ۲۰۰۳).

---

[۱] Social Composition
[۲] Correspondence
[۳] Social Tissue
[٤] Marin and Wellman
[٥] Source and Destination of a Relationship
[٦] Reciprocated Relationship
[۷] Dichotomous
[۸] Valued
[۹] Borgatti and Foster

شبکه‌های اجتماعی را می‌توان به دو دسته تقسیم کرد: یک‌حالته (یا تک‌مُد)[۱] و چندحالته (یا چندمُد)[۲]. شبکه‌های یک‌حالته، شبکه‌هایی هستند که دارای "یک نوع گره" برای نشان دادن کنشگران (معمولاً افراد)، زیرگروهها یا اجتماعات هستند. "روابط گره‌ها" در این شبکه‌ها می‌تواند نمایشگر وضعیت روابط میان افراد[۳] (مانند دوستی)، جابجایی یا انتقال مواد (مانند قرض گرفتن یا خرید کردن)، انتقال غیرمواد[۴] (مانند ارتباطات) بوده و یا فی‌المثل تعاملات، نقشهای رسمی و خویشاوندی (مانند ازدواج) را نشان دهند.
نوع دوم شبکه‌های اجتماعی که شبکه‌های دوحالته[۵] هستند شامل "دو مجموعه مختلف از گره‌ها" می‌باشند. مثلاً مردم دارای اطلاعات و منابع هستند؛ سرمایه‌گذاران سهام را از شرکتها خریداری می‌کنند یا شرکتها، کارکنانی را استخدام می‌کنند و مثالهای بسیار دیگری از همین دست، که همه نمونه‌هایی از شبکه‌های اجتماعی دوحالته هستند. (بحثهای اصلی در این زمینه، در فصل ۷ ارائه شده است.)
اکثر شبکه‌ها، یک‌حالته هستند، اما برخی مسائل خاص نیاز به تجزیه و تحلیل دو نوع گره دارند، مانند سازمانها و اعضای آن سازمانها و یا رویدادها و افرادی که در این رویدادها حضور دارند. (مارین و ولمن[۶] ۲۰۱۰).

٤-۲ خواص یک شبکه اجتماعی

٤-۲-۱ شبکه‌های بدون مقیاس

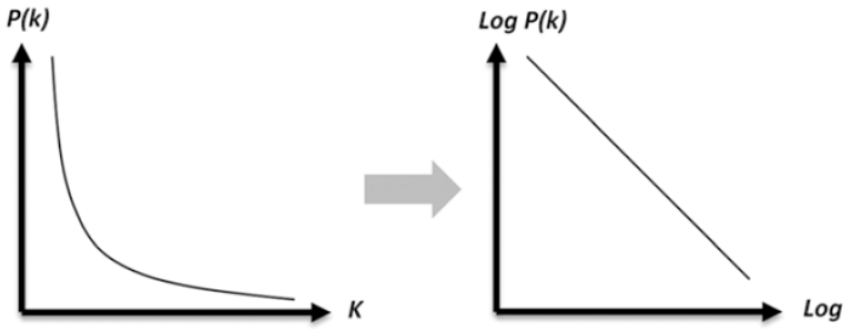
شبکه‌های بدون مقیاس شبکه‌هایی هستند که توزیع درجه آنها مطابق "توزیع قانون قدرت" باشد، یعنی:
$P(k) \sim k^{-\gamma}$
در این صورت: فقط چند گره، درجه زیادی دارند و درجه تعداد بسیاری از گره‌ها، کم است. توزیع درجه‌ای به صورت زیر محاسبه می‌شود:

---

[۱] One-Mode Networks
[۲] Multimode
[۳] Individual Evaluation
[٤] Transferring of Non-Materials
[٥] Two-Mode or Bimodal Networks
[٦] Marin and Wellman

$$P_k = \frac{1}{n} \# \left\{ i \middle| k_i = k \right\} \qquad ۱-۴$$

بسیاری از شبکه‌ها از جمله شبکه‌های استنادی، شبکه‌های بیولوژیکی، گراف *WWW*، گراف اینترنت و شبکه‌های اجتماعی دارای توزیع درجه‌ای قانون قدرت یا چوله راست[۱] هستند (شکل ۱-۴).



شکل ۱-٤ توزیع درجه‌ای قانون قدرت در شبکه‌های بدون مقیاس

"توزیع قانون قدرت" به این معنی است که تعداد اندکی از گره‌ها دارای اکثریت پیوندها بوده، در حالی که اکثریت گره‌ها، پیوندهای اندکی دارند. قانون مذکور بر اساس این ایده است که در شبکه یک هسته وجود دارد که متراکم است و حاشیه‌ای از گره‌ها[۲] که تعداد کمی ارتباطات دارند.

در این شبکه‌ها تعداد کمی از گره‌هایی که اتصالات خوبی دارند (یعنی قطبها)[۳] باعث می‌شوند قطر کل شبکه‌ها به طور قابل‌توجهی کاهش یابد. به نظر می‌رسد چنین توزیع درجه‌ای به طور پویا شکل گرفته و با ساز وکارهای کاملاً ساده و خود–سازمانده حفظ شده و باقی می‌ماند.

```
In:  import networkx as nx
     g = nx.scale_free_graph(40, alpha=0.41, beta=0.54, gamma=0.05,
     delta_in=0.2, delta_out=0, create_using=None, seed=None)
```

[۱] Right-Skewed

[۲] Periphery of Nodes

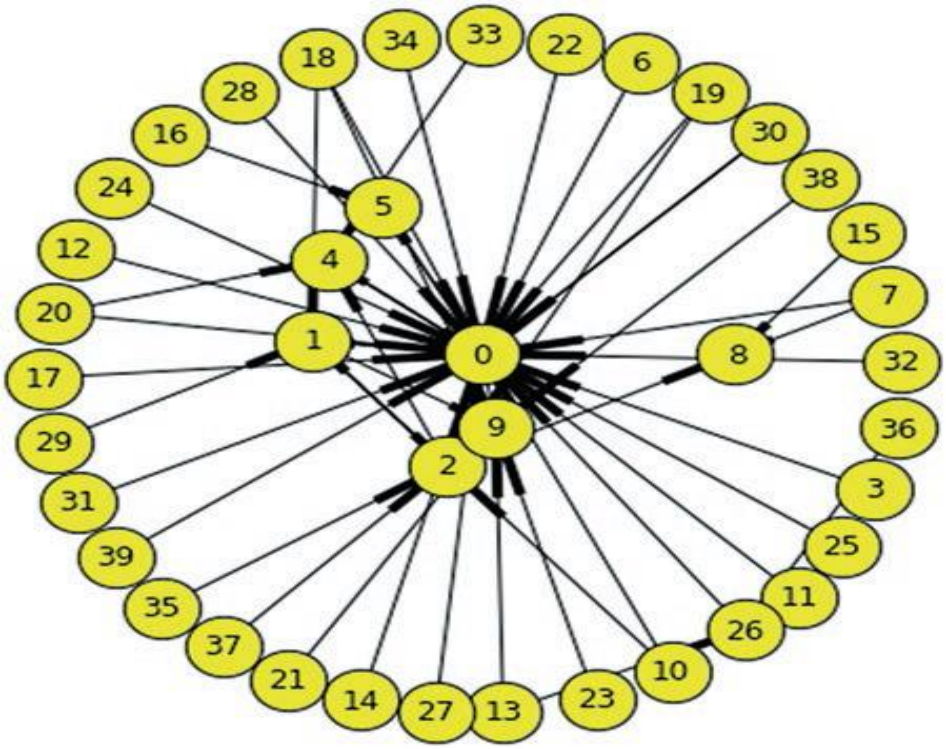
[۳] Hubs

تابع *nx.scale_free_graph* دارای پارامترهای زیر است:

- *n* (عدد صحیح): تعداد گره در گراف.
- آلفا (شناور): احتمال اضافه‌کردن یک گره جدید و اتصال به یک گره موجود که به طور تصادفی بر اساس "توزیع درجه ورودی" انتخاب شده است.
- بتا (شناور): احتمال اضافه‌کردن یک یال بین دو گره موجود. یک گره موجود به صورت تصادفی با توجه به "توزیع درجه ورودی" انتخاب می‌شود و دیگری به صورت تصادفی با توجه به "توزیع درجه خروجی" انتخاب می‌شود.
- گاما (شناور): احتمال اضافه‌کردن یک گره جدید و اتصال به یک گره موجود که به طور تصادفی با توجه به "توزیع درجه خروجی" انتخاب می‌شود.
- *delta_in* (شناور): بایاس یا سوگیری برای انتخاب گره‌ها از توزیع درجه ورودی.
- *delta_out* (شناور): بایاس یا سوگیری برای انتخاب گره‌ها از توزیع درجه خروجی.
- *create_using* (گراف، اختیاری (پیش فرض *MultiDiGraph*)): از این گراف برای شروع روند (به طور پیش فرض = ۳ حلقه) استفاده کنید.
- *seed* (عدد صحیح، اختیاری): بذر یا هسته برای مولد عدد تصادفی.

این تابع عملاً یک گراف جهتدار بی‌مقیاس را باز می‌گرداند (شکل ۲-۴).

```
In:  print(nx.info(g))
Out: Name:
      directed_scale_free_graph(40,alpha=0.41,beta=0.54,gamma=0.05,delta
      _in=0.2,delta_out=0)
      Type: MultiDiGraph
      Number of nodes: 40
      Number of edges: 78
      Average in degree: 1.9500
      Average out degree: 1.9500
 In: plt.figure(figsize=(8,8))
      layout = nx.spring_layout(g)
      nx.draw_networkx(g, layout=layout, with_labels=True,
      node_size=800, node_color='yellow')
```



شکل ۲-٤ گراف جهتدار بی مقیاس با ٤۰ گره

٤-۲-۲ شبکه‌های جهان کوچک

در شبکه‌های جهان کوچک[۱]، اغلبِ گره‌ها همگن[۲] بوده و با گامهای کمی در دسترس هستند. این بدان معناست که تعداد ارتباطات یا پیوندهای هر گره تقریباً یکسان است و فاصله بین هر دو گره، متناسب با لگاریتم رتبهٔ شبکه[۳] رشد می‌کند.

[۱] Small-World Networks

[۲] Homogeneous

[۳] Logarithm of the Order of the Network

به طور معمول، "شبکه‌های جهان کوچک"، دارای تعداد زیادی پیوند محلی و تعداد کمی راه میانبر دوربرد یا دوررس[۱] هستند. آنها دارای ضریب خوشه‌گی بالا، متوسط طول مسیر کوتاه و تعداد فراوانی گره‌های هاب هستند. این شبکه‌ها شامل اجتماعات چگال (متراکم) یا خوشه‌هایی هستند که به طور شُل و آزادانه توسط اتصالات مرزی (پلها یا رابطها[۲]) با هم ارتباط دارند. حتی اگر این شبکه‌ها یکنواخت[۳] باشند، به صورت نمایی[۴] واپاشی شده و یا از بین می‌روند.

کتابخانه *NetworkX* دارای یک پیاده سازی خوب برای "شبکه‌های جهان کوچک" است. یکی از معروفترین گرافها در این دسته‌بندی، گراف جهان کوچک واتس–استروگاتز *(WS)* است. این تابع، یک گراف جهان کوچک واتس–استروگاتز را باز می‌گرداند و سه پارامتر را می‌پذیرد:

n (int) که تعداد گره‌ها است.

k (in) که در آن هر گره به k گره همسایه نزدیک در یک توپولوژی حلقه متصل است.

***seed* (*optional int*)** برای مولد عدد تصادفی (*none* = پیش‌فرض) است.

نحوه ساخت گرافهای *WS* به شرح زیر است:

اول، یک حلقه از n گره ایجاد کنید. سپس هر گره در حلقه را با k گره همسایه نزدیک خود متصل نمایید (اگر k فرد است از k-1 همسایه استفاده کنید). سپس راههای میانبر را با جایگزین‌کردن برخی از یالها به شرح زیر ایجاد کنید: هر یال u-v در"حلقه nتایی با k گره همسایه نزدیک" را با احتمال p، با یال جدید u-w که بصورت تصادفی یکنواخت از گره موجودِ w انتخاب می‌شود، جایگزین کنید (شکل ۴–۳).

```
In: Import networkx as nx
            g = nx.watts_strogatz_graph(25, 5, 0.4)
            print(nx.info(g))
Out: Name: watts_strogatz_graph(25,5,0.4)
      Type: Graph
      Number of nodes: 25
      Number of edges: 50
```

---

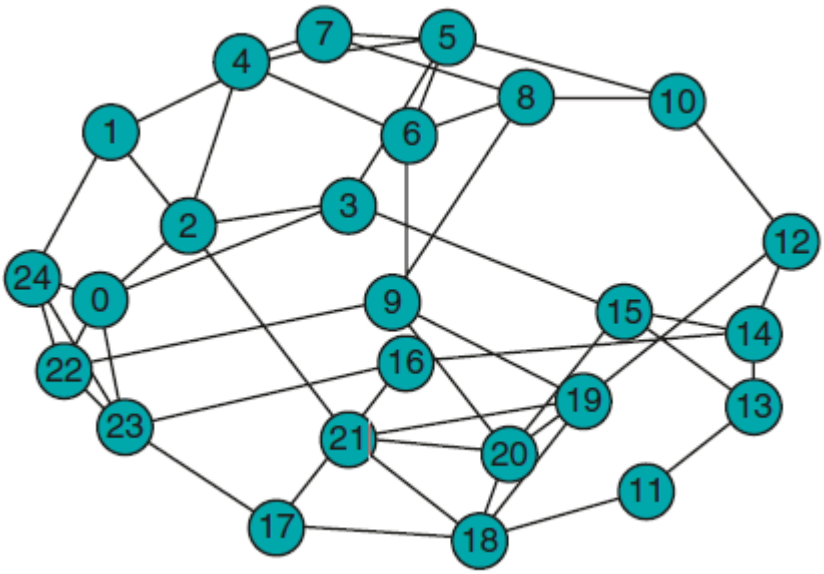
[۱] Long-Range Shortcuts

[۲] Connectors

[۳] Uniform

[۴] Exponentially

```
      Average degree: 4.0000
In: nx.draw_networkx(g, node_color='c', node_size=300)
Out:
```



شکل ۳-٤ گراف دنیای کوچک واتس-استروگاتز با ۲۵ گره

٤-۲-۳ ناوبری شبکه

ناوبری شبکه به این معنی است که سوای وجود پدیده جهان کوچک، افراد در یافتن مسیر در شبکه‌ها بخوبی عمل می‌کنند.

٤-۲-٤ عدد دانبار

در سال ۱۹۹۳ میلادی، رابین دانبار از گروه تحقیقاتی زیست‌شناسی تکاملی انسان در کالج دانشگاهی لندن متوجه رابطه‌ای بین شبکه‌های اجتماعی و تعامل انسانها شد.

از نظر دانبار[۱] و بر اساس محدودیتهای شناختی[۲]، تعداد افرادی که یک نفر می‌تواند با آنها رابطه اجتماعی پایدار[۳] برقرار کند به ۱۵۰ نفر محدود می‌شود. منظور از رابطه اجتماعی پایدار، روابطی است که در آن فرد بداند طرف مقابلش کیست و چه رابطه‌ای با سایر افراد دارد. بر این اساس، عدد پیشنهادی در این زمینه بین ۱۰۰ تا ۲۵۰ است، که معمولاً عدد ۱۵۰ استفاده می‌شود. عدد دانبار بیان کننده تعداد افرادی است که یک شخص می‌شناسد و با آنها تماس اجتماعی[۴] دارد. به بیان دیگر این عدد شامل افرادی نیست که شخصی آنها را می‌شناخته اما حالا با آنها رابطه اجتماعی ندارد و یا رابطه با آنان را متوقف کرده است. همچنین این عدد شامل آن افرادی نیست که شخص، آنها را بطور عام می‌شناسد اما با آنها هیچ رابطه اجتماعی ندارد؛ تعداد این افراد اخیر ممکن است خیلی زیاد باشد و احتمالاً بستگی به حافظه بلند مدت[۵] فرد دارد.

٤-۳ جمع‌آوری داده‌ها در شبکه‌های اجتماعی

یکی از خصوصیات شبکه‌های اجتماعی این است که شالوده داده‌های آنها ارزشهای فرهنگی[۶] و نمادها[۷] بوده و از طریق محرکها[۸]، معانی و علائم شکل می‌گیرند.

پیاده‌سازی تکنیکهای تحلیل شبکه‌های اجتماعی (*SNA*) نیازمند به یک "مجموعه داده کامل" (که همیشه در دسترس نیست) یا "مجموعه داده‌های نمونه" است که از مجموعه کامل استخراج می‌شود و بر اساس آن می‌توان موضوعاتی را در مورد همه جمعیت، استنتاج[۹] کرد.

---

۱ Robin Dunbar
۲ Cognitive Limit
۳ Stable Social Relationships
۴ Social Contact
۵ Long-Term Memory
۶ Cultural Values
۷ Symbols
۸ Motives
۹ Inference

محققان برای جمع‌آوری داده‌ها از روشهای سنتی مانند پرسشنامه‌ها، مصاحبه‌ها، مشاهدات و سوابق بایگانی استفاده می‌کنند در حالی که دادهٔ آنلاین را می‌توان با استفاده از رابط برنامه‌نویسی نرم افزارها[۱]، خزنده‌های وب[۲]، نظرسنجیهای آنلاین[۳] و برنامه‌های تخصصی[۴] جمع‌آوری کرد.

جمع‌آوری داده‌های شبکه‌های اجتماعی به ما اجازه می‌دهد تا به سؤالاتی همانند زیر پاسخ دهیم:

چه چیزی می‌توانیم در مورد پاسخ‌دهندگان بدانیم و یا یاد بگیریم؟ آیا داده‌های جمعیت‌شناختی[۵] مرتبط با آنها وجود دارد؟ چه نوع معیارهای کمی (مثلاً مرکزیت) را می‌توان از این داده‌ها استخراج کرد؟ علاوه بر این، چه نوع نتایج کمی و کیفی (مثلاً توانایی تبدیل شدن به یک شهروند خوب) را می‌شود اندازه‌گیری کرد؟

در خلال سالهای اخیر، روشهای بسیاری برای جمع‌آوری دادهٔ شبکه توسعه یافته‌اند که هدف آنها همواره پیداکردن ارتباط مابین کنشگران در شبکه است؛ معهذا ممکن است نوع داده‌ها، مدیریت داده‌ها و ابزار تجزیه و تحلیل داده‌ها در هر یک از این روشها متفاوت باشند. در برخی مواقع، محققان باید بیش از یک روش برای جمع‌آوری اطلاعات مشخصه کنشگر استفاده کنند. روشهای سنتی برای جمع‌آوری داده‌های اجتماعی عبارتند از (واسرمن و فاول[۶] ۱۹۹۴، ص. ۴۳):

۱) **پرسشنامه‌ها:** این روش، رایج‌ترین روش جمع‌آوری داده‌ها است. پرسشنامه‌ها به‌خصوص در شرایطی مفید هستند که پاسخ‌دهندگان بتوانند درباره روابط مورد مطالعه، گزارش دهند. پرسشنامه معمولاً شامل سوالاتی از قبیل اینکه آیا پاسخ‌دهنده با سایر افراد مرتبط هست یا خیر، می‌شود. پرسشنامه‌ها همچنین در مواردی که پاسخ‌دهندگان در یک شرکت کار می‌کنند، مفید هستند، زیرا می‌توان از یک گروه از افراد مرتبط، سوالات خاصی پرسید، مانند افرادی که دوست دارند با آنها غذا بخورند یا به آنها احترام بگذارند یا با آنها بیرون بروند یا با آنها مشورت کنند. طراحی پرسشنامه

---

[۱] APIs

[۲] Web Crawlers

[۳] Online Surveys

[۴] Specialized Applications

[۵] Demographic Data

[۶] Wasserman and Faust

می‌تواند به صور گوناگون باشد؛ مثلا: به صورت سیاهه‌ای از سوالات ترتیبی در مقابل فراخوان آزاد[۱] یا انتخاب آزاد در مقابل انتخاب معین[۲] و یا درجه‌بندی در مقابل رتبه‌بندی کامل[۳].

۲) **مصاحبه‌ها:** مصاحبه‌ها برای جمع‌آوری داده‌های شبکه یا به صورت ارتباطات چهره به چهره[۴] یا از طریق تلفن و یا از طریق هر وسیله دیگر مورد استفاده قرار می‌گیرند. از این روش، بیشتر زمانی استفاده می‌شود که استفاده از پرسشنامه نتایج مورد نظر را ارائه ندهد.

۳) **مشاهده:** داده‌های شبکه‌های اجتماعی را می‌توان با نظارت بر تعاملات مختلف بین افراد جمع‌آوری کرد. این روش برای مطالعه گروههای نسبتاً کوچک از افراد که مصاحبه‌های چهره به چهره آنان امکان پذیر نباشد، مورد استفاده قرار گرفته است. از این روش به طور گسترده در محیطهای مختلف برای نظارت بر رفتار و تعاملات افراد و همچنین جانوران تعاملدار مانند میمونها استفاده می‌شوند.

۴) **اسناد موجود در آرشیو[۵]:** در این روش، جمع‌آوری‌داده‌ها از طریق بررسی سنجشهای مربوط به گزارشهای تعاملات گذشته انجام می‌شود. این گزارشها ممکن است در رابطه با تعاملات سیاسی میان ملتها، گزارشهای استنادی محقق–محقق[۶]، مقالات مجلات، روزنامه‌ها، پرونده‌های دادگاه، جلسات و غیره باشند. اهمیت این روش در این است که داده‌ها می‌توانند اطلاعات مربوط به روابط طولانی مدت[۷] را فراهم کنند؛ اطلاعاتی که برای بازسازی روابط موجود در گذشته استفاده می‌شود.

۵) **روشهای دیگر:** سایر روشهای ممکن برای جمع‌آوری داده‌ها شامل طراحی ساختار اجتماعی، روشهای شناختی[۸]، مطالعات تجربی و همچنین انجام مطالعاتی است که هدف آنها جمع‌آوری

---

[۱] Roster Vs. Free Call
[۲] Free Vs. Fixed Choice
[۳] Ratings Vs. Complete Rankings
[۴] Face-to-Face Communications
[۵] Archival Records
[۶] Scholar-Scholar Citation Records
[۷] Longitudinal Relations
[۸] Cognitive

داده‌های شبکه‌های فردانه (یا خودمحور)[۱] است. از این روش اخیر، معمولاً برای تخمین اندازه و ترکیب (آمیزه) یک شبکه فردانهٔ تنها (خودمحور منفرد) استفاده می‌شود.

٤-٤ شش درجه جدایی

در سال ۱۹۶۹، یک روانشناس به نام استنلی میلگرام[۲]، آزمایشی انجام داد (که بعدها به نام خود او یعنی آزمایش میلگرام[۳] نامگذاری شد). او ۳۰۰ نامه را به بعضی آدرسهای تصادفی انتخاب شده در نبراسکا[۴] و کانزاس[۵] ارسال کرد. هر یک از این ۳۰۰ نامه، دستورالعملهایی در مورد چگونگی ارائه نامه به بوستون از طریق یک سری از آشنایان داشت. تنها ۶۴ نامه از ۳۰۰ نامه با یک طول زنجیره متوسط ۵,۵ به مقصد خود رسیدند.

٤-٥ شبکه‌های اجتماعی آنلاین

شبکه‌های اجتماعی آنلاین[۶] یا وب‌سایتهای رسانه‌های اجتماعی[۷] (که عملاً شکل نوین شبکه‌های‌اجتماعی سنتی هستند) دارای مکانیزمهای متعددی بوده که امکان درک، برقراری ارتباط، مشارکت و همکاری با یکدیگر را برای همه مردم فراهم می‌کند. پایگاههایی مانند فیسبوک، یوتیوب، لینکدین و توییتر این روزها به شدت برای سرگرمی و یا برای فرایندهای تجاری مورد استفاده قرار می‌گیرند. شما می‌توانید بازخورد خود را ارائه داده یا تجربیات خود را با فرد دیگری که انتخاب می‌کنید،

---

[۱] Ego-Network Data
[۲] Stanley Milgram
[۳] Milgram Experiment
[۴] Nebraska
[۵] Kansas
[۶] Online Social Networks
[۷] Social Media Websites

به اشتراک بگذارید. بسیاری از کسب و کارها، اهمیت و کیفیت خدمات ارائه شده توسط رسانه‌های‌اجتماعی را به رسمیت شناخته‌اند.

٤-٦ جمع‌آوری داده‌های اجتماعی آنلاین

جمع‌آوری داده‌های اجتماعی در گذشته، بسیار دشوار بود و به‌سختی به دست می‌آمدند. بسیاری از پژوهشگران این حوزه، مطالعات خود را فقط به مجموعه‌های کوچکی از داده‌های استاندارد، محدود می‌کردند و هرگز قادر نبودند مجموعه داده‌های مورد نظر خود را ایجاد کنند. ولیکن اینک، همه چیز با ظهور منابع آنلاین داده‌های‌اجتماعی مانند توییتر، فیسبوک و فلیکر که مقدار تولید داده توسط آنها بیش از حد انتظار است، تغییر کرده است.

شبکه‌های اجتماعی آنلاین به کاربران این امکان را می‌دهند که متن، صدا، ویدئو، ابرلینکها و عکسها را با دیگران به اشتراک بگذارند. این شبکه‌ها حجم زیادی از اطلاعات شخصی را ذخیره می‌کنند که نه تنها شامل اطلاعات مربوط به کاربران است، بلکه فعالیتهای اجتماعی آنلاین آنها را نیز نشان می‌دهند. بعنوان مثال، فیسبوک هر ماه بیش از ۳۰ میلیارد نسخه جدید محتوا را ذخیره می‌کند که توسط بیش از یک میلیارد کاربر فعال تولید می‌شود. فراتر از روشهای سنتی جمع‌آوری داده‌های اجتماعی، داده‌های آنلاین را می‌توان با روشهای زیر جمع‌آوری کرد:

۱. **رابط برنامه‌نویسی نرم‌افزارها یا *API*:** استفاده از *API*های ارائه شده توسط سرویسهای *OSN*. چنین استفاده‌ای شامل ارسال درخواستها[۱] به وب‌سایت مربوطه با کمک *API* برای جمع‌آوری اطلاعات است.

۲. **خزنده وب:** این روش شامل ساخت یک خزنده وب برای خزیدن در درون وب‌سایت است. خزنده از درخواست/پاسخ *HTTP* برای جمع‌آوری داده‌ها از وب‌سایت استفاده می‌کند.

۳. **نظرسنجیهای آنلاین[۱]:** نظرسنجیهای آنلاین، اطلاعات مربوط به فعالیتهای کاربر را با پرسش از او (در مورد چگونگی، زمان و کسانیکه با آنها محتوا به اشتراک می‌گذارد) جمع آوری می‌کنند.

[۱] Queries

۴. **گسیل برنامه‌های کاربردی[۲]:** در این روش یک برنامه سفارشی[۳] برای جمع‌آوری محتوا از وب‌سایت گسیل می‌شود. این رویکرد در مورد نظاره بر رفتار کاربر[۴] به هنگام مشاهده و استفاده از سایت و نیز جمع‌آوری داده‌هایی که کاربران به‌صورت آنلاین به اشتراک نمی‌گذارند، انعطاف‌پذیرتر است.

با این وجود، تمام روشهای جمع‌آوری داده‌های شبکه دچار برخی از نقاط ضعف یا نقص و کاستی، به ویژه در مورد دقت اطلاعات[۵]، اعتبار اطلاعات[۶]، قابلیت اطمینان اطلاعات[۷] و خطای اندازه‌گیری[۸] هستند (واسرمن و فاوست[۹] ۱۹۹۴، ص.۵۶).

٤-۷ نمونه‌گیری داده

در برخی موارد، کسب داده از تمام کنشگران موجود در مجموعه کنشگران یا عاملان مرتبط امکان‌پذیر نیست. بنابراین جمع‌آوری داده در مورد برخی از عاملان (برگرفته از میان همه کنشگران آن مجموعه)، کافی است و نتیجه‌گیری در مورد این جمعیت بعداً بر اساس همین "مجموعه نمونه" صورت می‌گیرد. داده‌های نمونه‌برداری شده در این مورد که می‌توان آنرا بعنوان "نماینده جمعیت بزرگتر" مشاهده کرد، "نمونه احتمالی" نامیده می‌شود.

چندین روش نمونه‌گیری توسعه یافته است. سه تکنیکی که غالباً از آنها استفاده می‌شود عبارتند از:

نمونه‌گیری از گره[۱]: در این روش، یک زیرمجموعه محدود از گره‌ها به‌همراه پیوندهایشان بعنوان داده‌های نمونه‌برداری شده، انتخاب می‌شوند.

---

۱ Online Surveys
۲ Deployed Applications
۳ Custom Application
۴ Monitoring User Behavior
۵ Information Accuracy
۶ Information Validity
۷ Information Reliability
۸ Measurement Error
۹ Wasserman and Faust

نمونه‌گیری از پیوند[۲]: در این روش، یک زیر مجموعه از پیوندها برای نمایش داده‌های نمونه‌برداری شده انتخاب می‌شود.

نمونه‌گیری گلوله‌برفی[۳]: این روش با مجموعه‌ای از گره‌های نمونه‌برداری شده به‌همراه همسایگانشان شروع می‌شود. این گره‌های نامزدشده، منطقهٔ مرتبهٔ اول[۴] شبکه را تشکیل می‌دهند. سپس، گره‌ها در این منطقه نمونه‌برداری می‌شوند و تمام ارتباطات آنها برای ایجاد منطقهٔ مرتبهٔ دوم استخراج می‌شوند. این فرایند چندین بار انجام می‌شود و در نهایت منجر به ایجاد شبکه‌ای از چندین منطقه می‌شود.

نمونه‌گیری گلوله‌برفی یکی از ساده‌ترین روشهاست (که بعنوان نمونه‌گیری زنجیره‌ای[۵] یا نمونه‌گیری پاسخگو-محور[۶] نیز شناخته می‌شود). در این روش، جمع‌آوری داده‌های نمونه با شناسایی جمعی از پاسخ‌دهندگان شروع می‌شود. این پاسخ‌دهندگان در مورد افرادی که با آنها رابطه دارند، گزارش می‌دهند. این کنشگران کاندید شده در واقع منطقهٔ مرتبهٔ اول شبکه را نشان می‌دهند. سپس، کنشگران در این منطقه، نمونه‌برداری می‌شوند و تمام ارتباطات و اتصالات آنها استخراج می‌شوند. از طریق این ارتباطات استخراج شده عملاً منطقهٔ مرتبهٔ دوم شبکه تشکیل می‌شود. این فرآیند چندین بار تکرار می‌شود تا یک شبکه با چندین منطقه ایجاد شود.

برای اجرای معمولی الگوریتم نیاز است عمق جستجو[۷] با یک مقدار از پیش تعیین‌شده محدود شود تا از مواردی نظیر انفجار داده[۸] که از ویژگیهای روش نمونه‌گیری گلوله‌برفی است و نیز محدودیت توان ذهن انسان در درک شبکه‌های اجتماعی (که همچنین بعنوان افق دید[۹] شناخته می‌شود) جلوگیری شود.

مثال نمونه‌گیری گلوله‌برفی زیر، از کتاب "تجزیه و تحلیل شبکه‌های اجتماعی برای شرکتهای نوپا یا استارتاپها[۱] " در اینجا نقل می‌شود. این مثال از داده موجود در وب‌سایتی به نام *LiveJournal* استفاده می‌کند که در روسیه و اروپای شرقی بسیار محبوب است.

---

۱ Node Sampling
۲ Link Sampling
۳ Snowball Sampling
۴ First-Order Zone
۵ Chain Sampling
۶ Respondent-Driven Sampling
۷ Depth of Search
۸ Explosion of Data
۹ Horizon of Observability

تابع زیر، فهرستی از دوستان با نام کاربریشان را از *API* انتخاب می‌کند و از طریق خطوط پاسخ، یالها را اضافه می‌کند. این تابع دو آرگومان می‌گیرد: یک شی گراف و یک دانه یا بذر رشته‌ای[۲]. این تابع، داده‌ها را برای یک گره معین، جمع‌آوری می‌کند.

```
In:  import sys
     import os
     import networkx as nx
     import urllib
In:
    def read_lj_friends(g, name):
        # fetch the friend-list from LiveJournal
response=urllib.urlopen('http://www.livejournal.com/misc/fdat
.bml? user='+name)
          for line in response.readlines():
             #Comments in the response start with a '#'
             if line.startswith('#'): continue
             #the format is"<name"(incoming)or">name" (outgoing)
             parts=line.split()
              #make sure that we do not have an empty line
              if len(parts)==0: continue
              #add the edge to the network
              if parts[0]=='<':
                  g.add_edge(parts[1],name)
              else:
                  g.add_edge(name,parts[1])
```

کد پایتون زیر، روش نمونه‌گیری گلوله‌برفی را بر روی داده‌های جمع‌آوری شده از سایت *LiveJournal* پیاده‌سازی می‌کند. این تابع دو آرگومان نیاز دارد: یک شی گراف و نام یک گره مرکزی. آرگومان اختیاری *max_depth* عمق جستجو را تعیین می‌کند (مقدار پیش‌فرض آن عدد یک است).

```
In:
```

---

[۱] Social Network Analysis for Startups

[۲] String Seed

```
def snowball_sampling(g, center, max_depth=1, current_depth=0,
taboo_list=[]):
   # if we have reached the depth limit of the search, bomb out.
   print center, current_depth, max_depth, taboo_list
   if current_depth==max_depth:
       print 'out of depth'
       return taboo_list
   if center in taboo_list:
       print 'taboo'
       return taboo_list #we've been here before
   else:
       taboo_list.append(center) # we shall never return
   read_lj_friends(g, center)
   for node in g.neighbors(center):
       taboo_list=snowball_sampling(g, node,
current_depth=current_depth+1, max_depth=max_depth,
taboo_list=taboo_l ist)
   return taboo_list
```

در اینجا یک شی گراف (*g = net. Graph()*) را تعریف خواهیم کرد، تابعی که با گره‌ها و یالها پر می‌شود. همچنین نام کاربری یکی از کاربران (که در این مورد فردی بنام *Valerois* هست) را به تابع *snowball_sampling ()* منتقل می‌کنیم. کاربر مربوطه کسی است که می‌خواهیم شبکه دوستی او را نمونه‌برداری کنیم. تابع *snowball* مذکور، نیز به نوبه خود، تابع *read_lj_friends ()* را برای جمع‌آوری اطلاعات مورد نیاز، فراخوانی می‌کند.

```
In: g=net.Graph()
    snowball_sampling(g,'valerois')
Out:
    valerois 0 1 []
    zhabaevent 1 1 ['valerois']
    out of depth
    farbys 1 1 ['valerois']
    out of depth
    zlove 1 1 ['valerois']
```

```
out of depth
cute_n_gorgeous 1 1 ['valerois']
out of depth
st_shtuchka 1 1 ['valerois']
out of depth
vorodis 1 1 ['valerois']
……
```

جهت ذخیره نتیجه نمونه‌گیری‌گلوله‌برفی در یک فرمت *pajek* :

```
In: net.write_pajek(g, "LiveJournaData.net")
```

برای باز کردن فایل در *NetworkX* ، می‌توان از تابع *nx.read_pajek ("LiveJournalData.net")* استفاده کرد.

٤-۸ تجزیه و تحلیل شبکه اجتماعی

تجزیه و تحلیل شبکه اجتماعی (*SNA*) شامل تجزیه و تحلیل ساختار روابط بین کنشگران (یعنی عاملان، بازیگران، افراد، گروههای انسانی یا سازمانی و بسیاری دیگر) می‌باشد. این رویکرد در پژوهشهای اجتماعی با چهار ویژگی متمایز است: داده‌های رابطه‌ای سیستماتیک[۱]، شهود ساختاری[۲]، مدلهای گرافیکی[۳] و مدلهای ریاضی. *SNA* متدولوژی خود را با مبنا قرار دادن زندگی و نگاه به روابط درون آن آغاز می‌کند: زندگی از روابط تشکیل شده و الگوهایی که از این روابط شکل می‌گیرند. تحلیل شبکه اجتماعی، علمی است که نشان می‌دهد مردم چگونه بر یکدیگر تاثیری می‌گذارند، که این نیز به نوبه خود منجر به شکل‌گیری ساختار بدنه دانش[٤] می‌شود. (فریمن[٥] ۲۰۰۴، ص.۶).

[۱] Systematic Relational Data
[۲] Structural Intuition
[۳] Graphical Models
[٤] Construction of Knowledge Body
[٥] Freeman

هدف نهایی *SNA* بررسی روابط میان افراد در یک شبکه اجتماعی است که علاوه بر شناخت ساختار کلی شبکه می‌تواند به معنای شناخت انواع روابط همچون نفوذ، مهربانی، ارتباط، مشاوره، دوستی، دشمنی، اعتماد، درگیری و یا بسیاری چیزهای دیگر باشد. از این رو، محققان علاقه‌مند به مطالعه چنین روابطی و تاثیر آنها بر رفتار فردی و عملکرد گروهی هستند.

SNA به بررسی رفتار انفرادی (مثلاً یک فرد یا یک سازمان) در سه سطح تحلیل، خرد[۱]، میانی[۲] و کلان[۳] و نیز تعاملات بین آنها، می‌پردازد. بنابراین، تحلیلگران شبکه ممکن است **(الف):** درگیر تجزیه و تحلیل یک شبکه به طور کامل (تحلیل جمعانه یا جامعه–محور) بوده یا **(ب):** به تجزیه و تحلیل بخشی از شبکه بپردازند (تحلیل زیرشبکه‌ها، تحلیل بخشانه یا گروه–محور) و یا **(ج):** ارتباطات یک گره خاص (تحلیل شبکه‌های فردانه یا فرد–محور) را بررسی کنند.

امروزه *SNA* مفاهیم، روشها، برنامه‌های کاربردی، نرم‌افزار تحلیلی و همچنین محققان حرفه‌ای خود را دارد و از یک استعاره پیشنهادی[۴] به یک ابزار تحلیلی و در نهایت یک پارادایم علمی تکامل یافته است. بسیاری از محققان در علوم کامپیوتری، اقتصاد، انسان‌شناسی (مردم‌شناسی)[۵]، زیست‌شناسی و غیره به *SNA* علاقه‌مند شده‌اند، به طوریکه *SNA* به یک ابزار تحلیلی قدرتمند برای بسیاری از مسائل دنیای واقعی تبدیل شده است.

راهی که تحلیلگران شبکه‌های اجتماعی داده‌ها را جمع‌آوری و تجزیه و تحلیل می‌کنند با آنچه که دانشمندان علوم اجتماعی انجام می‌دهند، متفاوت است.

با این حال، *SNA* دارای نواقصی در مسائل مربوط به اعتماد، حریم‌خصوصی[۶] و استراتژی در سازمانها و کمبود در ارائه نتایج موفقیت آمیز برای کاربردهای خاص همچون در ارزیابی آموزش و پرورش است.

---

[۱] Micro
[۲] Meso
[۳] Macro
[۴] Solely a Proposed Metaphor
[۵] Anthropology
[۶] Privacy

٤-۹ تحلیل شبکه اجتماعی در مقابل تحلیل پیوند

مفهوم دیگری که در بسیاری جهات به *SNA* شباهت دارد، تجزیه و تحلیل پیوند[۱] است. هر دو روش از ایده مشابهی در مورد کاربرد گره‌ها و یالها برای مدلسازی شبکه‌ها بهره می‌برند و هر دو تلاش می‌کنند تا عاملان کلیدی و مهم را در یک شبکه پیدا کنند. با این حال، تفاوتی هم با یکدیگر دارند. در روشِ "تحلیل پیوند" این امکان یا این جواز وجود دارد که انواع مختلف گره‌ها و یالها در یک شبکه خاص، بطور همزمان و با هم وجود داشته باشند؛ این در حالی است که حضور همزمان انواع گره و پیوند، ممکن است نتایج نامعتبری ایجاد کند. راهی که *SNA* برای حل این مشکل، یعنی شبکه‌ای با انواع مختلف گره‌ها و یالها بکار می‌گیرد، استفاده از متدولوژی شبکه‌های چندحالته است.

٤-۱۰ توسعه تاریخی

بروز و ظهور دانش تجزیه و تحلیل شبکه اجتماعی نتیجه همکاری و تعامل سه حوزه مختلف است:

(۱) تحلیلگران جامعه‌شناسی[۲] که در کار با گروههای‌کوچک داده، به پیشرفتهای فنی در نظریه گراف دست یافتند.

(۲) محققان هاروارد[۳]، که در دهه ۱۹۳۰ الگوهایی از روابط بین فردی و شکل‌گیری محافل، را کشف کردند.

(۳) مردم‌شناسان منچستر[۴]، که ارتباطات مردم و ساختارهای جامعه را در جوامع قبیله‌ای و روستایی مورد مطالعه قرار دادند.

ریشه‌های *SNA* به سال ۱۹۳۴ برمی‌گردد، یعنی زمانی که یعقوب مورنو[۵] کتاب مشهور خود را با عنوان «چه کسی باید زنده بماند؟»[۱] نوشت. این کتاب نقطه عطفی در توسعه این رشته بود و بدنبال آن

---

[۱] Link Analysis

[۲] Sociometric Analysts

[۳] Harvard Researchers

[۴] Manchester Anthropologists

[۵] Jacob Moreno

نوآوریهای مهم دیگری در نظریه شبکه اجتماعی توسط گروهی از محققان دانشگاه هاروارد در دهه ۱۹۷۰ ارائه شدند.

آلفرد رادکلیف براون[۲]، یکی از مردم‌شناسان اجتماعی معروف انگلیسی، با ارائه دیدگاههای خویش در یک قالب غیرفنی و انجام پژوهشهایی در حوزه کارکردگرایی[۳] و سازگاری ساختاری کمک زیادی به رشد این حوزه از علم کرد. به‌طوریکه می‌توان دید در دهه‌های ۱۹۳۰ تا ۱۹۷۰ میلادی تعداد زیادی از محققان از مفاهیم آلفرد در ساختار اجتماعی استفاده کرده‌اند.

علاوه بر مورنو و رادکلیف براون، دانشمندان دیگری نیز در گسترش این زمینه با استفاده از روشهای سیستماتیک کمک کردند، از جمله استفان بورگاتتی[۴]، کتلین کارلی[۵]، لینتون فریمن[۶]، استنلی واسمن[۷] و هریسون وایت[۸].

چهار ویژگی زیر، وضعیت استفاده فعلی از *SNA* را مشخص می‌کند:

- مطالعه *SNA*، به دلیل کنکاش در باره ترکیب ساختار انواع روابطی که کنشگران یا عاملان اجتماعی را بهم پیوند می‌دهد، بسیار انگیزه‌بخش است.
- *SNA* بر اساس پذیرش ورودیهای تجربی سیستماتیک، بنا شده و یا شکل می‌گیرد.
- *SNA* به شدت بر استفاده از نمایشهای گرافیکی، تکیه دارد.
- *SNA* همچنین بر بکارگیری مدلهای محاسباتی و ریاضی متکی است.

SNA تنها با این چهار ویژگی تعریف نمی‌شود، بلکه فراتر می‌رود و کاربردهای رو به رشد و گسترده‌ای را شامل می‌شود (فریمن ۲۰۰۴، ص. ۳–۵). با این‌وجود، اخیراً تنها همین چهار ویژگی در مدلهای تحقیقاتی علمی یکپارچه، ترکیب می‌شوند؛ به گونه‌ای که تحلیلگران اجتماعی یک یا تعدادی از این ویژگیها را در یک زمان به‌کارگرفته‌اند.

---

[۱] Who Shall Survive
[۲] Alfred Radcliffe-Brown
[۳] Coadaptation
[۴] Stephen Borgatti
[۵] Kathleen Carely
[۶] Linton Freeman
[۷] Stanley Wasserman
[۸] Harrison White

به محض اینکه اعضای "جامعه تحلیلگران شبکه اجتماعی" متوجه شدند که رویکرد ساختاری *SNA* می‌تواند تعمیم داده شود، برخی برنامه‌های مفید برای محیطهای تجربی مختلف در دسترس قرار گرفتند. یک مطالعه توسط اُت و روسو[۱] در سال ۲۰۰۲ برای اندازه‌گیری رشد *SNA* در طول یک مقطع تاریخی خاص یعنی از سال ۱۹۶۳ تا سال ۲۰۰۰ انجام شد. آنها "سه پایگاه داده" مرتبط با "سه حوزه علم" (یعنی جامعه‌شناسی، پزشکی و روانشناسی) را مطالعه کرده و تنها تحقیقات و مطالعاتی که عبارت "تجزیه و تحلیل شبکه اجتماعی" در "عنوان موضوع" خود داشتند، را مورد توجه قرار دادند. بدینوسیله تعداد ۱۶۰۱ مقاله در جامعه‌شناسی، ۳۰۸ مقاله در پزشکی و ۱۰۵ مقاله در روانشناسی بازیابی شد. این تحقیق نشان می‌دهد که رشد واقعی این حوزه از سال ۱۹۸۱ آغاز شده و هیچ نشانه‌ای از کاهش در رشد آن بوجود نیامده است. همچنین تحقیق فوق نشان می‌دهد که توسعه در این زمینه ابتدا از جامعه‌شناسی آغاز شده، سپس به پزشکی و در نهایت به روانشناسی رسیده است. موفقیتی که تجزیه و تحلیل شبکه‌های اجتماعی در دهه هشتاد قرن گذشته بدست آورد مرهون نهادینه‌سازی *SNA* در اواخر دهه هفتاد و نیز انتشار برخی از کتابهای درسی و نرم افزارهای کامپیوتری آن زمان است.

امروزه، تحلیلگران شبکه اجتماعی دارای سازمان بین‌المللی[۲] مختص خود به نام "شبکه بین‌المللی برای تجزیه و تحلیل شبکه اجتماعی[۳]" یا *INSNA* هستند که این سازمان، جلسات سالانه برگزار کرده و نشریات حرفه‌ای را منتشر می‌کند. همچنین برخی از مراکز تحقیقاتی شبکه[۴] در سراسر جهان افتتاح شده و دوره‌های عمومی و فنی زیادی نیز توسط دانشگاهها ارائه شده است.

٤-۱۱ اهمیت تحلیل شبکه اجتماعی

دانشمندان تجزیه و تحلیل شبکه اجتماعی، داده‌های شبکه را در سطوح مختلف تجزیه و تحلیل می‌کنند؛ این تحلیلها از سطح افراد، خانواده‌ها و گروهها شروع شده و تا سطح ملتها ادامه دارد. بنابراین، آنها کار مهمی انجام می‌دهند و تعیین می‌کنند که چگونه مسائل را در جامعه حل کنند، چگونه سازمانها می‌توانند

[۱] Otte and Rousseau

[۲] International Organization

[۳] International Network for Social Network Analysis

[۴] Centers for Network Research

کارایی بهتری داشته باشند و چگونه افراد می‌توانند سریعتر به اهداف خود برسند. اگرچه تجزیه و تحلیل شبکه‌های اجتماعی روشی نسبتاً جدید است، اما در کنار علم آمار، یک ابزار روش‌شناسی[۱] قدرتمند در رشته‌های مختلف است. پیاده‌سازی برنامه‌های *SNA* توصیفی[۲]، شامل تجزیه و تحلیل جنبه‌های ساختاری شبکه می‌شود، در حالیکه پیاده‌سازی برنامه‌های *SNA* توضیحی[۳]، شامل توضیح این مفهوم است که چگونه ویژگیهای گره، به وضعیت ساختارشان در درون شبکه، وابسته است.

استفاده از *SNA* با تشویق دانشمندان و محققان در زمینه‌های مختلف علمی آغاز شد، آنها اهمیت تحلیل شبکه اجتماعی را بعنوان ابزار تحلیلی موفق و بالقوه برای بسیاری از کاربردها متوجه شدند. بعنوان مثال در:

۱- سلامتی: در تحقیقات مربوط به همه‌گیری (اپیدمیولوژی) و جلوگیری از بیماریهای واگیردار

۲- جرایم اینترنتی: بررسی اجتماعات هکرکننده آنلاین

۳- کسب و کار: مطالعه تأثیر *SNA* و تحلیل احساسات در پیش‌بینی روند کسب و کار

۴- شبکه‌های اجتماعی حیوانات[۴]: بررسی روابط و ساختار اجتماعی تجمعات حیوانی و تعاملات مستقیم و غیرمستقیم بین گروهها

۵- ارتباطات: مطالعه خواص ساختاری مختلف گرافهای خدماتی پیام کوتاه.

علاوه بر این، *SNA* درسایر مکانها و در کنار سایر ابزارها مانند بازیابی اطلاعات، اجتماعات یا انجمنهای نفوذ اطلاعاتی[۵] و بررسی گروههای تروریستی قرار گرفته است.

٤-۱۲ ابزارهای مدلسازی تحلیل شبکه اجتماعی

ابزارهای مدلسازی تجزیه و تحلیل شبکه اجتماعی همان بسته‌های نرم‌افزاری هستند که می‌توانند برای شناسایی، ارائه، تجزیه و تحلیل و تجسم شبکه‌های اجتماعی استفاده شوند. مانند: *Pajek*، نرم‌افزاری

---

[۱] Methodological Tool

[۲] Descriptive SNA Applications

[۳] Explanative SNA Applications

[۴] Animal Social Networks

[۵] Information Fusion Communities

برای تجزیه و تحلیل و تجسم شبکه‌های بزرگ، *UCINET*، *StOCENT*، *Gephi*، *Network Workbench* و *NodeXL*.

چندین ابزار خاص–منظوره برای مدلسازی *SNA* توسعه داده شده‌اند که می‌توانند برای انجام وظایف تجزیه و تحلیل خاص مانند شناسایی زیرگروهها، شبکه‌های دانش، جمعیتهای پنهان، شبکه‌های خویشاوندی و شبکه‌های ساختاری مورد استفاده قرار گیرند. برخی از این ابزارهای خاص–منظوره عبارتند از: *NEGOPY*، *InFlow* و*SocioMetric LinkAlyzer*. برخی دیگر از ابزارهای مدلسازی *SNA* دارای امکان برنامه نویسی هستند، مانند *NetworkX* ، *JUNG*، *iGraph*، *Prefuse* و*SNAP*.

از آنجاییکه بسیاری از مفاهیم و تکنیکهایی که تجزیه و تحلیل شبکه‌های اجتماعی بر اساس آنها انجام می‌شود از نظریه گراف ریاضی اخذ شده اند، ضروری است گاهی اوقات به مفاهیم اصلی در نظریه گراف پرداخته شود.

هدف اصلی *SNA* یافتن الگوهای محلی و جهانی، شناسایی عاملان متنفذ و تاثیرگذار و نیز بررسی پویاییهای شبکه است. بنابراین، *SNA* بدنبال تحلیل پیوند بین افراد، سازمانها و حتی کشورهاست.

در طی سه فصل بعدی، به سه سطح تجزیه و تحلیل از جمله پیدا کردن بازیگران کلیدی شبکه اجتماعی در سطح فردی (فردانه)، شناسایی اجتماعات متمایز شبکه در سطح گروهی (بخشانه) و در نهایت اجرای تجزیه و تحلیل منسجم در سطح کل شبکه (جمعانه) پرداخته خواهد شد.

منابع

Borgatti SP, Foster PC (2003) The network paradigm in organizational research: a review and typology. *J Manag* 29(6):991–1013

Freeman LC (2004) *The development of social network analysis: a study in the sociology of science*. BookSurge, LLC, North Charleston

Marin A, Wellman B (2010) *Social network analysis: an introduction*. In: Carrington P, Scott J (eds) Handbook of social network analysis. SAGE Publications, London

Wasserman S, Faust K (1994) *Social network analysis: methods and applications*. Cambridge University Press, Cambridge

فصل پنجم

تجزیه‌وتحلیل درسطح گره

این فصل به بسط دانش خواننده در مورد چگونگی تجزیه و تحلیل شبکه در سطح گره یا کنشگر یعنی "تحلیل فردانه" می‌پردازد. منظور از این دانش آنست که دریابیم چگونه یک شبکه اجتماعی را از ابتدا[۱] ایجاد کنیم، چگونه آن شبکه را وارد[۲] (نرم‌افزار) کنیم، چگونه بازیگران یا کنشگران کلیدی آنرا با استفاده از معیارهای مرکزیت پیدا کنیم و بالاخره اینکه چگونه آن شبکه را مشاهده‌پذیر و متجسم یا مصور[۳] کنیم. همچنین در ادامه الگوریتمهای مهمی را معرفی خواهیم کرد که با استفاده از آنها می‌توان شناخت و بینش مناسبی از درون گرافها کسب کرد.

---

[۱] Scratch

[۲] Import

[۳] Visualize

۵-۱ تجزیه و تحلیل شبکه فردانه

وقتی به شبکه‌ای تنها از منظر یک کنشگر نگاه کنیم، به گونه‌ای که انگار صرفاً آن کنشگر در مرکز توجه ماست، شبکه‌ای حاصل می‌شود که به آن شبکه فرد-محور یا فردانه[۱] گویند. با تغییر آن کنشگر خاص (یا آن عامل کانونی[۲] یا فردانه[۳])، می‌توانیم به سمت مناطق مختلف در همه قسمتهای شبکه حرکت کنیم. در این نوع شبکه‌ها، نه همه کنشگران و عاملان بلکه تنها کنشگر کانونی، عملاً نقطه تمرکز تجزیه و تحلیل است.

تجزیه و تحلیل شبکهٔ فرد-محور یا فردانه، از یک سو، در ادبیات حمایت اجتماعی[۴] ریشه داشته و از سوی دیگر، از ادبیاتی نشأت گرفته که برای درک روابط میان افراد توسعه یافته است. در این نوع از تحلیل، تمام اطلاعات مربوط به شبکه از روی آن عامل یا کنشگر کانونی به‌دست می‌آید، فلذا موضوعات دیگر خیلی حائز اهمیت نیستند؛ موضوعاتی نظیر آنکه آیا سایر گره‌ها (گره‌های دگر-خود یا گره‌های دگرانه[۵]) وجود دارند یا خیر و یا آیا دگرانه‌ها (گره‌های همنشین در کنار آن کنشگر کانونی) به یکدیگر متصل هستند یا خیر. پارادایم فردانه[۶] در دهه ۲۰۰۰ با ظهور سایتهای شبکه اجتماعی محبوب، مانند فیسبوک و لینکدین توجه زیادی را به خود جلب کرده است.

تجزیه و تحلیل شبکهٔ فردانه شامل بررسی روابطی است که از یک فرد خاص (یا یک نمونه از افراد) به سمت مخاطبین آنها گسترش می‌یابد. انواع تجزیه و تحلیل شبکه فردانه با اهداف متفاوتی صورت می‌گیرد، برای مثال، جهت جمع‌آوری اطلاعات مربوط به روابط در یک مجموعه خاص، مانند یک کلاس درس در یک دانشکده یا یک دپارتمان در یک شرکت.

بعضاً مشاهده شده که توجه و علاقه تحلیلگران شبکه در خلال کارشان از مطالعه ویژگیهای کل شبکه (که به آن تجزیه و تحلیل جمع-محور[۷] یا جمعانه می‌گویند) به سمت بررسی زیرشبکهٔ یک گره خاص

---

[۱] Ego-Centered Network
[۲] Focal Actor
[۳] Ego
[۴] Social Support Literature
[۵] Alters
[۶] Ego-Centered Paradigm
[۷] Sociocentric Analysis

(که به آن تجزیه و تحلیل شبکه فرد–محور یا فردانه می‌گویند) تغییر جهت یافته است. لازم به ذکر است که این دو روش در مطالعه شبکه‌ها یعنی تحلیل شبکه فردانه و تحلیل شبکه جمعانه، دو رشته شناخته شده در شبکه‌های تحقیقاتی[۱] هستند. هر دوی آنها نسبتاً مستقل از یکدیگر توسعه یافته‌اند اما در عین حال آنها خیلی از یکدیگر فاصله نمی‌گیرند. درحالی‌که تجزیه و تحلیل داده‌های شبکه کامل[۲] یا جمعانه به طور عمده بر خواص ساختاری مانند چگالی، بینابینی و خوشه‌ها/محافل تمرکز می‌کند، تجزیه و تحلیل داده‌های فرد–محور یا فردانه بیشتر بر محتوای ارتباط یک فرد متمرکز است؛ مثلاً آن فرد با چه کسی در مورد مسائل شخصی‌اش صحبت می‌کند، ملاقاتهایش را در چه زمانی و در چه مکانی انجام می‌دهد و چندگانگی یا تکثر رابطه‌اش چگونه است (منظور از چندگانگی رابطه آنست که رابطه آن فرد، چند سطح یا لایه دارد). لازم به ذکر است که این دو رویکرد تجزیه و تحلیل شبکه، با هم رقابت نمی‌کنند، بلکه یکدیگر را تکمیل می‌کنند. به طور معمول ما به ترکیبی از هر دو تحلیل یعنی تحلیل شبکه فردانه و تحلیل شبکه جمعانه توجه می‌کنیم (بعنوان مثال، چگالش و تعداد مولفه‌ها).

اندازه شبکه فردانه را می‌توان براساس تعداد همنشینان یا دگرانه‌های آن شبکه بدست آورد (منظور از دگرانه، سایر کنشگران مرتبط با فرد مورد مطالعه است). این دگرانه‌ها یا هم‌نشینان، یک "مولفه یا سازند[۳]" را شکل می‌دهند که روابط منسجمی[۴] در میان خود و روابط ضعیف[۵] با دیگر کنشگران دارند. روابط بین فرد و دگرانه‌های موجود در سازند، همانند یک کلاف درهم[۶] است بدان معنا که معمولاً چگالی روابط فرد (گره فردانه) با هم‌نشینان موجود در سازند (گره‌های دگرانه)، بالاست. اعضای دگرانه (یا همنشینان) در درون یک سازند (یا مولفه)، معمولاً بدلیل داشتن یکی از ویژگیهای زیر مشخص می‌شوند: یا (الف) آن اعضا متقابلاً بهم مرتبط بوده یا (ب) ویژگی خاصی را به اشتراک گذاشته‌اند یا (ج) اینکه دارای وابستار مشترکی[۷] بوده و یا بالاخره (د) اینکه با همدیگر رخ می‌دهند. درک ساختار یک

---

[۱] Research Networks

[۲] Complete Network Analyses

[۳] Component

[۴] Cohesive Relations

[۵] Weak Relations

[۶] String

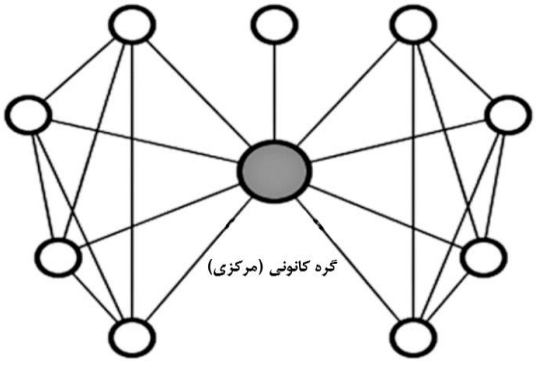
[۷] Joint Affiliation

نظر به استفاده از کلمه "وابستگی" بعنوان معادل چندین واژه انگلیسی مختلف، ما در اینجا بعنوان یک واژه اختصاصی فارسی برای کلمه Affiliation که به معنای وابستگی سازمانی است، واژه "وابستار" را پیشنهاد داده‌ایم.

سازند، به درک و تحلیل محیط اطرافِ فردانه کمک خواهد کرد. گاه ممکن است بیش از یک سازند به شبکه فردانه متصل باشد که البته هر سازند، اندازه و چگالی خاص خودش را دارد. اگرچه فردانه می‌تواند به مولفه‌ها یا سازندهای بسیاری مرتبط باشد (که خاصیتی از شبکه است به نام تنوع[۱])، اما روابط فردانه تنها بر روی تعداد محدودی از این سازندها متمرکز است و بر سایر سازندهایی که برتری یا برجستگی خاصی ندارد، تمرکزی ندارد (که خاصیت دیگری است از شبکه به نام سلسله‌مراتبی[۲] یا لایه‌بندی[۳]).

هر یک از این دو روند تجزیه و تحلیل شبکه‌ها، نرم‌افزار خاص خودشان را دارند. بعنوان مثال، *UCINET* یک برنامه مناسب برای تحلیل شبکه‌های کامل یا جمعانه است، در حالی که هر دو نرم‌افزار *SPSS* و *STATA* می‌توانند برای تحلیل شبکه‌های فردانه مورد استفاده قرار گیرند.

جمع‌آوری داده‌ها در یک شبکه معمولاً زمانی جالب است که جمعیت شبکه بسیار زیاد باشد؛ دراین حالت، نمونه‌ای از اشیا (یا فردانه‌ها) را انتخاب کرده و دادهٔ مربوط به آنها و همچنین هم‌نشینان (یا دگرانه‌های) آنها و پیوند میان آنها را جمع‌آوری می‌کنیم (شکل۱-۵).



شکل ۱-۵ شبکه فردانه شامل یک گره کانونی (گره بزرگتر) و نُه گره هم‌نشین (دگرانه) آن

تحلیل شبکه فردانه روش بسیار مفیدی برای درک ساختار شبکه‌های پیچیده و دیدن نحوه شکل‌گیری این ساختارها از طریق اتصالات یا پیوندهای محلی میان کنشگران منفرد است. با جمع‌آوری اطلاعات در مورد زیرشبکه‌ای که بر روی یک گره خاص متمرکز است (چنین زیرشبکه‌ای، شبکه فردانه نام دارد)

[۱] Diversity

[۲] Hierarchy

[۳] Stratification

می‌توانیم تصویری از شبکه‌های موضعی یا محلی آن گره (یا شخص) ایجاد کنیم. به‌طورمثال روابط دوستی در فیسبوک یا لینکدین نمونه‌هایی از زیرشبکه‌های متمرکز بر روی یک گره مشخص هستند که آن گره خاص در اینجا، صاحب حساب کاربری است.

مثال ۱

قصد داریم یک شی *NetworkX* از گراف زنان در شهر داویس جنوبی*Davis Southern Women* ایجاد کنیم. هیستوگرام را رسم کرده و سپس با استفاده از بسته *matplotlib* گراف را رسم خواهیم‌کرد.

```
In: g = nx.davis_southern_women_graph()
```

بیایید فهرست گره‌ها و یالهای گراف را بررسی کنیم:

```
In:  g.nodes()

Out:

     ['Flora Price', 'Nora Fayette', 'E10', 'Verne Sanderson', 'E12',

     'E14', Frances Anderson', 'Dorothy Murchison', 'Evelyn Jefferson',

     'Ruth DeSand', Helen Lloyd', 'Olivia Carleton', 'Eleanor Nye',

     'E11', 'E9', 'E8', 'E5', E4', 'E7', 'E6', 'E1', 'Myra Liddel',

     'E3', 'E2', 'Theresa Anderson', Pearl glethorpe', 'Katherina

     Rogers', 'Brenda Rogers', 'E13', 'Charlotte McDowd', Sylvia

     Avondale', 'Laura Mandeville']
```

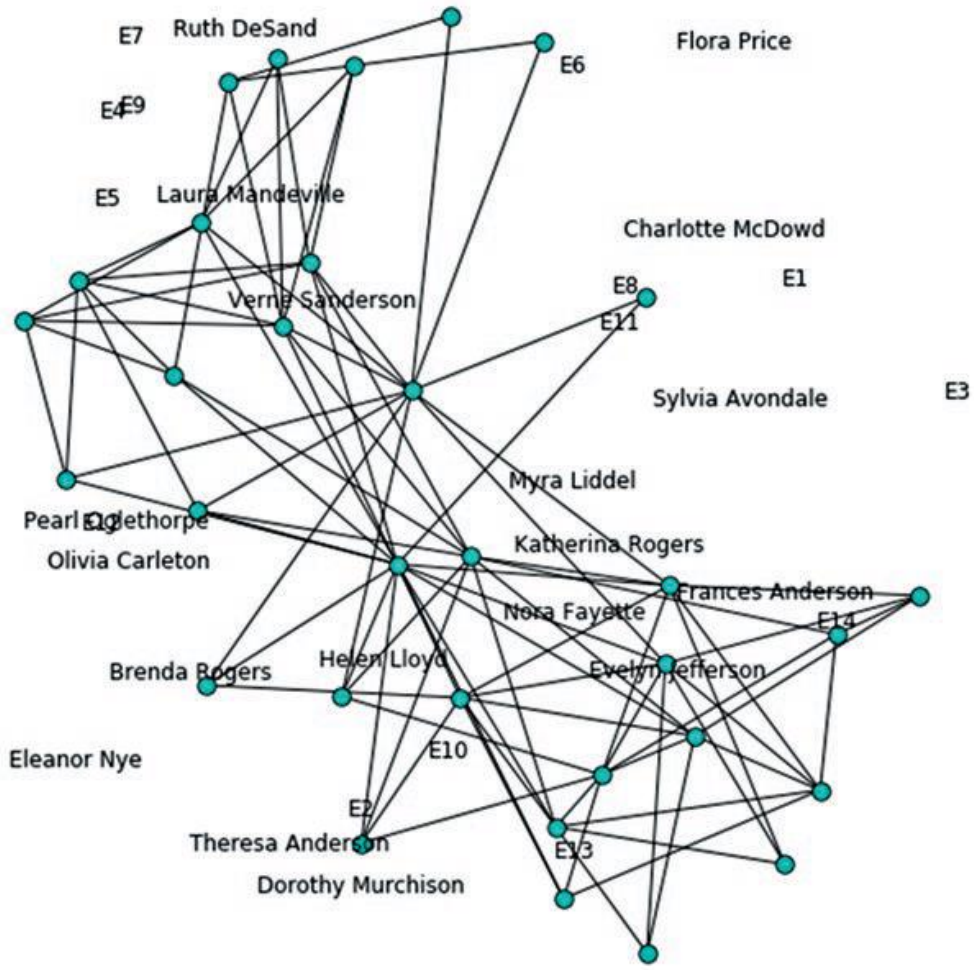
یک معیار مهم برای یک گره در یک گراف، "درجه" آنست که این معیار برای گرافهای بدون جهت، تعداد اتصالات گره است. از سوی دیگر، برای گرافهای جهتدار، یک گره ممکن است دارای دو نوع درجه باشد. "درجه ورودی" که نشاندهنده پیوندهای واردشونده به گره است و "درجه خروجی" که نشاندهنده پیوندهای خارج‌شونده از گره است. بیایید درجه هر گره از این گراف را محاسبه کنیم:

برای مشاهده و یا مصورسازی شکل به روش زیر عمل می‌کنیم:

```
In:  plt.figure(figsize=(10, 10))

     pos = nx.random_layout(g)
```

```
nx.draw(g, node_size = 90, node_color='cyan', with_labels=False)
nx.draw_networkx_labels(g, pos)
plt.show()
```

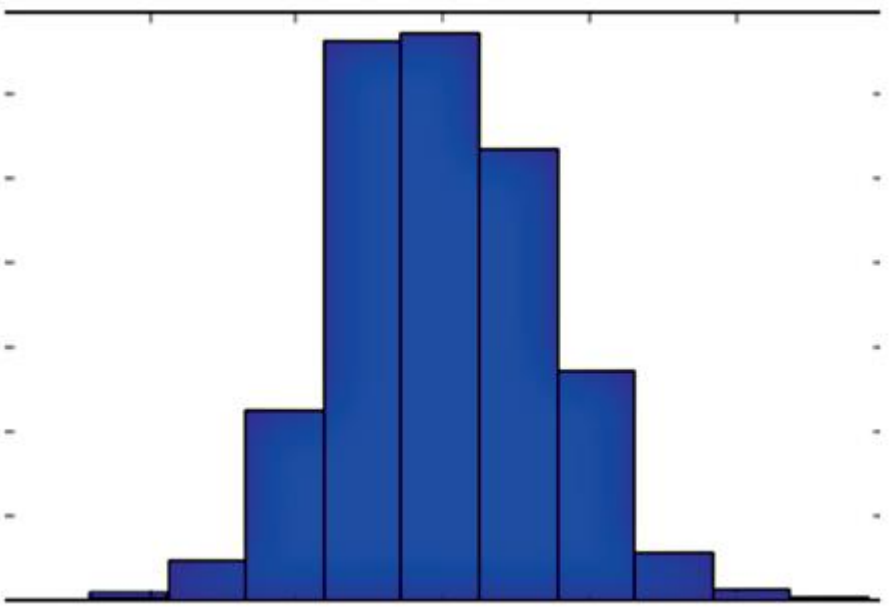
و بدینسان گراف مصورشده در شکل ۲–۵ را دریافت می‌کنیم:



شکل ۲–۵ گراف زنان در شهر داویس جنوبی

```
In:  g.degree()

Out:

    {'Brenda Rogers': 7, 'Charlotte McDowd': 4, 'Dorothy Murchison': 2,

    'E1': , 'E10': 5, 'E11': 4, 'E12': 6, 'E13': 3, 'E14': 3, 'E2': 3,

    'E3': 6, 'E4': , 'E5': 8, 'E6': 8, 'E7': 10, 'E8': 14, 'E9': 12,

    'Eleanor Nye': 4, 'Evelyn efferson': 8, 'Flora Price': 2, 'Frances

    Anderson': 4, 'Helen Lloyd': 5, Katherina Rogers': 6, 'Laura

    Mandeville': 7, 'Myra Liddel': 4, 'Nora ayette': 8, 'Olivia

    Carleton': 2, 'Pearl Oglethorpe': 3, 'Ruth DeSand': 4, Sylvia

    Avondale': 7, 'Theresa Anderson': 8, 'Verne Sanderson': 4}
```

معمولاً برای تقریب "توزیع درجه گره‌ها" در گرافها (و بویژه در گرافهای بزرگ) از نموداری همچون هیستوگرام استفاده می‌شود. فرض کنید که ما یک گراف تصادفی با ۵۰۰۰ گره داریم و احتمال ایجاد پیوند ۰٫۰۱ است. هیستوگرام درجه گره به شرح شکل (۵–۳) است.



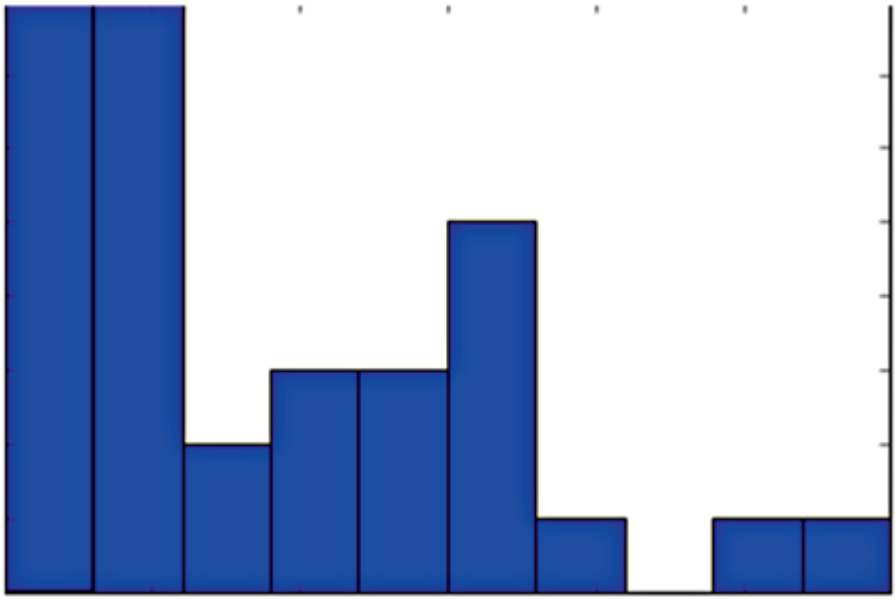
شکل ۳-۵ توزیع درجه‌ای یک گراف تصادفی با ۵۰۰۰ گره

```
In:  plt.hist(nx.fast_gnp_random_graph(5000, 0.01).degree().values())
Out: (array([ 13., 94., 451., 1330., 1345., 1074., 547.,
     114., 26., 6.]), array([ 26. , 31.3, 36.6, 41.9,
     47.2, 52.5, 57.8, 63.1, 68.4, 73.7, 79. ]),
     <a list of 10 Patch objects>)
```

با بازگشت به مثالمان، بیایید هیستوگرام گراف شهر زنان داویس جنوبی را بکشیم (شکل ۴-۵):

```
In:  plt.figure(1)
     plt.hist(nx.degree(g).values())
Out:
     (array([ 8., 8., 2., 3., 3., 5., 1., 0., 1., 1.]),
     array([ 2. , 3.2, 4.4, 5.6, 6.8, 8. , 9.2, 10.4,
```

```
    11.6, 12.8, 14. ]),
    <a list of 10 Patch objects>)
```



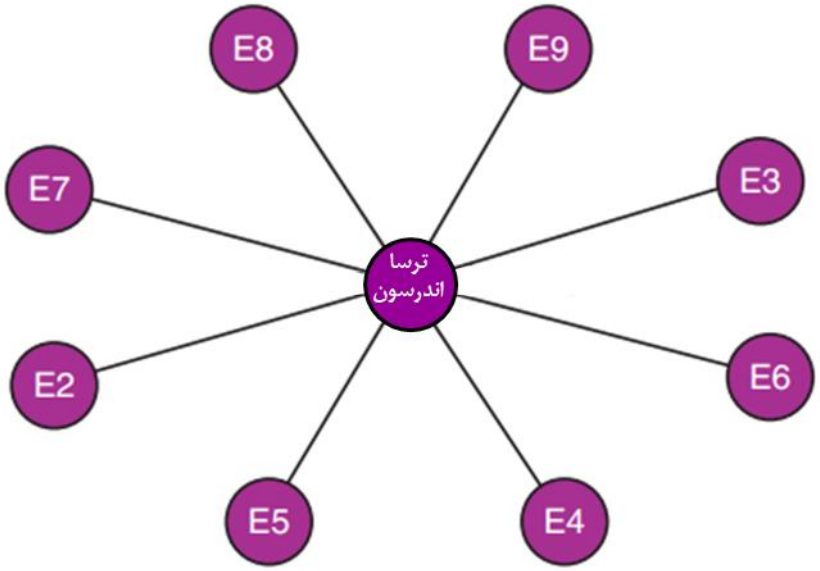
شکل ٤-٥ توزیع درجه‌ای از شبکه شهر زنان داویس جنوبی

اکنون، بیایید یک "شبکه فردانه" از نمودار گرافیکی زنان شهر داویس جنوبی استخراج کنیم. کتابخانه *NetworkX* یک تابع آماده[۱] برای استخراج این شبکه فراهم می‌کند. این تابع، یک شی گراف (*graph object*) را به ما نشان می‌دهد که می‌توانیم تمام عملیات شبکه را بر روی آن اجرا کنیم (شکل ۵-۵).

```
In:  nx.ego_graph(g, 'Theresa Anderson')
Out: <networkx.classes.graph.Graph at 0xd735b00<
In:  ego_net = nx.ego_graph(g, 'Theresa Anderson')
In:  print(nx.info(ego_net))
```

[۱] Ready-Made Function

```
Out: Name:
     Type: Graph
     Number of nodes: 9
     Number of edges: 8
     Average degree: 1.7778
In:  pos=nx.shell_layout(ego_net)
     nx.draw_networkx(ego_net, pos, node_size=800,
     node_color='magenta')
Out:
```

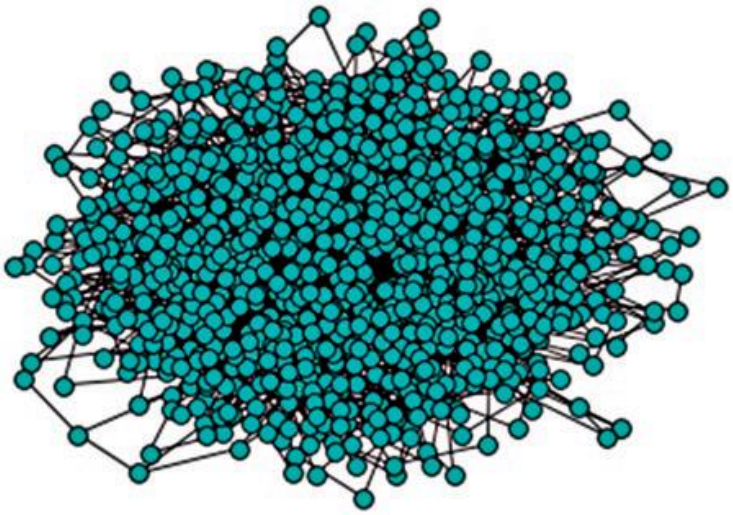


شکل ۵-۵ شبکه فرد-محور (فردانه) از یک گره در گراف زنان شهر داویس جنوبی

مثال ۲

در این مثال قصد داریم، ابتدا یک گراف بر اساس مدل باراباشی–آلبرت با هزار گره (یعنی $n = 1000$) و و حداقل درجه برای هر گره معادل ۲ (یعنی $m=2$) بسازیم. پس از آن، می‌خواهیم یک زیرگراف از این گراف اصلی به گونه‌ای استخراج کنیم که نمایشگر شبکه "گره‌ای با بزرگترین درجه" در کل شبکه باشد (شکلهای ۶–۵ و ۷–۵).

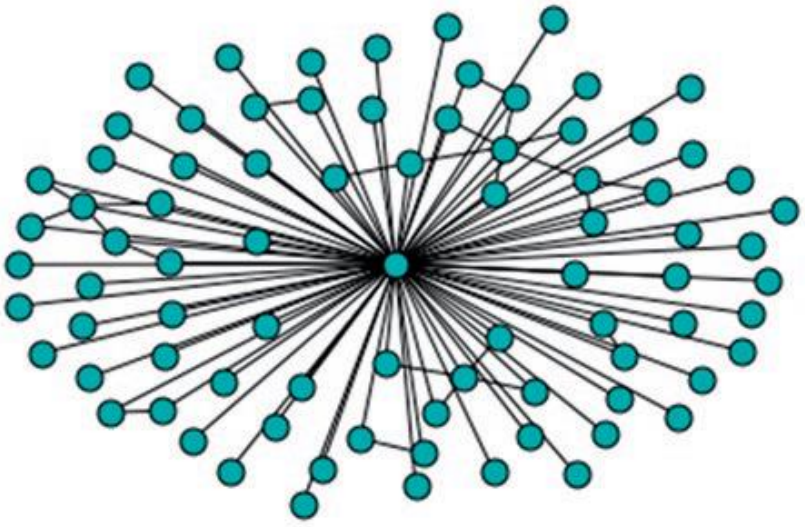
```
# Create a Barabasi-Albert graph model

In:  G = nx.barabasi_albert_graph(1000, 2)


# Let's draw the graph

In:  pos= nx.spring_layout(G)

In:  nx.draw(G, pos, node_color='c', node_size=60 ,with_labels=False)
```



شکل ٦–۵ نمودار باراباشی آلبرت با ۱۰۰۰ گره

```
# find node with largest degree
```

```
In:  node_degree = G.degree()


# calculate number of nodes

In:  G.order()

Ou:  1000


# calculate number of edges

In:  G.size()

Out: 1996


# find node with the largest degree

In:  node_degree= G.degree()

In:  (largest_deg, degree)=

sorted(node_degree.items(),key=itemgetter(1))[-1]


# Create ego graph of node with the largest degree

In:  node_ego= nx.ego_graph(G,largest_deg)


# Draw ego graph

In:  pos= nx.spring_layout(node_ego)

In:  nx.draw(node_ego, pos, node_color='c', node_size=120

,with_labels=False)
```



شکل ۷-۵ شبکه فرد-محور (فردانه) یک گره از گراف باراباشی-آلبرت

مثال ۳

در این مثال، مجموعه داده‌های گراف اجتماعی دنیای مارول[۱] را بررسی می‌کنیم.

قبل از حل مثال، یکی دو نکته تاریخی از ویکی پدیا برای فهم بهتر مثال در زیر آورده شده است: شرکت جهان‌گستر مارول[۲] که معمولاً به نام مارول کُمیکس[۳] و قبل از آن تحت نام شرکت انتشاراتی «تایملی» شناخته می‌شد، یک شرکت آمریکایی است که در سال ۱۹۳۹ تأسیس شد و کتاب کُمیک عرضه می‌کند. همچنین در رابطه با این کتابها، فعالیتهایی در سینما و تلویزیون دارد. در سال ۲۰۰۹ شرکت والت دیزنی با خرید ۴,۲۴ میلیارد دلاری، مارول اینترتینمنت (شرکت مادر مارول کُمیک) را تصاحب کرد.

فیلمهای دنیای سینمایی مارول، مجموعه فیلمهای ابرقهرمانی است که بر پایه شخصیتهای "مارول کُمیکس"ساخته و منتشر می‌شوند. تولید این فیلمها از سال ۲۰۰۷ آغاز شده و تاکنون ادامه داشته است. این مجموعه فیلمها، پرفروش‌ترین مجموعه فیلم تاریخ سینماست. "کوین فایگی" تهیه‌کننده اصلی تمامی فیلمهای دنیای سینمای مارول بوده است. "آوی آراد" تهیه‌کننده دو فیلم منتشر شده در سال ۲۰۰۸ و "گیل آن هرد"

---

[۱] Marvel Universe Social Graph

[۲] Marvel Worldwide, Inc.

[۳] Marvel Comics

تهیه‌کننده هالک شگفت‌انگیز بود. رابرت داونی جونیور، کریس ایوانز، کریس همسورث، ساموئل ال. جکسون و اسکارلت جوهانسون، بازیگرانی هستند که در بیشتر فیلمهای دنیای سینمایی مارول حضور داشته‌اند.

اولین فیلم دنیای سینمایی مارول "مرد آهنی" بود که "پارامونت پیکچرز" آن را توزیع کرد. پارامونت در ادامه مرد آهنی ۲ (۲۰۱۰)، ثور (۲۰۱۱) و کاپیتان آمریکا: نخستین انتقام‌جو (۲۰۱۱) را منتشر کرد و یونیورسال پیکچرز هم عهده‌دار توزیع هالک شگفت‌انگیز (۲۰۰۸) بود. "والت دیزنی استودیوز موشن پیکچرز" کار توزیع فیلمهای مارول را با انتقام‌جویان شروع کرد، که آخرین فیلم فاز اول بود. فاز دو شامل فیلمهای مرد آهنی۳ (۲۰۱۳)، ثور: دنیای تاریک (۲۰۱۳)، کاپیتان آمریکا: سرباز زمستان (۲۰۱۴)، نگهبانان کهکشان (۲۰۱۴)، انتقام‌جویان: عصر اولتران (۲۰۱۵) و مرد مورچه‌ای (۲۰۱۵) است.

هم‌چنین مرد عنکبوتی: دور از خانه نیز در سال ۲۰۱۹ منتشر شده و سه فیلمی که هنوز نامشان مشخص نشده، قرار است در ۲۰۲۰ منتشر شوند. ساخت نگهبانان کهکشان بخش ۳ نیز تأیید شده و سال ۲۰۲۰ برای انتشار آن مشخص گردیده است. اندکی از اطلاعات مربوط به این فیلمها و قهرمانان آنها در جدول (۵-۱) آمده است.

جدول ۱-۵ شخصیتهایی که حداقل در دو فیلم از دنیای سینمایی مارول حضور داشته‌اند.

شخصیت	فیلمهای مرد مورچه‌ای	فیلمهای انتقام‌جویان	فیلمهای کاپیتان آمریکا	کاپیتان مارول	فیلمهای نگهبانان کهکشان	هالک شگفت‌انگیز	فیلمهای مرد آهنی	فیلمهای ثور
بروس بنر هالک		مارک روفالو				ادوارد نورتون لو فریگنو	مارک روفالو	مارک روفالو
جیمز باکی سرباز زمستان	سباستین استن	سباستین استن	سباستین استن					
کلینت بارتون شاهین‌چشم		جرمی رنر	جرمی رنر					جرمی رنر
پگی کارتر	هایلی اتول	هایلی اتول	هایلی اتول					
درکس د دسترویر		دیوید باتیستا			دیو باتیستا			
نیک فیوری		ساموئل ال. جکسون		ساموئل ال. جکسون	ساموئل ال. جکسون		ساموئل ال. جکسون	ساموئل ال. جکسون

مجموعه داده‌های "گراف اجتماعی دنیای مارول" توسط راسلولو[۱]، ریکاردو آلبریچ[۲] و جو میرو[۳] جمع‌آوری و معرفی شده است. مجموعه داده‌های فوق با گردآوری نام کاراکترها یا شخصیتها و مرتبط کردن آنها با کتاب کُمیک[۴] یا طنزنامه‌ای (که در آن حضور داشته‌اند) ایجاد شده است.
فرمت داده‌ها بصورت *csv* است، حدود ۲۱ مگابایت بوده و دارای ۶۴۲۶ گره و ۱۶۷،۲۱۹ یال است. ابتدا کتابخانه‌های مهم را وارد کنید:

```
In: import networkx as nx
    import unicodecsv as csv
    import numpy as np
    import scipy.stats
    import matplotlib.pyplot as plt
    from operator import itemgetter
```

در گام بعدی، بیایید تابعی تعریف کنیم که "شبکه قهرمانی"[۵] را با اضافه‌کردن یالهای بین قهرمانها ایجاد کند (از این رو، گره‌ها به طور خودکار اضافه می‌شوند). این کار بر اساس مجموعه داده‌ای که به تابع (قهرمان–به–قهرمان[۶]) منتقل می‌کنیم، انجام می‌شود:

```
In: def graph_func(path):
            graph = nx.Graph(name="Heroic Network")
            with open(path, 'rU') as data:
                    reader = csv.reader(data)
                    for row in reader:
```

[۱] Cesc Rosselló
[۲] Ricardo Alberich
[۳] Joe Miro
[۴] Comic book
[۵] Heroic Network
[۶] Hero-to-Hero

```
                graph.add_edge(*row)

        return graph
```

سپس تابع را فراخوانی کرده و مسیر نسبی مجموعه داده (*hero-network.csv*) را گذر یا عبور می‌دهیم. براساس احضار یا فراخوانی این تابع، یک متغیر جدید به نام گراف تعریف می‌شود:

```
graph = graph_func("dataset_local_path")
```

حال تعداد گره‌ها و یالها را محاسبه می‌کنیم:

```
In:  graph.order()

Out: 6426

In:  graph.size()

Out: 167219

In:  graph.nodes()

Out:

     [u'QUESADA, JOE', u'ZEITGEIST II/AXEL CL', u'LOGA II',

     u'PIECEMEAL/GILBERT E', u'BENWAY, DR.', u'FRIEDLANDER, SHARON',

     u'DREAMING CELESTIAL/T', 'MODRED THE MYSTIC', u'THOMAS, DR. STAN',

     u'FIREWALL/THEARY/MIN ', u'DR. AUSTUS', u'RAYMOND, NORA JONES',

     u'DEATHRAVEN/JOSHUA RA', u'MARROW/SARAH ', u'EPOCH',…….]
```

بیایید نگاهی بیاندازیم به "تصویر لحظه‌ای"[۱] یالهایی که در شبکه وجود دارند. این تصویر نشاندهنده ارتباطات قهرمان–به–قهرمان بوده و حاکی از "شناخت" آنها از یکدیگر می‌باشد:

```
In: graph.edges()

Out:
```

[۱] Snapshot

```
[(u'QUESADA, JOE', u'ZURI'),
(u'QUESADA, JOE', u'GLADIATOR/MELVIN POT'),
(u'QUESADA, JOE', u'SMITH, KEVIN'),
(u'QUESADA, JOE', u'OKOYE'),
(u'QUESADA, JOE', u'RALF'),
(u'QUESADA, JOE', u'PALMIOTTI, JIMMY'),
(u'QUESADA, JOE', u'BLAKE, BECKY'),
(u'QUESADA, JOE', u'MYSTERIO/QUENTIN BEC'),
(u'QUESADA, JOE', u'WATSON-PARKER, MARY '),
(u'QUESADA, JOE', u'QUINN, ASHLEY'),
(u'QUESADA, JOE', u'MCKENZIE, LYDIA'),……]
```

این شبکه بزرگتر از آنست که برای مصور نمودنش بتوان تنها از کتابخانه *NetworkX*، استفاده کرد. با این حال برخی اطلاعات کلی شبکه را می‌توان با استفاده از روش *built-in nx.info()* در کتابخانه *NetworkX* محاسبه نمود:

```
In: nx.info(graph)
Out:
   'Name: Heroic Social Network\nType: Graph\nNumber of nodes:
   6426\nNumber of edges: 167219\nAverage degree: 52.0445'
```

ما قصد داریم شبکه کوچکتری را از شبکه قهرمانی قبلی (شبکه قهرمان–به–قهرمان) استخراج کنیم. شبکه مشتق شده جدید شامل "کتاب طنزی" است که در آن هر کاراکتر یا شخصیت ظاهر می‌شود. این کار به معنی ساخت یک شبکه "قهرمان–به–کمیک"[۱] است که نشاندهنده منبع ارتباط آنها است.

[۱] Hero-to-Comic

با بررسی کُمیکهایی که قهرمانان در آن، کنار یکدیگر ظاهر می‌شوند، می‌توان قدرت "شناخت" روابط را محاسبه نمود. این بدان معناست که اگر دو قهرمان در تعداد کمیکهای بیشتری با هم ظاهر شوند، با احتمال بالاتری یکدیگر را می‌شناسند.
کد زیر از این کتاب استخراج شده است:

```
def graph_from_gdf(path):
    graph = nx.Graph(name="Characters in Comics")
    with open(path, 'rU') as data:
       reader = csv.reader(data)
       for row in reader:
           if 'nodedef' in row[0]:
               handler = lambda row,G: G.add_node(row[0],
TYPE=row[1])
            elif 'edgedef' in row[0]:
               handler = lambda row,G: G.add_edge(*row)
            else:
               handler(row, graph)
     return graph
```

تابع را فرامی‌خوانیم و مسیر نسبی مجموعه داده‌ها (*comic-hero-network.gdf*) را انتقال می‌دهیم:

```
graph = graph_from_gdf("dataset_local_path")
In:   nx.info(graph)
      Out: 'Name: Characters in Comics\nType: Graph\nNumber of nodes:
      19090\nNumber of edges: 96104\nAverage degree: 10.0685'
In:   graph.order()
```

```
Out:  19090

In:   graph.size()

Out:  96104
```

اکنون قصد داریم یک زیرگراف را برای یک گره خاص (قهرمان) از شبکه "کمیک–به–قهرمان"[۱] استخراج کنیم. هدف ما بررسی آن چیزی است که در سطح فردی (فردانه) این شبکه، اتفاق می‌افتد. استخراج "گراف فردانه" از یک "شبکه بزرگتر"، یک پیاده سازی ساده در *NetworkX* دارد. این تابع سه آرگومان یا ورودی دارد: گراف اجتماعی گسترش‌یافته، گره فردانه و حداکثر طول مسیر.

```
ego_net = nx.ego_graph(graph, character, hops)
```

تعریف محدوده شبکه فردانه یک امر نسبی است یعنی بستگی به تعداد جهشها یا پرشهایی[۲] دارد که می‌خواهیم بعد از دور شدن از کنشگر اصلی یا فردانه داشته باشم (انجام هر پرش شامل پیمودن n گام در همسایگی فرد اصلی است).

```
In:  graph = graph_from_gdf("…/comic-hero-network.gdf")

In:  graph.order()

Out:  19090

In:  graph.size()

Out: 96104
```

فرض کنید که می‌خواهیم شبکه فردانه *"GRIFFIN II / JOHNNY HO"* را مطالعه کنیم و در ابتدا فقط همسایگان مستقیم[۳] (یا دگرانه‌های) آن قهرمان را در نظر بگیریم. تعداد پرشها در این مورد یک است (شکل ۸–۵).

```
In: ego_net = nx.ego_graph(graph, "GRIFFIN II/JOHNNY HO", 1)
```

---

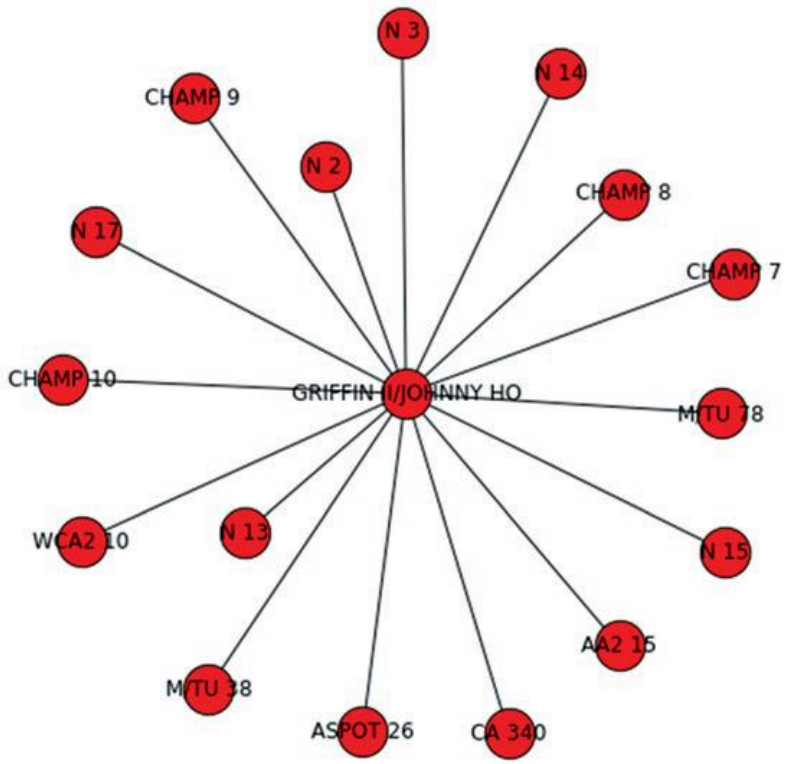
[۱] Comic-to-Hero Network

[۲] Hops

[۳] Direct Neighbors

حال اجازه دهید شبکه فردانه را رسم کنیم:

```
In: plt.figure(figsize=(10, 10))
    pos = nx.spring_layout(ego_net)
    nx.draw_networkx(ego_net, pos, with_labels=True,
    node_size=800,
    alpha=1, node_color='r')
```



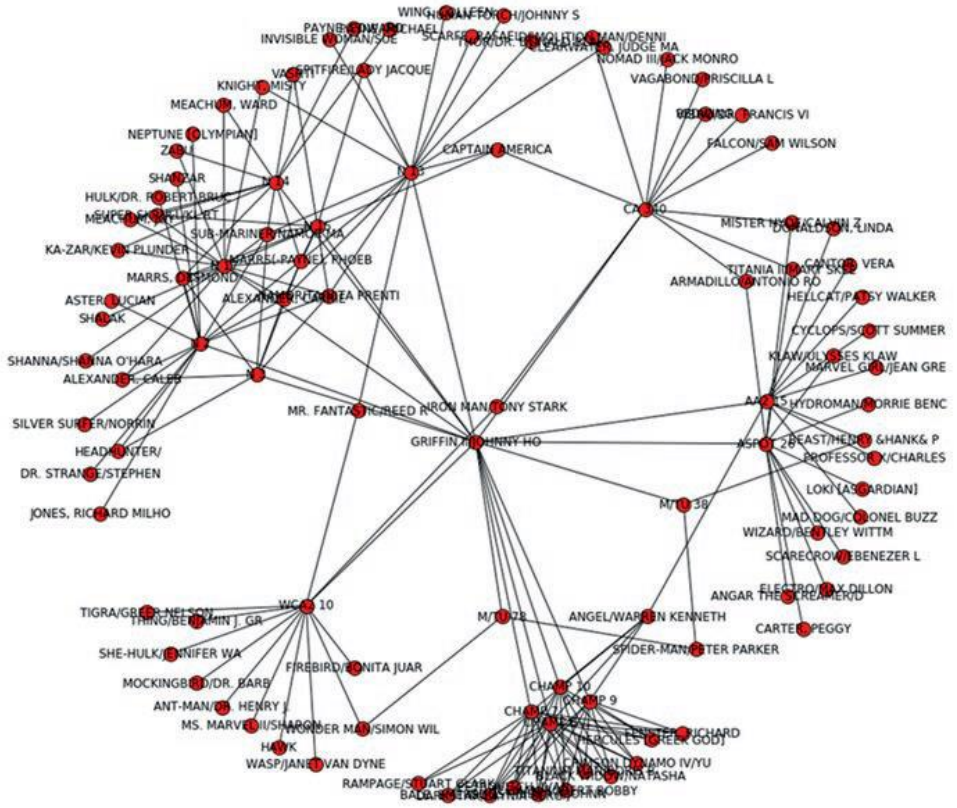
شکل ۸-۵ شبکه فردانه یک گره از گراف مارول جهانی

گراف تک پرش قبلی نشان می‌دهد که قهرمان جانی هو (که بعنوان گریفین دوم هم شناخته می‌شود) در مرکز گراف قرار دارد و به کمیکهایی که در آن ظاهر شده متصل است. به نظر می‌رسد که او در ۱۶

کُمیک ظاهر شده (*AA2 15*، *ASPOT 26*، *CA 340* و غیره) که این نتیجه را می‌توان با محاسبه تعداد گره‌ها و یالها نیز تایید نمود:

```
In:  ego_net.order()

Out: 17

In:  ego_net.size()

Out: 16
```

متناوباً، اگر بخواهیم به جای یک پرش، دو پرش را بررسی کنیم (به معنی بررسی و نفوذ در همسایگانِ همسایگانِ فردانه) (شکل ۹-۵):



شکل ۹-۵ شبکه فردانه یک گره با دو پرش

```
In: ego_net = nx.ego_graph(graph, "GRIFFIN II/JOHNNY HO", 2)


In: ego_net.order()
Out: 100


In: ego_net.size()
Out: 175


In: plt.figure(figsize=(20, 20))
    pos = nx.spring_layout(ego_net)
    nx.draw_networkx(ego_net, pos, with_labels=True,
    node_size=800,
    alpha=1, node_color='r')
```

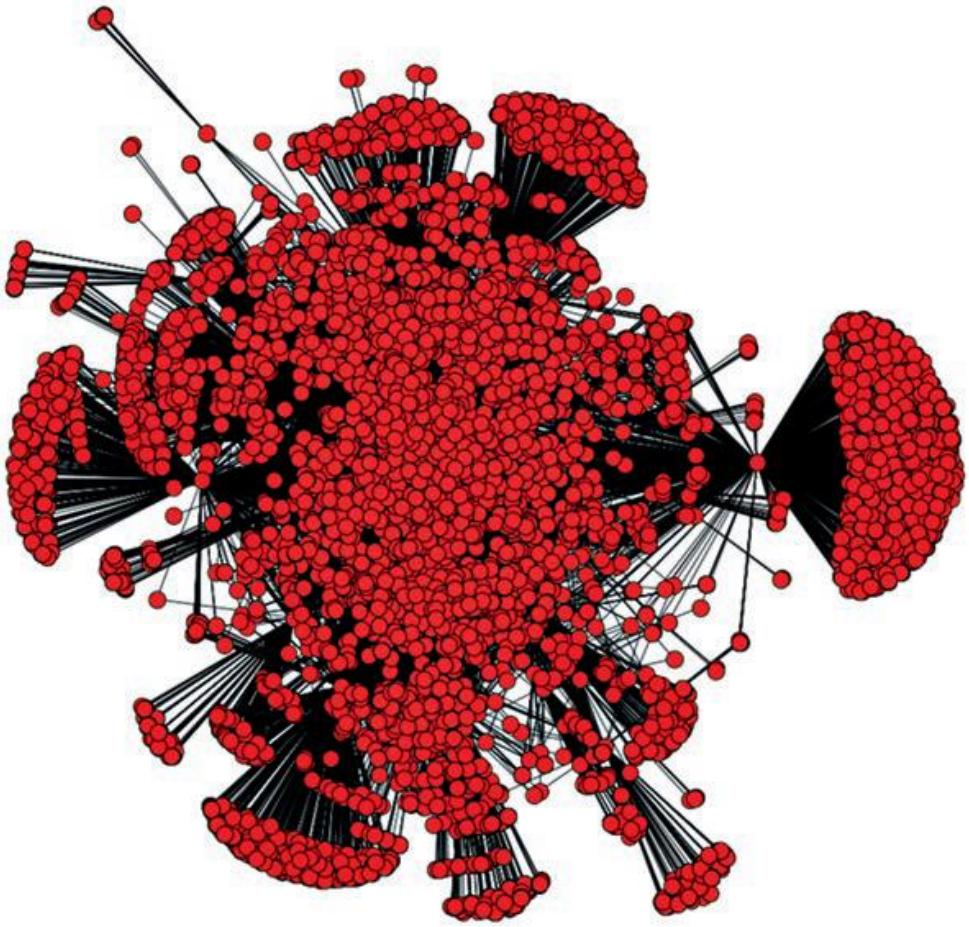
گراف دو پرشی شکل ۹–۵، حجم شبکه را با افزودن همسایگانِ همسایگان مستقیم گسترش می‌دهد. فردانه گریفین *II* همچنان در مرکز گراف قرار دارد.

اکنون بیایید وضعیت سه پرش را بررسی کنیم (شکل ۱۰–۵):

```
In:  ego_net = nx.ego_graph(graph, "GRIFFIN II/JOHNNY HO", 3)


In:  ego_net.order()
Out: 8163


In:  ego_net.order()
Out: 20797
```

```
In:  plt.figure(figsize=(20, 20))
     pos = nx.spring_layout(ego_net)
     nx.draw_networkx(ego_net, pos, with_labels=True,
     node_size=200,
     alpha=1, node_color='r')
```

نمودار سه پرشی شکل ۱۰–۵، به طور قابل توجهی اندازه شبکه را گسترش می‌دهد. این نمودار سه سطح (همسایه همسایه همسایه) را بررسی می‌کند. هرچند خوشه‌ها به وضوح آشکار می‌شوند، اما شبکه فردانه کمتر دیده می‌شود.



شکل ۱۰-۵ شبکه فردانهٔ یک گره با سه پرش

۵-۲ شناسایی افراد تاثیرگذار در شبکه

در تمام انواع شبکه‌ها، گره‌ها معمولاً از یکدیگر مستقل نیستند، بلکه آنها با یک یا چند یال بهم متصل می‌شوند. از آنجا که این گره‌ها همبند هستند، می‌توانند بر یکدیگر تاثیر بگذارند. با این مقدمه کوتاه،

حال یک سوال مطرح می‌شود: کدام گره‌ها، در میان تعداد زیادی از گره‌های همبند، اهمیت بیشتری دارند؟

معیار مرکزیت می‌تواند به چنین سؤالی پاسخ دهد. چنین معیاری اهمیت گره‌ها از دیدگاههای مختلف را بدست می‌آورد.

"معیار مرکزیت" یک ابزار مهم برای تحلیل گراف است که اهمیت نسبی یک گره خاص در یک گراف (یا اهمیت افراد در یک شبکه) را مشخص می‌کند. افرادی که در محیط‌هایی با محدودیت کمتر کار می‌کنند و شانس بیشتری نسبت به دیگران دارند، عملاً در یک موقعیت ساختاری مناسبتری قرار دارند. داشتن موقعیت مطلوب بدان معنی است که مثلاً فرد دارای فرصت بهتری برای انجام مبادلات و معاملات است یا اثرگذاری بیشتری دارد و یا به یک مقام و موقعیت ویژه به لحاظ تمکین و تسلیم و توجه از طرف افرادی با موقعیتهای ضعیفتر دست می‌یابد. اهمیت شناسایی بازیگران کلیدی در یک شبکه، در آگاهی یافتن از میزان فعال بودن هر کاربر است، زیرا کاربران فعال، روابط اجتماعی بیشتری با کاربران دیگر شبکه برقرار می‌کنند.

مرکزیت، اصطلاحی است که برای تعریف اهمیت یک گره استفاده می‌شود. شاخص مرکزیت یکی از ابزارهای پر استفاده برای کشف این موضوع است که یک فرد تا چه حد در مرکز یک شبکه قرار دارد. (این شبکه ممکن است یک باشگاه اجتماعی[۱] یا وب گسترده جهانی[۲] باشد). معیارهای مهم مرکزیت که بطور معمول استفاده می‌شوند عبارتند از: درجه‌ای، نزدیکی، بینابینی و بردار ویژه. اگر چه این معیارها مهم هستند، اما برای تجزیه و تحلیل یک شبکه اجتماعی، معیارهای مرکزیت معمولاً به تنهایی کافی نیستند، زیرا پیدا کردن بانفوذ ترین افراد (کاربران کلیدی) در یک شبکه نیاز به تجزیه و تحلیل ساختار شبکه نیز دارد.

معیارهای مرکزیت و ساختار شبکه می‌تواند هر دو برای درک ساختار و سازمان شبکه‌های مخرب[۳] و شناسایی جایگاه گره‌های مشکوک در این شبکه‌ها مورد استفاده قرار گیرند. لذا بر اساس این شناخت می‌توان در مسیر بی‌ثبات‌سازی[۴] و کاهش اثربخشی[۵] اینگونه شبکه‌های اجتماعی گام برداشت.

---

[۱] Social Club

[۲] World Wide Web

[۳] Malicious Networks

[۴] Destabilizing

[۵] Effectiveness

به منظور یافتن کاربران مهم در یک شبکه، معیارها یا سنجه‌هایی از جمله مرکزیت بردار ویژه، درجه‌ای، نزدیکی، و بینابینی توسعه داده شدند. تحقیقات گذشته نشان می‌دهد که در میان اینها، سه معیار نخست هبستگی[۱] بالایی دارند و نتایج مشابهی را در شناسایی مهمترین کاربران در یک شبکه نشان می‌دهند. بسیاری از اوقات، نتیجه محاسبه از طریق سه مورد از این چهار معیار (یا حتی همه این چهار معیار مرکزیت)، باعث می‌شود یک فرد (یا چند نفر) در میان ده نفرِ اولِ[۲] فهرست مرکزیت قرار گیرند.
در این بخش، ما این چهار معیار مرکزیت استاندارد را که نشانگر قدرت و اثرگذاری هستند، مد نظر قرار داده و بکارگیری آنها را برای چند مجموعه داده، بحث خواهیم کرد.

۵-۲-۱ مرکزیت درجه‌ای

مرکزیت درجه‌ای، یک گره با بیشترین درجه (یعنی بیشترین تعداد اتصالات) را بعنوان گره مرکزی شبکه در نظر می‌گیرد. مرکزیت درجه‌ای متمرکز بر تک تک گره‌ها است و به‌آسانی با شمارش تعداد یالهایی که یک گره دارد محاسبه می‌شود.
مرکزیت درجه‌ای معیاری برای اندازه‌گیری شهرت یا محبوبیت است و گره‌هایی را که می‌توانند اطلاعات را به سرعت در یک منطقه محلی منتشر کنند، شناسایی می‌کند. در فیسبوک، این معیار همان تعداد دوستان است. در توییتر نیز تعداد دوستان بوده و در یک سایت مانند *Reddit* ، شمارش تعداد آراء[۳] است.

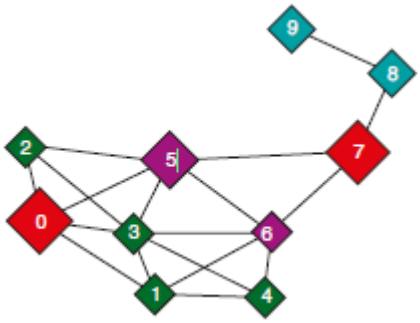
مثال ۱

در این مثال، گراف کایت کراکارت[۴] را تحلیل می‌کنیم. این گرافِ بادبادک مانند[۵] به‌خوبی در ادبیات شبکه‌های اجتماعی جهت نشان‌دادن تفاوتهای معیارهای مختلف مرکزیت مورد بررسی قرار گرفته است. در این مثال، یالها جهتدار نیستند که به معنای پذیرش یا توافق متقابل بین دو گره در دو سرِ یال است.

---

[۱] Correlated
[۲] Top Ten
[۳] Upvote
[۴] Krackhardt Kite Graph
[۵] Kite-Like

این گراف بادبادک شکل، شامل ده گره در دو گروه مختلف گره است. گروه اول، گره‌های متصل بهم شماره صفر تا شماره شش را داراست و گروه دوم، گره‌های زنجیروار شماره هفت تا شماره نه را دربرمی‌گیرد (شکل ۱۱-۵).



شکل ۱۱-۵ گراف کایت کراکارت

```
import networkx as nx
%matplotlib inline
import matplotlib.pyplot as plt
In: g = nx.krackhardt_kite_graph()
In: colors = ['r', 'g', 'g', 'g', 'g', 'm', 'm', 'r', 'c', 'c']
    sizes = [800, 300, 300, 300, 300, 600, 300, 800, 400, 400]
    labels = {0: '0', 1:'1', 2:'2', 3:'3', 4:'4', 5:'5', 6:'6',
    7:'7', 8:'8', 9:'9'}

In: nx.draw_networkx(g, node_color=colors,
   node_shape='D',with_labels=True,
   labels=labels,node_size=sizes)
   plt.show()
```

شکل شماره ۱۱–۵ نشان می‌دهد که گراف کایت کراکارت شامل یک هسته واحد بزرگ و یک حاشیه پیرامونی کوچک است.

کتابخانه *NetworkX* در پایتون، یک روش ساده برای "محاسبه مرکزیت درجه‌ای" دارد که درجه را بعنوان یک نگاشت پایتون[۱] باز می‌گرداند که در واقع منظور از نگاشت پایتون مجموعه‌ای از زوج مرتبهای گره–ارزش[۲] است، که شماره گره (یا نام گره، در موارد دیگر) را در کنار ارزش درجه‌ای آن نمایش می‌دهد:

```
In: nx.degree_centrality(g)

Out: {0: 0.4444444444444444,

     1: 0.4444444444444444,

     2: 0.3333333333333333,

     3: 0.6666666666666666,

     4: 0.3333333333333333,

     5: 0.5555555555555556,

     6: 0.5555555555555556,

     7: 0.3333333333333333,

     8: 0.2222222222222222,

     9: 0.1111111111111111}
```

اجازه دهید نتایج فوق را بصورت نزولی (یعنی از بیشترین مرکزیت درجه‌ای به کمترین آنها) بازآرایی کنیم:

```
In:  print(sorted(nx.degree_centrality(g).items(),key=itemgetter

     (1), reverse=True))
```

[۱] Python Map

[۲] Node-Value Pairs

```
Out:

[(3, 0.6666666666666666), (5, 0.5555555555555556), (6,
0.5555555555555556),

(0, 0.4444444444444444), (1, 0.4444444444444444), (2,
0.3333333333333333),

(4, 0.3333333333333333), (7, 0.3333333333333333), (8,
0.2222222222222222),

(9, 0.1111111111111111)]
```

توجه کنید که گره شماره ۳ بیشترین مقدار مرکزیت درجه‌ای را دارا می‌باشد، زیرا از حداکثر تعداد اتصالات (شش اتصال) برخوردار است. از سوی دیگر، گره شماره ۹ دارای کمترین مرکزیت درجه‌ای است زیرا تنها به یک گره دیگر متصل است.
همانطور که مشاهده می‌کنید، بازهٔ مربوط به فاصله میان حداقل و حداکثر درجه، تقریباً بالا است. بیایید توزیع درجه‌ای را بررسی کنیم (شکل ۱۲–۵):

```
In: degree = nx.degree(g)

    plt.hist(degree.values(), bins = 10)

    plt.title("Degree Values for the Krackhardt Kite Graph")

    plt.xlabel("Degree")

    plt.ylabel("Frequency")

    plt.show()

Out:
```

شکل ۱۲–۵ "توزیع درجه‌ای گراف" کایت کراکارت را نشان می‌دهد. همانگونه که مشاهده می‌کنید تعداد زیادی از گره‌ها در نزدیکی میانگین قرار گرفته و تعداد کمی از گره‌ها در دو طرف میانگین، توزیع

یافته‌اند. این وضع منجر، ما را به این نتیجه می‌رساند که توزیع فعلی درجات، به نحوی شبیه به توزیع نرمال[۱] است؛ یعنی متقارن بوده و دارای منحنی زنگوله‌ای[۲] با یک مقدار بیشینه است.

شکل ۱۲-۵ توزیع درجه‌ای گره‌ها در گراف کایت کراکارت

مثال ۲

در اینجا کار را بر روی مثال شبکه اجتماعی قهرمان[۳] ادامه می‌دهیم تا دریابیم که کدامیک از شخصیتها، کنشگران کلیدی در شبکه کاراکترهای کتابهای طنز[۴] هستند.

بیایید با محاسبه درجه گره مربوط به هر قهرمان در شبکه قهرمان–به–قهرمان کار را شروع کنیم و سپس ده گره با بیشترین درجه را انتخاب نماییم:

```
In: import operator

In: graph = graph_from_csv("…/hero-network.csv")

In: all_deg = sorted(graph.degree().items(),

key=operator.itemgetter(1), reverse=True)
```

[۱] Normal Distribution

[۲] Bell-Shaped Curves

[۳] Hero Social Network

[۴] Network of Comic Book Characters

```
In: for node in all_deg[0:10]: print node
Out:
     (u'CAPTAIN AMERICA', 1908)
     (u'SPIDER-MAN/PETER PAR', 1737)
     (u'IRON MAN/TONY STARK ', 1522)
     (u'THING/BENJAMIN J. GR', 1416)
     (u'MR. FANTASTIC/REED R', 1379)
     (u'WOLVERINE/LOGAN ', 1371)
     (u'HUMAN TORCH/JOHNNY S', 1361)
     (u'SCARLET WITCH/WANDA ', 1325)
     (u'THOR/DR. DONALD BLAK', 1289)
     (u'BEAST/HENRY &HANK& P', 1267)
```

روش دیگر برای محاسبه ده قهرمان برتر در شبکه قهرمان-به-قهرمان، استفاده از یک تابع مفید، *degree_centrality*، از کتابخانه *NetworkX* است که ما را قادر می‌سازد تا درصد گره‌های متصل به هر گره (یعنی نسبت تعدادگره‌های متصل به هر گره به تعداد کل گره‌های موجود) را محاسبه کند.

```
In: centrality = nx.degree_centrality(graph)
In: all_deg = sorted(centrality.items(), key=itemgetter(1),
reverse=True)
In: for node in all_deg[0:10:]: print "%s: %0.3f" % node
Out:
    CAPTAIN AMERICA: 0.297
    SPIDER-MAN/PETER PAR: 0.270
    IRON MAN/TONY STARK : 0.237
```

```
    THING/BENJAMIN J. GR: 0.220
    MR. FANTASTIC/REED R: 0.215
    WOLVERINE/LOGAN : 0.213
    HUMAN TORCH/JOHNNY S: 0.212
    SCARLET WITCH/WANDA : 0.206
    THOR/DR. DONALD BLAK: 0.201
    BEAST/HENRY &HANK& P: 0.197
```

برای محاسبه میانگین درجه شبکه قهرمان–به–قهرمان باید درجه هر گره را پیدا کرده، درجه‌ها را با هم جمع نموده و در نهایت نتیجه را بر تعداد کل گره‌های موجود در شبکه تقسیم کنیم:

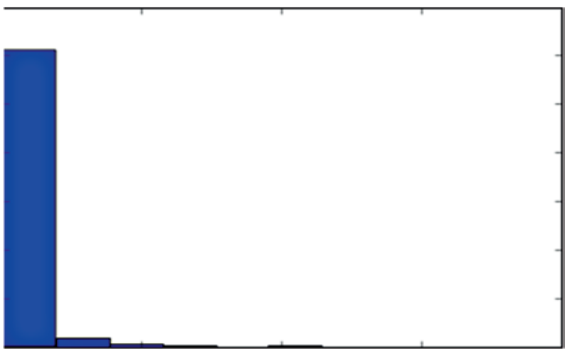
```
In:  degrees = graph.degree()
     sum_of_edges = sum(degrees.values())
In:  avg_deg = float(sum_of_edges/(graph.order()))
     print(avg_deg)
Out: 52.0
```

یک روش بهتر برای نشان دادن درجه گره استفاده از یک هیستوگرام است. *NetworkX* یک تابع به نام *degree_histogram* دارد. با این حال، این تابع یک فهرست طولانی از فراوانی یا فرکانسهای درجه‌ها را بازمی‌گرداند، که ممکن است، اطلاعات کافی ارائه ندهد.

```
In: nx.degree_histogram(graph)
Out:
    [0, 53, 86, 104, 150, 182, 178, 185, 230, 224, 245, 254,
    187, 194,60, 193, 123, 143, 123, 108, 131, 119,…]
```

یک راه بهتر می‌تواند استفاده از کتابخانه *matplotlib* در پایتون و به‌خصوص تابع *plt.hist ()* به شرح زیر باشد (شکل ۱۳-۵):

```
In:   plt.hist(graph.degree().values(), bins=500)

Out:

      plt.title("Degree Values of the Hero Network")

      plt.xlabel("Degree")

      plt.ylabel("Frequency")

      plt.show()
```



شکل ۱۳-۵ توزیع درجه‌ای شبکه قهرمان -به-قهرمان

۵-۲-۲ مرکزیت نزدیکی

برای تعیین گره‌های مرکزی در شبکه‌ها، می‌توان از معیار مرکزیت نزدیکی بهره برد. این معیار، همه گره‌هایی که با گره‌های دیگر پیوند دارند را بررسی کرده و از میان آنها، گره‌هایی که کوچکترین متوسط

طول مسیر[۱] (یا توالی روابط[۲]) دارند را بعنوان گره‌های مرکزی در نظر می‌گیرد. مرکزیت نزدیکی مهم است، زیرا نه‌تنها ارتباط بلاواسطه[۳] یک کنشگر با دیگر کنشگران را مورد توجه قرار می‌دهد (که از این بابت همانند معیار مرکزیت درجه‌ای است) بلکه ارتباطات غیرمستقیم[۴] تمام گره‌های دیگر در شبکه را نیز مد نظر قرار می‌دهد. در واقع معیاری برای اندازه‌گیری در دسترس‌بودن است، یعنی اینکه چگونه اطلاعات به‌سرعت از یک گره واحد به تمام گره‌های دیگر گسترش یابد.

مثال ۱

با استفاده از مثال گراف کایت کراکارت، می‌توان از کد زیر برای محاسبه مرکزیت نزدیکی تمام گره‌های موجود در گراف استفاده کرد:

```
In: nx.closeness_centrality(g)

Out: {0: 0.5294117647058824,

 1: 0.5294117647058824,

 2: 0.5,

 3: 0.6,

 4: 0.5,

 5: 0.6428571428571429,

 6: 0.6428571428571429,

 7: 0.6,

 8: 0.42857142857142855,

 9: 0.3103448275862069}

In: print(sorted(nx.closeness_centrality(g).items(), (1),
```

[۱] Smallest Average Path Length

[۲] Sequence of Relationships

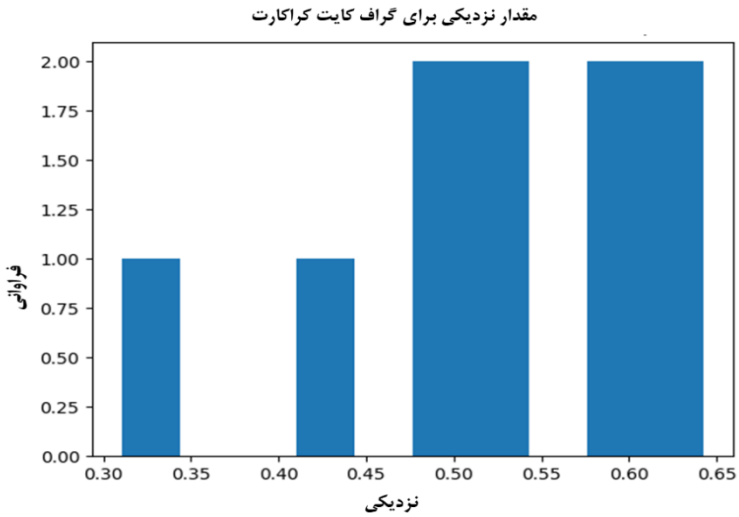
[۳] Immediate Connections

[۴] Indirect Ties

```
    key=itemgetter reverse=True))
Out:
    [(5, 0.6428571428571429), (6, 0.6428571428571429), (3, 0.6),
    (7,0.6), (0, 0.5294117647058824), (1, 0.5294117647058824),
    (2, 0.5),(4, 0.5),(8, 0.42857142857142855),
    (9, 0.3103448275862069)]
```

همانطور که مشاهده می‌کنید، گره‌های شماره ۳، ۵، و ۶ بیشترین مرکزیت نزدیکی را دارند، زیرا آنها در وسط شبکه حضور دارند و با تعداد گام کمی می‌توانند به گره‌های دیگر برسند. با این حال، چرا این گره‌ها چنین مرکزیت نزدیکی بالایی دارند؟ این وضعیت بدلیل موقعیت ساختاری آنهاست. آنها در هسته گراف کایت قرار دارند جایی که در آن ارتباطات بیشتری وجود دارد و بنابراین میانگین فاصله کوتاهتری دارند. می‌توان این موضوع را با یک گره، مانند گره شماره ۹ که در حاشیه و انتهای رشته گراف کایت واقع شده است، مقایسه کرد. این گره کمترین مقدار مرکزیت درجه‌ای و مرکزیت نزدیکی را دارد.
بیایید نگاهی به "توزیع مرکزیت نزدیکی" گراف کایت کراکارت بیندازیم (شکل ۱۴–۵).

```
In: closeness = nx. closeness_centrality(g)
   plt.hist(closeness.values(), bins = 10)
   plt.title("Closeness Values for the Krackhardt Kite Graph")
   plt.xlabel("Closeness")
   plt.ylabel("Frequency")
   plt.show()
Out:
```



شکل ١٤-٥ توزیع نزدیکی گره‌ها در گراف کایت کراکارت

همانطور که "توزیع مرکزیت نزدیکی" نشان می‌دهد، مرکزیت نزدیکی به‌شدت چوله نیست. باز هم فقط چند گره، دنبالهٔ گراف (یا دُم گراف) را تشکیل می‌دهند، در حالیکه بقیه گره‌ها یک منحنی زنگوله‌ای را نمایش می‌دهند که در انتهای پایین طیف، مقادیر کمتری دارند. توزیع فوق نشان می‌دهد که هسته گراف در این مثال، شامل گره‌هایی است که فاصلهٔ بین آنها بیش از دو قدم نیست.

مثال ۲

می‌توانیم "مرکزیت نزدیکی" گراف قهرمان-به-قهرمان را بررسی نموده و ده قهرمانی که دارای مرکزیت نزدیکی بالایی هستند را پیدا کنیم. قصد داریم این مقدار را در سه حالت مختلف محاسبه کنیم: بدون آرگومان، نرمال‌شده (برای تعداد گره‌های گراف) و وزندار به منظور اعمال وزن یالها.

```
In: close_cent = nx.betweenness_centrality(graph)

    norm_close_cent = nx.betweenness_centrality(graph,

    normalized=True)

    weigh_close_cent = nx.betweenness_centrality(graph,

    weight="weight")
```

روش نرمالسازی از دو روش دیگر سریعتر است، بنابراین در ادامه با استفاده از این روش، ده گره برتر را مشخص خواهیم کرد:

```
In: all_close = sorted(norm_close_cent.items(),
    key=itemgetter(1), reverse=True)
In: for node in all_close[0:10:]: print "%s: %0.3f" % node
Out:
    SPIDER-MAN/PETER PAR: 0.074
    CAPTAIN AMERICA: 0.057
    IRON MAN/TONY STARK : 0.037
    WOLVERINE/LOGAN : 0.036
    HAVOK/ALEX SUMMERS : 0.036
    DR. STRANGE/STEPHEN : 0.029
    THING/BENJAMIN J. GR: 0.025
    HAWK: 0.025
    HULK/DR. ROBERT BRUC: 0.024
    MR. FANTASTIC/REED R: 0.024
```

همانطور که مشاهده می‌شود؛ فهرست قهرمانان فوق، همان فهرستی است که در محاسبه مرکزیت درجه‌ای محاسبه شد.

۵-۲-۳ مرکزیت بینابینی

درحالیکه مرکزیتهای درجه‌ای و نزدیکی بر مبنای مفهوم قابلیت دسترسی اشخاص[۱] هستند، مرکزیت بینابینی، براین ایده استوار است که یک فرد در صورتی مهمتر است، که جایگاه او در شبکه، نقش میانجی‌گری و واسطه‌گری[۲] بیشتری داشته باشد. این معیار بر اساس فاصله ژئودزیک یا کروی است، به این معنی که یک کنشگر تنها وقتی می‌تواند نقش مهمی در شبکه ایفا کند که مابین (فاصله کروی) بسیاری از جفت گره‌های موجود در شبکه قرار گرفته باشد. گره‌هایی که در تعداد بیشتری از مسیرهای کوتاه بین گره‌های دیگر در گراف قرار داشته باشند، مرکزیت بینابینی بیشتری دارند.

مثال ۱

یک راه ساده برای پیاده‌سازی اندازه‌گیری مرکزیت بینابینی در مورد گراف کایت کراکارت در *NetworkX*، می‌تواند بصورت زیر می‌باشد:

```
In: nx.betweenness_centrality(g)

    Out: {0: 0.023148148148148143,

    1: 0.023148148148148143,

    2: 0.0,

    3: 0.10185185185185183,

    4: 0.0,

    5: 0.23148148148148148,

    6: 0.23148148148148148,

    7: 0.38888888888888884,

    8: 0.2222222222222222,

      9: 0.0}

In: print(sorted(nx.betweenness_centrality(g).items(), (1),

    key=itemgetter reverse=True))
```

[۱] Person's Reachability

[۲] Intermediary

```
Out:

    [(7, 0.38888888888888884), (5, 0.23148148148148148), (6,

    0.23148148148148148), (8, 0.2222222222222222), (3,

    0.10185185185185183), (0, 0.023148148148148143), (1,

    0.023148148148148143), (2, 0.0), (4, 0.0), (9, 0.0)]
```

همانطور که می‌بینید، گره شماره ۷ بیشترین "مرکزیت بینابینی" را داراست. این امر واضح است زیرا تنها گره‌ای می‌باشد که گره‌های حاشیه‌ای (گره‌های شماره ۸ و ۹) را به گره‌های موجود در هسته متصل می‌کند. از طرف دیگر، گره‌های ۹ و ۸ در مرز شبکه قرار دارند و در هیچ یک از کوتاهترین مسیرهای شبکه قرار ندارند.

مثال ۲

همچنین می‌توان مرکزیت بینابینی برای گراف قهرمان–به–قهرمان را بررسی کرد و ده قهرمانی که دارای مرکزیتهای بینابینی بیشتری هستند را مشخص نمود. ما قصد داریم این مقدار را در سه حالت مختلف محاسبه کنیم: بدون آرگومان، نرمال‌شده (برای تعداد گره‌های گراف) و وزندار (برای اعمال اوزان یالها).

```
In: bet_cent = nx.betweenness_centrality(graph)

    norm_bet_cent = nx.betweenness_centrality(graph,

    normalized=True)

    weigh_bet_cent = nx.betweenness_centrality(graph,

    weight="weight")

Out:
```

همانطور که برای محاسبه "مرکزیت درجه‌ای" برای ده گره برتر در شبکه عمل کردیم، اکنون می‌خواهیم "مرکزیت میانی" گره‌های برتر شبکه (را با استفاده از نسخه نرمال از این تابع) محاسبه کرده و بررسی کنیم که آیا نتایج مشابهی بدست می‌دهد یا خیر؟

```
In: all_bet = sorted(norm_bet_cent.items(), key=itemgetter(1),
reverse=True)
In: for node in all_bet[0:10:]: print "%s: %0.3f" % node
Out:
    SPIDER-MAN/PETER PAR: 0.074
    CAPTAIN AMERICA: 0.057
    IRON MAN/TONY STARK : 0.037
    WOLVERINE/LOGAN : 0.036
    HAVOK/ALEX SUMMERS : 0.036
    DR. STRANGE/STEPHEN : 0.029
    THING/BENJAMIN J. GR: 0.025
    HAWK: 0.025
    HULK/DR. ROBERT BRUC: 0.024
    MR. FANTASTIC/REED R: 0.024
```

شش قهرمان در این فهرست (*Spider-Man*، *Captain America*، *Iron Man*، *Wolverine*، *Thing / Benjamin*، و *Mr. Fantastic*) از جمله کسانی هستند که در فهرست ده قهرمان با مرکزیت درجه‌ای بالا ظاهر شدند. با این حال، سه قهرمان برتر در فهرست قبلی (فهرست مربوط به نتایج مرکزیت نزدیکی) هنوز در سه موقعیت اول فهرست فعلی حضور دارند.

۵-۲-٤ مرکزیت بردار ویژه

این معیار متناسب با مجموع امتیازات مرکزیت همسایگان هر گره می‌باشد (رتبه–صفحه، بعنوان مثال، یک رتبه‌بندی بر اساس بردار ویژه تصادفی[۱] است). این سنجه، معیاری از اثرگذاری وابسته[۲] است: چه کسی نزدیکتر به پرنفوذترین افراد در گراف است؟ به عبارت دیگر، وقتی یک گره مهم است که به گره‌های مهم دیگر متصل باشد. همچنین بدین معناست که یک گره مرتبط با تعداد کمی از گره‌های متنفذ (و با تاثیرگذاری بالا)، ممکن است رتبه و امتیاز بیشتری نسبت به یک گرهِ مرتبط با تعداد زیادی از گره‌های با نفوذ متوسط، داشته باشد. به عبارت دیگر، افراد خوش–ارتباط (با ارتباطات قویتر)[۳] در مقایسه با افراد کم–ارتباط (با ارتباطات ضعیفتر)[۴]، ارزش بیشتری دارند.

مثال ۱

می‌توانیم تابع *nx.eigenvector_centrality (g)* از کتابخانه *NetworkX* را برای "محاسبه معیار مرکزیت بردار ویژه" در گراف کایت کراکارت استفاده کنیم. این معیار نه تنها همسایگان مستقیم هر گره (همانند معیار مرکزیت درجه‌ای)، بلکه ساختار کلی شبکه را نیز بررسی می‌کند.

```
In: nx.eigenvector_centrality(g)

Out: {0: 0.3522091841983857,

       1: 0.3522091841983857,

       2: 0.28583482369644964,

       3: 0.4810206692001181,

       4: 0.28583482369644964,

       5: 0.39769090281372055,

       6: 0.39769090281372055,
```

[۱] Stochastic Eigenvector Scoring

[۲] Related Influence

[۳] Well-Connected

[۴] Ill-Connected

```
        7: 0.19586101425312444,
        8: 0.04807425308073237,
        9: 0.011163556091491361}
In:
        print(sorted(nx.eigenvector_centrality(g).items(),
        key=itemgetter(1) , reverse=True))
Out:
        [(3, 0.4810206692001181), (5, 0.39769090281372055), (6,
        0.39769090281372055), (0, 0.3522091841983857), (1,
        0.3522091841983857), (2, 0.28583482369644964), (4,
        0.28583482369644964), (7, 0.19586101425312444), (8,
        0.04807425308073237), (9, 0.011163556091491361)]
```

همانطور که مشاهده می‌شود، گره شماره ۳ بیشترین مرکزیت بردار ویژه را دارا می‌باشد، در حالیکه گره شماره ۹ کمترین مقدار را دارد. یکی دیگر از مشاهدات در اینجا این است که گره‌های شماره ۵ و ۶ که رتبه دوم و سوم را دارا هستند به ترتیب، در میان سه نفر برتر در معیارهای مرکزیت قبلی (درجه‌ای، نزدیکی و بینابینی) بودند. یعنی آنها در موقعیتی قرار گرفته‌اند که بعنوان مهره‌های مرزی یا پل ارتباطی شناخته می‌شوند. آنها بین دو بخش گراف (هسته و حاشیه) قرار دارند اما عضو کامل هیچ یک از آنها نیستند. هر دوی آنها به گره ۷، یعنی گارد ورودی به حاشیه و به گره ۳، یعنی گره‌ای با بالاترین مرکزیت درجه‌ای، متصل هستند.

برای یک سازمان، حضور افرادی با مشخصهٔ مهره مرزی (پل–مانند) ضروری است. این افراد دارای ارتباطات قابل توجهی در درون سازمان و خارج از آن هستند. آنها می‌توانند تصمیمات داخلی را تحت تاثیر قرار داده و همچنین سازمان را به محیط خارجی مرتبط کنند. مهره‌های مرزی یا پلها، توانایی تسهیل جریان ترافیک اطلاعات را داشته و می‌توانند برای سازمان، اطلاعات محیطی را تامین کنند؛ اطلاعاتی که می‌تواند شامل جنبه‌های حیاتی برای بقا و رشد سازمان باشد. این سازمانها به نوبه خود

می‌توانند به این اطلاعات پاسخ داده و بر این اساس عمل کنند. به‌همین دلیل است که مهره‌های مرزی گاهی اوقات افراد دنیادیده[۱] نامیده می‌شود؛ زیرا دیگران در صورت پیوند داشتن با آنها، با دنیای خارج مرتبط می‌شوند.

مثال ۲

در این مثال می‌خواهیم با استفاده از "مرکزیت بردار ویژه"، مجدداً گراف قهرمان-به-قهرمان را بررسی کرده و ده قهرمانی که دارای بیشترین مرکزیت بردار ویژه هستند، را پیدا کنیم. یکی از پیاده‌سازیهای موجود برای انجام این کار، *nx.eigenvector_centrality* در کتابخانه *NetworkX* می‌باشد، اگرچه پیاده‌سازی دیگری نیز وجود دارد که بصورت *nx.eigenvector_centrality_number_numpy* است. روش اول به‌دلیل تعداد ثابت تکرارها، برای حل مقدار واقعی سریعتر از روش *numpy* است و برای نمودارهای بزرگ مناسبتر می‌باشد.

```
In: eigen_cent = nx.eigenvector_centrality(graph)

In: all_eigen = sorted(eigen_cent.items(), key=itemgetter(1),

reverse=True)

In: for node in all_eigen[0:10:]: print "%s: %0.3f" % node

Out:

    CAPTAIN AMERICA: 0.117

    IRON MAN/TONY STARK : 0.103

    SCARLET WITCH/WANDA : 0.101

    THING/BENJAMIN J. GR: 0.101

    SPIDER-MAN/PETER PAR: 0.100

    MR. FANTASTIC/REED R: 0.100

    VISION : 0.099
```

[۱] Cosmopolitans

```
 HUMAN TORCH/JOHNNY S: 0.099
WOLVERINE/LOGAN : 0.098
BEAST/HENRY &HANK& P: 0.095
```

جالب است که در این فهرست *Captain America* و *Iron Man* هنوز در سه موقعیت اول قرار دارند. با این حال، *Spider-Man* موقعیت خود را از دست داده و در جایگاه پنجم قرارگرفته است. یک قهرمان جدید، یعنی *Vision*، بعنوان یک ورودی جدید در موقعیت هفتم ظاهر شده است.

۵-۳ رتبه-صفحه

الگوریتم رتبه-صفحه بر اساس تکنیک رتبه‌بندی صفحه[۱] در موتور جستجوی گوگل است. این رتبه، نشاندهنده مقدار احتمالی است که یک کاربر با کلیک کردن تصادفی بر لینکهای پیاپی و یا بهم مرتبط (بعنوان مثال، به هنگام گشت و گذار در وب یا وبگردی[۲]) ممکن است به یک صفحه خاص وارد شود. امتیازات رتبه-صفحه بین ۰ و ۱ می‌باشد. بعنوان مثال، رتبه-صفحه ۰,۷۵ به این معنی است که به احتمال ۷۵٪ ممکن است یک فرد با کلیک بر روی یک لینک تصادفی به سندی با رتبه-صفحه با مقدار ۰,۷۵ هدایت شود. از این رو، یک صفحهٔ وب با احتمال بیشتر، مهمتر از یک صفحهٔ وب با احتمال کمتر است. تفاوت اصلی بین رتبه-صفحه و سه معیار مهم مرکزیت (یعنی مرکزیتهای درجه‌ای، نزدیکی و بینابینی) این است که این سه معیار اخیر، نوعاً مفاهیمی بر پایه گراف هستند، لذا به‌شدت وابسته به موقعیتهای ساختاری گره‌ها[۳] می‌باشند؛ درحالیکه معیار رتبه-صفحه نشاندهنده جریان اثرگذاری یا نفوذی[۴] است که از طریق شبکه، انتشار می‌یابد. گرچه رتبه-صفحه از برخی جنبه‌ها با مرکزیت بردار ویژه مشابهت دارد ولیکن این الگوریتم عملکرد بسیار خوبی در شبکه‌های بسیار بزرگ و پویا دارد.

---

[۱] Page Ranking Technique

[۲] Surfing the Web

[۳] Structural Positions of Nodes

[۴] Flow of Influence

مثال

در این مثال، قصد داریم یک شبکه توییتر ساختگی (فرضی)[۱] با ۱۵ نفر ایجاد کنیم. سپس یالها را اضافه کرده، شبکه را تحلیل نموده و در نهایت آن را ذخیره می‌کنیم. ابتدا کتابخانه‌های لازم را باید وارد نماییم:

```
In: import networkx as nx
    import networkx.algorithms as alg
    import numpy as np
    import matplotlib.pyplot as plt
```

سپس، شئ گراف را ایجاد می‌کنیم:

```
In: graph = nx.DiGraph()
```

اکنون، ۱۵ گره به نمایندگی از ۱۵ نفر، به همراه سن آنها و تعداد پستهای هر فرد در شبکه را اضافه می‌کنیم:

```
In: graph.add_node('Will', {'age': 22, 'posts': 75})
    graph.add_node('Linda', {'age': 30, 'posts': 12})
    graph.add_node('Smith', {'age': 33, 'posts': 207})
    graph.add_node('Mark', {'age': 29, 'posts': 382})
    graph.add_node('Bill', {'age': 29, 'posts': 107})
    graph.add_node('Jack', {'age': 32, 'posts': 372})
    graph.add_node('Rose', {'age': 19, 'posts': 71})
    graph.add_node('Peter', {'age': 28, 'posts': 111})
    graph.add_node('Hilda', {'age': 27, 'posts': 75})
    graph.add_node('Mary', {'age': 27, 'posts': 56})
```

[۱] Fabricated Twitter Network

```
graph.add_node('Glenn', {'age': 34, 'posts': 89})
graph.add_node('Rick', {'age': 22, 'posts': 121})
graph.add_node('George', {'age': 31, 'posts': 43})
graph.add_node('Markus', {'age': 24, 'posts': 113})
graph.add_node('Alex', {'age': 32, 'posts': 67})
```

حال شبکه پیروان یا دنبالگران را تشکیل می‌دهیم که شامل اطلاعات "شناخت" است، یعنی: "چه‌کسی چه کسی را می‌شناسد". برای این منظور از روش *nx.add_edge ()* استفاده می‌کنیم. به‌علاوه برای تعیین قدرت هر رابطه از ابرداده[۱] "وزن" استفاده می‌کنیم:

```
In:
graph.add_edge('Will', 'Alex', {'Weight': 3,
'relationship': 'friend'})
graph.add_edge('Will', 'Markus', {'Weight': 2,
'relationship': 'friend'})
graph.add_edge('Will', 'George', {'Weight': 1,
'relationship': 'friend'})
graph.add_edge('Linda', 'Alex', {'Weight': 5,
'relationship': 'friend'})
graph.add_edge('Linda', 'Markus', {'Weight': 4,
'relationship': 'friend'})
graph.add_edge('Linda', 'George', {'Weight': 1,
'relationship': 'friend'})
graph.add_edge('Smith', 'Alex', {'Weight': 3,
```

[۱] Metadata

```
'relationship': 'friend'})
graph.add_edge('Smith', 'Markus', {'Weight': 4,
'relationship': 'friend'})
graph.add_edge('Smith', 'George', {'Weight': 4,
'relationship': 'friend'})
graph.add_edge('Mark', 'Alex', {'Weight': 2,
'relationship': 'friend'})
graph.add_edge('Mark', 'Markus', {'Weight': 2,
'relationship': 'friend'})
graph.add_edge('Mark', 'George', {'Weight': 5,
'relationship': 'friend'})
graph.add_edge('Bill', 'Alex', {'Weight': 3,
'relationship': 'friend'})
graph.add_edge('Bill', 'Markus', {'Weight': 2,
'relationship': 'friend'})
graph.add_edge('Bill', 'George', {'Weight': 1,
'relationship': 'friend'})
graph.add_edge('Jack', 'Alex', {'Weight': 1,
'relationship': 'friend'})
graph.add_edge('Jack', 'Markus', {'Weight': 2,
'relationship': 'friend'})
graph.add_edge('Jack', 'George', {'Weight': 1,
'relationship': 'friend'})
```

```
graph.add_edge('Rose', 'Alex', {'Weight': 4,
'relationship': 'friend'})
graph.add_edge('Rose', 'Markus', {'Weight': 4,
'relationship': 'friend'})
graph.add_edge('Rose', 'George', {'Weight': 5,
'relationship': 'wife'})
graph.add_edge('Peter', 'Alex', {'Weight': 2,
'relationship': 'friend'})
graph.add_edge('Peter', 'Markus', {'Weight': 2,
'relationship': 'friend'})
graph.add_edge('Peter', 'George', {'Weight': 4,
'relationship': 'friend'})
graph.add_edge('Hilda', 'Alex', {'Weight': 2,
'relationship': 'friend'})
graph.add_edge('Hilda', 'Markus', {'Weight': 2,
'relationship': 'wife'})
graph.add_edge('Hilda', 'George', {'Weight': 1,
'relationship': 'friend'})
graph.add_edge('Mary', 'Alex', {'Weight': 4,
'relationship': 'friend'})
graph.add_edge('Mary', 'Markus', {'Weight': 5,
'relationship': 'friend'})
graph.add_edge('Mary', 'George', {'Weight': 4,
```

```
'relationship': 'friend'})
graph.add_edge('Glenn', 'Alex', {'Weight': 3,
'relationship': 'friend'})
graph.add_edge('Glenn', 'Markus', {'Weight': 3,
'relationship': 'friend'})
graph.add_edge('Glenn', 'George', {'Weight': 1,
'relationship': 'friend'})
graph.add_edge('Rick', 'Alex', {'Weight': 5,
'relationship': 'friend'})
graph.add_edge('Rick', 'Markus', {'Weight': 2,
'relationship': 'friend'})
graph.add_edge('Rick', 'George', {'Weight': 1,
'relationship': 'friend'})
```

برای بازیابی فهرست گره‌ها در گراف، روش *nx.nodes ()* برای هر پارامتر را به آسانی اجرا کنید:

```
In:  graph.nodes()
Out: ['Rick', 'Alex', 'Glenn', 'Rose', 'Bill', 'Smith',
     'Mark', Will', 'Jack', 'George', 'Markus', 'Linda', 'Peter',
     'Mary', 'Hilda']
```

از طرف دیگر، اجرای *nx.nodes (data = True)*، ابر دادهٔ هر گره را باز می‌گرداند:

```
In: graph.nodes(data = True)
Out:
[('Rick', {'age': 22, 'posts': 121}),
```

```
  ('Alex', {'age': 32, 'posts': 67}),
  ('Glenn', {'age': 34, 'posts': 89}),
  ('Rose', {'age': 19, 'posts': 71}),
  ('Bill', {'age': 29, 'posts': 107}),
  ('Smith', {'age': 33, 'posts': 207}),
  ('Mark', {'age': 29, 'posts': 382}),
  ('Will', {'age': 22, 'posts': 75}),
  ('Jack', {'age': 32, 'posts': 372}),
  ('George', {'age': 31, 'posts': 43}),
  ('Markus', {'age': 24, 'posts': 113}),
  ('Linda', {'age': 30, 'posts': 12}),
  ('Peter', {'age': 28, 'posts': 111}),
  ('Mary', {'age': 27, 'posts': 56}),
  ('Hilda', {'age': 27, 'posts': 75})]
In:  graph.edges()
Out: [('Rick', 'Markus'), ('Rick', 'Alex'), ('Rick', 'George'),
     ('Alex',Glenn'), ('Alex', 'Rose'), ('Alex', 'Bill'),
     ('Alex', 'Smith'),('Alex', Mark'), ('Alex', 'Will'),
     ('Alex', 'Jack'),…]
In:  graph.edges(data=True)
Out:
     [('Rick', 'Markus', {'Weight': 2, 'relationship':
      'friend'}),('Rick', 'Alex', {'Weight': 5, 'relationship':
```

```
      'friend'}),('Rick', 'George', {'Weight': 1,'relationship':
      'friend'}),('Alex', 'Glenn', {'Weight': 3, 'relationship':
      'friend'}),('Alex', 'Rose', {'Weight': 4, 'relationship':
      'friend'}),('Alex', 'Bill', {'Weight': 3, 'relationship':
      'friend'}),('Alex', 'Smith', {'Weight': 3, 'relationship':
      'friend'}),('Alex', 'Mark', {'Weight': 2, 'relationship':
      'friend'}),…]
In:   print"Density of the graph: ", nx.density(graph)

Out:  Density of the graph: 0.171428571429
```

نتیجه نشان می‌دهد که شبکه توییتر ساخته شده ما تُنُک است. این وضعیت بدان دلیل است نسبت یالهای موجود به یالهای ممکن، عدد پایینی است. در حالیکه مجموع تعداد کل یالهای ممکن در گراف (که با رابطه $n * (n-1) / 2$ سنجیده می‌شود) برابر با ۱۰۵ یال است، اما ما در اینجا تنها ۳۶ یال در گراف داریم. در واقع اتصالات موجود، حدود ۳۴٫۳٪ از همهٔ اتصالات ممکن است.

```
In:   print(nx.info(graph))

Out:  Name:
      Type: DiGraph
      Number of nodes: 15
      Number of edges: 36
      Average in degree: 2.4000
      Average out degree: 2.4000
```

اکنون بیایید "مرکزیت درجه‌ای گره‌ها" را در نمودار محاسبه کنیم. درجه هر گره، تعداد اتصالات مستقیم آن گره می‌باشد. کتابخانه *NetworkX* دارای تابعی بنام *nx.degree_centrality ()* است که نسبت درجه هر گره به حداکثر تعداد ممکن اتصالات گراف (یعنی تعداد کل یالهای ممکن) را محاسبه می‌کند:

```
In:  nx.degree_centrality(graph)
Out: {'Alex': 0.8571428571428571, 'Bill': 0.21428571428571427,
     'George':.8571428571428571, 'Glenn': 0.21428571428571427,
     'Hilda':.21428571428571427, 'Jack': 0.21428571428571427,
     'Linda':.21428571428571427, 'Mark': 0.21428571428571427,
     'Markus':.8571428571428571, 'Mary': 0.21428571428571427,
     'Peter':.21428571428571427,…}
```

اگر بخواهیم درجه‌ها را از بیشترین مقدار به کمترین مقدار مرتب کنیم، آنگاه خواهیم داشت:

```
In: centrality = sorted(nx.degree_centrality(graph).items(),
    key=lambda e: e[1], reverse=True)
    print(centrality)
Out: [('Alex', 0.8571428571428571), ('Markus',
     0.8571428571428571),('George', 0.8571428571428571),
     ('Mary', 0.21428571428571427),('Glenn',
     0.21428571428571427), ('Rose', 0.21428571428571427),
      ('Bill', 0.21428571428571427), ('Smith',
      0.21428571428571427),('Mark', 0.21428571428571427),
        ('Will', 0.21428571428571427),('Rick',
      0.21428571428571427), ('Linda', 0.21428571428571427),
```

```
('Hilda', 0.21428571428571427), ('Peter',
0.21428571428571427), ('Jack', 0.21428571428571427)]
```

همانطور که مشاهده می‌شود، الکس، مارکوس و جُرج سه نفر از کسانی هستند که در گراف مورد مطالعه، بیشترین ارتباط را دارند.

تابع دیگری از *NetworkX* مثل *nx.assortativity.average_neighbor_degree ()* وجود دارد که برای هر گره، متوسط درجه همسایگان آنرا محاسبه می‌کند. با واژگان شبکه‌های اجتماعی[۱]، چنین معیاری امکان یافتن کسانی را می‌دهد که با متصلترین افراد شبکه، دوست هستند:

```
In: assortativity =
    sorted(nx.assortativity.average_neighbor_degree(graph).
    items(),key=lambda e: e[1], reverse=True)
    print(assortativity)
Out: [('Peter', 12.0), ('Glenn', 12.0), ('Rose', 12.0), ('Bill',
    12.0),('Smith', 12.0), ('Mark', 12.0), ('Will', 12.0),
    ('Rick', 12.0),('Linda', 12.0), ('Hilda', 12.0), ('Mary',
    12.0), ('Jack', 12.0),('Alex', 3.0), ('George', 3.0),
    ('Markus', 3.0)]
```

به نظر می‌رسد که پیتر، گلن، رز، بیل، اسمیت، مارک، ویل، ریک، لیندا، هیلدا، مری و جک همه با افرادی با بالاترین درجه در شبکه یعنی الکس، مارکوس و جُرج ارتباط دارند:

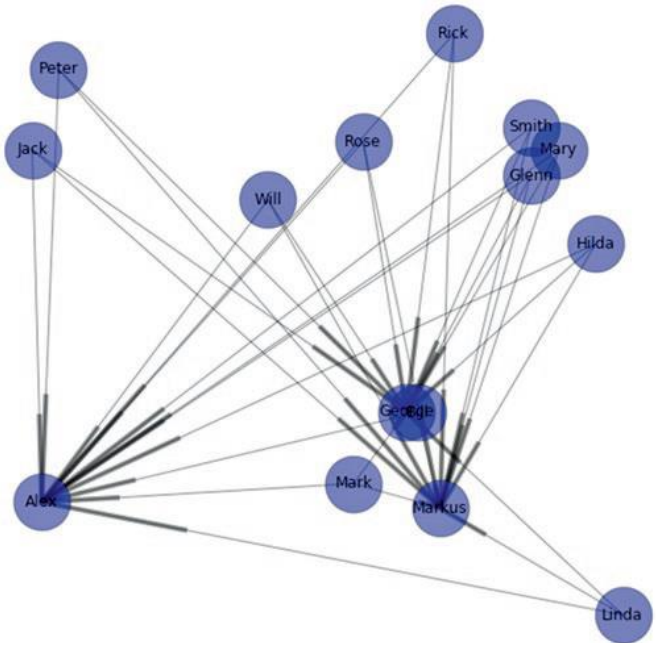
جهت رسم گراف (شکل ۱۵-۵):

```
In: plt.figure(figsize=(12,12))
    layout = nx.random_layout(graph)
    nx.draw_networkx(graph, layout, node_color='c',
```

[۱] Social Networking Terminology

```
    node_size=1000, alpha = 0.4)
```

Out:



شکل ۱۵-۵ ارزش رتبه-صفحه یک شبکه توییتر ساخته شده با ۱۵ نفر

با اجرای الگوریتم برای مجموعه داده‌های توییتر ساختگی، نتایج مشابهی به دست می‌آیند؛ محاسبه مرکزیت درجه‌ای:

```
In:  nx.pagerank(graph)
```

Out:

```
    {'Alex': 0.17460386794062185,
    'Bill': 0.03968236634817788,
    'George': 0.17460386794062185,
    'Glenn': 0.03968236634817788,
```

```
    'Hilda': 0.03968236634817788,
    'Jack': 0.03968236634817788,
    'Linda': 0.03968236634817788,
    'Mark': 0.03968236634817788,
    'Markus': 0.17460386794062185,
    'Mary': 0.03968236634817788,
    'Peter': 0.03968236634817788,
    'Rick': 0.03968236634817788,
    'Rose': 0.03968236634817788,
    'Smith': 0.03968236634817788,
    'Will': 0.03968236634817788}
```

می‌توان گراف را در فرمت *GraphML* ذخیره نمود که توسط بسته تصویرسازی *Gephi* قابل خواندن است:

```
In: nx.write_graphml(graph, "…/graph.graphml")
```

بطور جایگزین، می‌توان نتیجه را در فرمت فایل *png* نیز ذخیره نمود:

```
In: plt.savefig("…/graph.png")
```

۵-۴ همسایگان

```
In: G = nx.Graph() # or DiGraph, MultiGraph, MultiDiGraph, etc
    G.add_path([0,1,2,3])
    G.neighbors(0)
```

۵-۵ پلها

گرانووتر[۱] در سال ۱۹۷۳ مقاله‌ای با عنوان" قدرت پیوندهای ضعیف"[۲] منتشر نمود که در آن بر اهمیت "پلهای شبکه" برای انتشار اطلاعات در شبکه‌های اجتماعی تاکید نمود. استدلال اصلی گرانووتر این بود که بدلیل اینکه دوستان افراد، احتمالاً خودشان باهم دوست هستند پس اگر ما با "افرادی که یکدیگر را نمی‌شناسند" ارتباط برقرار کنیم آنگاه احتمالاً پلهایی را ایجاد می‌کنیم که گروههای منفصل را بهم متصل می‌نماید. اهمیت پلها در این است که آنها فاصله کلی بین گره‌ها در یک شبکه را کاهش می‌دهند که به نوبه خود سرعت گسترش اطلاعات در سراسر شبکه را افزایش می‌دهند.

در حالیکه اکثر تحلیلگران شبکه به شناسایی کاربران کلیدی و کسانی که در مرکز شبکه قرار دارند می‌پردازند، ولیکن تعیین "پلهای شبکه" می‌تواند در برخی مواقع مهمتر باشد، هرچند اغلب چالشهای بسیاری هم دارد. پلها بر اساس تعریف، پیوندهایی هستند که دو گروه جداگانه را متصل می‌کنند و لنگر یا سرپل[۳] این پیوندها، همان گره‌های پلی یا مرزی هستند. افرادی که بعنوان پل عمل می‌کنند، معمولاً ویژگیهای شخصیتی خاصی دارند، از جمله اینکه آنها از شنیدن ایده‌های جدید دیگران استقبال می‌کنند. "درجه پل بودن هر گره انفرادی" به صورت زیر تعریف می‌شود:

۱-۵

$$B_i = \frac{\sum_{j=1}^{k}\left[C - C_{ij}^{'}\right]}{k} + \frac{\sum_{j=1}^{l}\left[C - C_{ji}^{'}\right]}{l} \; (i \neq j)$$

که در آن، C انسجام شبکه مشاهده شده است، $C_{ij}^{'}$ انسجام شبکه در زمانی است که پیوندهای i به j حذف می‌شوند و $C_{ji}^{'}$ انسجام در زمانی است که پیوندها از j به i حذف می‌شوند، و k و l به ترتیب تعداد پیوندهای ارسالی و دریافتی هستند.

در حالیکه همواره از "معیار مرکزیت بینابینی" برای پیدا کردن گره‌ها با موقعیت پل، استفاده می‌شود، "معیار پل بودن" برای یافتن آن دسته از گره‌هایی که عملاً حاشیه‌ای یا مرکزی نیستند، اما موقعیت

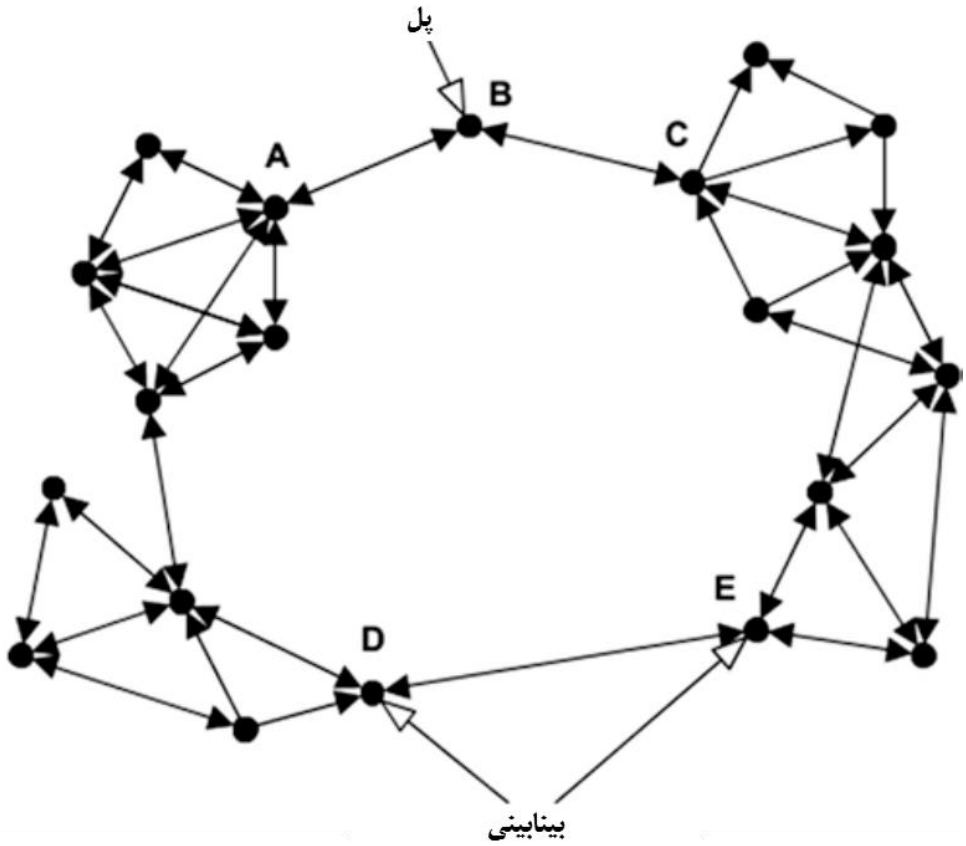
---

[۱] Granovetter

[۲] The Strength of the Weak Ties

[۳] Anchors

واسطه‌ای خوبی در داخل شبکه دارند، مهم است. این گره‌ها ممکن است بعنوان رله‌ها یا واسطه‌های[۱] مهمی در فرآیند انتشار، عمل کنند، بنابراین باید تقویت شوند تا شبکه انسجام یابد. معهذا، هنوز هم بین این دو معیار، تا حدی همبستگی وجود دارد به طوری که آنها می‌توانند گره‌های مشابهی را زمانی که ساختار شبکه یکنواخت است، شناسایی کنند. شکل ۱۶–۵ موقعیت گره‌های بینابینی و پل–مانند را در یک شبکه نشان می‌دهد.



شکل ۱۶–۵ موقعیت گره‌های بینابینی و گره‌های پل–مانند

[۱] Relays

در شکل ۱۶-۵، گره B در بالای شبکه یک گره پل است، در حالی که D و E گره‌های بینابینی هستند. تفاوت بین دو مورد فوق این است که گره B تنها دو پیوند دارد و اگر هر یک از این دو پیوند حذف شود، میانگین فاصله‌ها در سایر گره‌ها بیشتر می‌شود. از سوی دیگر، گره‌های D و E (در میان گره‌های دیگر شبکه) دارای مقادیر بینابینی بالا و تقریباً برابر هستند.

۵-۶ از کدام الگوریتم مرکزیت استفاده شود؟

تاکنون معیارهای مختلفی برای محاسبه مرکزیت در شبکه‌ها مورد بحث قرار گرفته است. درحالیکه این معیارها رویکرد مناسبی در درک نیروهای شبکه اجتماعی هستند، مع‌الوصف، هنوز هم یک سوال وجود دارد: از بین تمام موارد، کدام معیار مرکزیت بهتر است؟ در حقیقت، جواب این سوال به هدف ما از این تجزیه و تحلیلها بستگی دارد. با این حال، در بیشتر موارد، از همهٔ این چهار معیار، برای داشتن درک درستی از ساختار شبکه، استفاده می‌شود. در اینجا خلاصه کوتاهی برای استفاده از هر معیار آمده است:

۱) مرکزیت درجه‌ای، معیاری برای شهرت و محبوبیت است. این معیار، گره‌هایی را تعیین می‌کند که می‌توانند به سرعت اطلاعات را در یک منطقه محلی منتشر کنند.

۲) مرکزیت بینابینی بر این ایده استوار است که فردی در شبکه مهمتر است که بیشتر واسطه باشد.

۳) مرکزیت نزدیکی، معیاری برای اندازه‌گیری میزان دسترسی است؛ این معیار نشان می‌دهد چگونه اطلاعات از یک گره به سرعت به تمام گره‌های دیگر منتشر می‌شود.

۴) مرکزیت بردار ویژه، نوعی سنجش اهمیت وابستگی است. یعنی تعیین اینکه چه کسی به افراد پر نفوذ در گراف نزدیکتر است.

با توجه به سه معیار مرکزیت (درجه‌ای، بینابینی، و نزدیکی)، می‌توان گفت مرکزیت درجه‌ای، نسبتاً سرراستتر و سریعتر است و تنها نیاز به شمارش تعداد یالهای متصل به هر گره دارد. از سوی دیگر، مرکزیتهای بینابینی و نزدیکی، نیاز به بکارگیری درخت پوشای کمینه دارند که به نوبه خود، الگوریتم کراسکال[۱] را اجرا می‌کند. این الگوریتم دارای پیچیدگی زمان اجرا با مرتبه $O(ElogE)$ است که در آن

[۱] Kruskal's Algorithm

E تعداد یالهاست. برای معیار مرکزیت بردار ویژه، از آنجا که الگوریتم باید تکرار شود تا از طریق همسایگان هر گره، درجه موزون را محاسبه کند، الگوریتم دارای مرتبه عملیاتی *O(nodes*average_degree)* است. این امر باعث می‌شود که بکارگیری الگوریتم با توجه به پیچیدگی محاسباتی، مخصوصاً برای شبکه‌های بسیار بزرگ، کاری زمانبر و هزینه‌بر باشد.

فصل ششم

تحلیل در سطح گروه

در این فصل، قصد داریم برخی مباحث و تکنیکهای شناسایی "گروههای منسجم" در شبکه‌ها، مانند محافل، ضریب خوشه‌گی، تحلیل سه‌وندی، حفره‌های ساختاری، واسطه‌گری (یا کارگزاری)، تراگذری، خوشه‌بندی سلسله‌مراتبی و مدلهای بلوک را ارائه دهیم. همه این مفاهیم بر اساس نحوه اتصال گره‌ها در شبکه هستند؛ یعنی وضعیتِ تفرق یا تجمع گروهی از گره‌ها در شبکه (و کثرت یا قلت پیوند میان آنان) موجب پدیدار شدن این مفاهیم شده است. در میان همه موضوعات فوق، دو مبحث: تحلیل انسجام و تعیین واسطه‌گری، همچنان دو عنوان مهم پژوهشی در حوزه تحلیل شبکه اجتماعی است.

٦-١ زیرگروههای منسجم

تحلیلگران شبکه‌های اجتماعی از سه شاخص اصلی یعنی مرکزیت درجه‌ای، مرکزیت نزدیکی و مرکزیت بینابینی برای شناسایی افراد مهم در یک شبکه استفاده می‌کنند. "مرکزیت درجه‌ای" به گره‌هایی با بالاترین درجه بعنوان افراد مرکزی در شبکه نگاه می‌کند. "مرکزیت نزدیکی"، بازیگران اصلی را آن راسهایی می‌داند که با کوتاهترین متوسط طول مسیر، عملاً ارتباط دهنده یک گره به سایر گره‌های موجود در شبکه هستند. "مرکزیت بینابینی"، گره‌ای با بیشترین تعداد کوتاهترین مسیر بین جفت گره‌ها (که برای ارتباط با هم باید از آن گره عبور کنند) را بعنوان مهمترین کنشگر شبکه در نظر می‌گیرد.

با این حال، این معیارها به ما نمی‌گویند که چه نیروهایی سبب می‌شوند که تعدادی گره دور هم جمع شده (به هم متصل شوند) و پیوندهای خوبی با یک گره مرکزی برقرار کنند و یا اینکه چه نیروهایی شبکه‌ها را از هم جدا و یا تجزیه می‌کنند. همچنین آنها نمی‌گویند که یک گره خاص آیا تبدیل به گرهی با درجه بالا شده و خوشه‌سازی می‌کند (یعنی به دیگر گره‌ها قویاً متصل می‌شود) یا بجای این تمرکز، در جایی خالی از شبکه، گوشه خلوت و کنج عزلت می‌گزینند.

برخلاف فصل قبل، که تحلیل "گره‌های فردی" و الگوهای اتصال آنها را بررسی کردیم (یعنی تحلیل فردانه)، در این فصل، ما تمام "زیرگرافها و خوشه‌ها" را بررسی خواهیم کرد (یعنی تحلیل بخشانه). انجام این کار به ما کمک می‌کند تا ببینیم آیا چنین گروههای منسجمی با الگوهای دیگر شبکه، به لحاظ هنجارها[۱]، رفتار، تراکم (چگالی) و یا ویژگیهای اجتماعی[۲]، تفاوت دارند یا خیر.

زیرگروههای منسجم، مناطقی هستند که چگالی گره‌ها در آنجا زیاد است. آنها عمدتاً در شبکه‌های غیرجهتدار (در مقایسه با شبکه‌های جهتدار) یافت می‌شوند. اهمیت شناسایی زیرگروههای منسجم شبکه‌ها، در این است که زیرگروههای منسجم، خوشه‌های اجتماعی بالقوه در شبکه هستند.

روشهای متعددی برای شناسایی "زیرگروههای منسجم" در یک شبکه توسعه داده شده‌اند، روشهایی مانند: k–هسته‌ای (یک زیرشبکه که در آن، هر راس به حداقل k راس دیگر در زیرشبکه متصل است)، محافل (یعنی مولفه‌هایی با بیشینه اتصال، که معمولاً دارای سه راس هستند) و m–قطاع[۳] (یعنی یک زیرشبکه بیشین که دارای خطوطی با کثرت یا چندگانگی برابر یا بیشتر از m بوده و رئوس زیرشبکه با این خطوط تلاقی دارند).

---

[۱] Norms

[۲] Social Characteristics

[۳] M-Slice

٦-٢ محافل

در این بخش، قصد داریم دوستیهای متقابل را براساس داده‌های فیسبوک تحلیل کنیم. برخلاف داده‌های توییتر و برخی منابع دیگر که طبیعتاً بیشتر باز و در دسترس هستند و می‌توان داده‌ها را در یک دوره خاص ردیابی[۱] کرد، فیسبوک محدودیتهای دسترسی به داده‌های شخصی را اعمال می‌کند، به‌خصوص اگر بخواهیم یک شبکه اجتماعی را تحلیل کنیم. این امر موجب می‌شود فیسبوک جامعه بسیار بسته‌تری[۲] باشد.

کتابخانه *NetworkX* یک پیاده‌سازی خوب برای پیداکردن دوستیهای متقابل در شبکه‌های اجتماعی را از طریق تابع *nx.find_cliques ()* فراهم می‌کند. قصد داریم از داده‌های فیسبوک برای ساخت گراف دوستی متقابل استفاده کرده و سپس از تابع *nx.find_cliques ()* برای تجزیه و تحلیل محافل موجود در گراف بهره بگیریم.

پیدا کردن محافل در یک گراف، به لحاظ محاسباتی، جزء مسائل *NP*–کامل است؛ یعنی ترکیبات این مسئله با رشد داده‌ها، به‌صورت نمایی رشد می‌کنند. یافتن محافل برای گرافهای بسیار بزرگ چه بسا مستلزم صرف زمان زیادی باشد یا ممکن است نتایج تقریبی بدست دهد.

داده‌هایی که ما قصد داریم برای این تمرین از آنها استفاده کنیم، توسط دانشگاه استنفورد به آدرس مندرج در پاورقی[۳] تهیه شده است. با این حال، به دلیل اینکه مجموعه داده‌ها، نسبتاً بزرگ است و پیدا کردن محافل در یک گراف مستلزم صرف زمان با تابع نمایی است، ممکن است برای اجرای برنامه با همه داده‌ها، مشکل کمبود حافظه داشته باشیم. لذا، می‌توانیم بخش کوچکی از مجموعه داده اصلی را استخراج کرده و از آن برای تجزیه و تحلیلهای خود استفاده کنیم.

اجازه دهید برخی از اطلاعات مربوط به این گراف را بررسی کنیم:

```
In: print(nx.info(fb))
Out:
Name:
Type: Graph
```

---

۱ Crawl

۲ Closed Community

۳ Stanford University (https://snap.stanford.edu/data/egonets-Facebook.html)

```
Number of nodes: 1820
Number of edges: 7846
Average degree: 8.6220
```

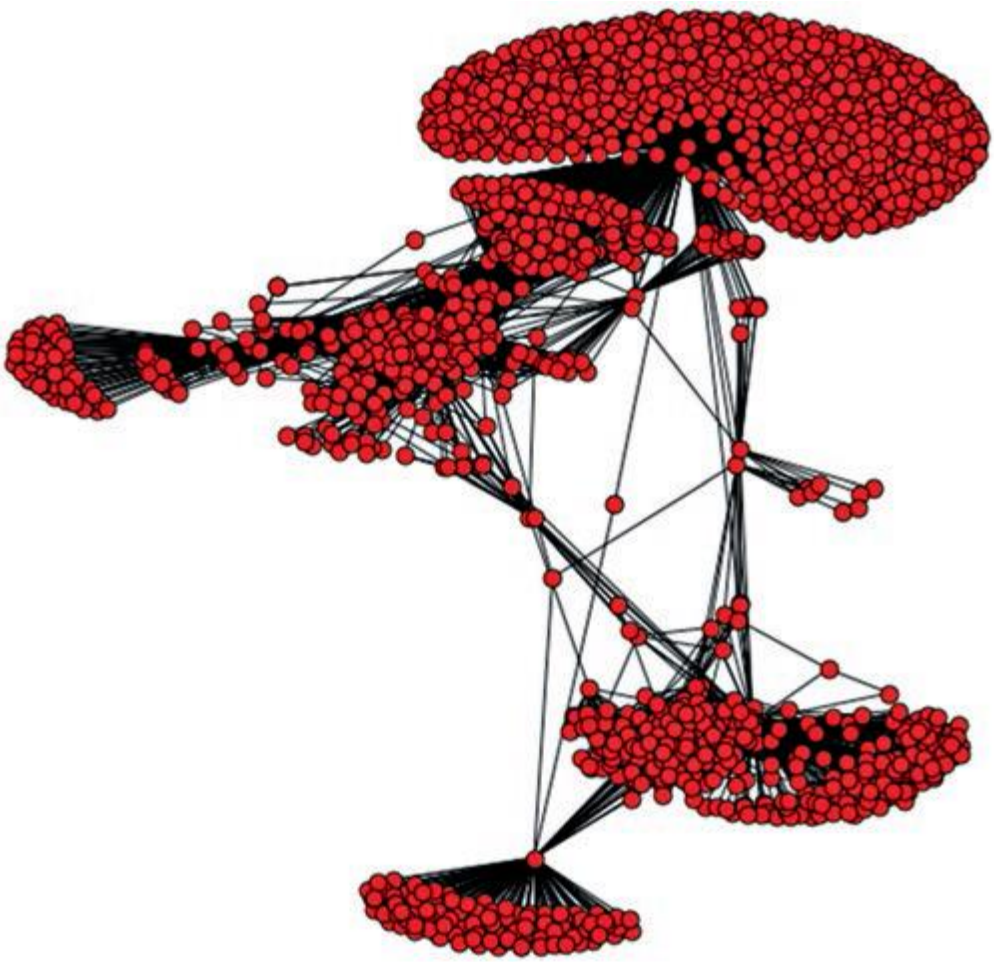
حالا ببینیم گراف چگونه به نظر می‌رسد:

```
In:  fb = nx.read_edgelist("…/facebook_combined_minimized.txt",
     create_using=nx.Graph(), nodetype=int)
In:  cliques = [x for x in nx.find_cliques(fb)]
```

در فایلهای "فهرست یال"[۱]، ستون اول، نام "گره از" است در حالیکه ستون دوم، نام "گره به" است. ممکن است برخی اطلاعات اضافی پس از ستون دوم برای نشان دادن قدرت ارتباط باشد.
اکنون، بیایید شکل را بسازیم (شکل۱-۶):

```
In:  pos = nx.spring_layout(fb)
     plt.figure(figsize=(12,12))
     plt.axis('off')
     nx.draw_networkx(fb, pos=pos, with_labels=False, node_size=80)
Out:
```

[۱] Edge-List Files



شکل ۱-٦ نمایش گرافیکی مجموعه داده‌های فیسبوک

مجموعه داده‌ها به وضوح نشان می‌دهد که چندین (حدود شش) منطقه متمایز بزرگ وجود دارد. این مناطق در یک طرف گراف قرار ندارند، بلکه در دو طرف هستند، درحالی که همزمان با تعدادی یال بلند متصل می‌شوند.

برای مشاهده محافل موجود در گراف داریم:

```
In:  cliques
Out:
```

```
[[1, 0, 133, 280, 322, 315, 236],
[1, 0, 73, 48, 88, 126],
[1, 0, 73, 48, 299],
[1, 0, 236, 322, 88],
[1, 0, 53, 88, 48, 322],
[1, 0, 53, 315, 322],
[1, 0, 53, 299, 48],
[1, 0, 53, 299, 194, 346, 92],
[1, 0, 53, 54, 48],
[1, 0, 119, 48, 54],
[1, 0, 119, 280],
[1, 0, 126, 48, 88, 322],
[2, 0, 226, 149, 343, 333, 20],
[2, 0, 226, 149, 343, 326, 312, 115, 20],
…]
```

الگوریتم محفل، یک خروجی تولید می‌کند که براحتی با چشم غیرمسلح قابل تفسیر است. حال بیایید بررسی کنیم که چند محفل در گراف وجود دارد:

```
In:  n_cliques = len(cliques)
     print(n_cliques)
Out: 10193
```

و در ادامه بیایید نگاهی به اندازه محافل (با تکرار) در گراف داشته باشیم:

```
In:  sizes_of_cliques = [len(x) for x in cliques]
     print(sizes_of_cliques)
Out:
     [7, 6, 5, 5, 6, 5, 5, 7, 5, 5, 4, 6, 7, 9, 9, 8, 11, 11, 11, 11, 7,
     8, 7, 4, 4, 5, 10, 4, 4, 4, 5, 5, 5, 5, 3, 4, 6, 6, 5, 5, 7, 7, 5,
     6, 7, 5, 4, 6, 5, 5, 8, 6, 5, 8, 5, 6, 8, 5, 6, 5, 5, 5, 4, 6, 7,
     7, 7, 6, 5, 5, 6, 5, 6, 11, 11, 11, 11, 11, 11, 10, 10, 11, 11, 11,
     11, 11, 11, 4, 6, 6, 6, 7, 5, 5, 8, 10, 10, 8, 10, 10, 10, 10, 9,
     11, 12, 11, 12, 9, 9, 12, 13, 12, 13, 11, 11, 12, 9, 9, 11, 11, 12,
     12, 9, 9, 10, 10, 11,…]
```

```
In:  maximum_clique_size = max(sizes_of_cliques)
     print(maximum_clique_size)
Out: 16
```

در اینجا اجازه دهید محفلهای با حداکثر اندازه ۱۶ را پیدا کنیم:

```
In:  maximu_cliques = [x for x in cliques if len(c) ==
      maximum_clique_size]
     print(maximu_cliques)
Out:
     [[9, 0, 271, 26, 67, 56, 122, 277, 322, 21, 170, 200, 186, 25, 188,
     323], [9, 0, 271, 26, 67, 56, 122, 277, 322, 21, 170, 200, 186, 25,
     188, 252], [9, 0, 271, 26, 67, 56, 122, 277, 322, 21, 170, 200,
     186, 142, 188, 323], [9, 0, 271, 26, 67, 56, 122, 277, 322, 21,
     170, 200, 186, 142, 188, 252], [9, 0, 271, 26, 67, 56, 122, 277,
     322, 285, 170, 200, 186, 25, 188, 323], [9, 0, 271, 26, 67, 56,
     122, 277, 322, 285, 170, 200, 186, 25, 188, 252],
     …]
```

و همچنین تعداد آنها:

```
In:  n_maximum_cluqies = len(maximu_cliques)
     print(n_maximum_cluqies)
Out: 26
```

این بدان معناست که ۲۶ "محفل بیشینه" متفاوت در گراف داریم که هر کدام از آنها مشتمل بر ۱۶ گره (یا نفر) هستند. به بیان دیگر، در تمام آن ۲۶ محفل متنوع (که متشکل از ۱۶ نفرند)، همه افراد یکدیگر را می‌شناسند.

محاسبه متوسط اندازه محافل:

```
In:  average_clique_size = sum(sizes_of_cliques)/n_cliques
     print(average_clique_size)
Out: 8
In:  maximum_clique_sets = [set(x) for x in maximu_cliques]
In:  print(maximum_clique_sets)
Out:
     [set([0, 322, 67, 200, 9, 170, 271, 323, 277, 186, 56, 25, 26, 188,
```

```
122, 21]), set([0, 322, 67, 200, 9, 170, 271, 252, 277, 186, 56,
25, 26, 188, 122, 21]), set([0, 322, 67, 200, 9, 170, 142, 271,
323, 277, 186, 56, 26, 188, 122, 21]),…]
```

اجازه دهید از یک تابع کاهشی[۱] برای پیداکردن آن دسته از گره‌ها (یا دوستانی) استفاده کنیم که در میان همه "محافل بیشینه" منتجه، مشترک هستند. به یاد داشته باشید که ما ۲۶ محفل با اندازه بیشینه داریم:

```
In:  friends_in_all_maximum_cliques = list(reduce(lambda x, y:
     x.intersection(y), maximum_clique_sets))
     print(friends_in_all_maximum_cliques)
Out: [0, 67, 200, 122, 271, 277, 56, 186, 26]
```

شاید جالبترین نکته این است که ۹ نفر وجود دارند که در همه ۲۶ "محفل بیشینه" دیده می‌شوند.

٦-۳ ضریب خوشه‌گی

ضریب خوشه‌گی[۲] نسبتی از همسایگان یک گره مشخص است که با یکدیگر نیز همسایه می‌باشند. همسایگان یک گره، مجموعه‌ای از گره‌ها هستند که با یک یال به آن گره متصلند و البته خود گره را در بر نمی‌گیرد. این معیار را می‌توان هم بصورت محلی و هم بصورت کلی (سراسری) محاسبه کرد. این معیار در سطح محلی (و برای شبکه‌های فردانه) بر "همسایگی یک گره" تاکید می‌کند، در حالیکه در سطح کلی (و برای شبکه جمعانه) نمایشگر "درجه خوشه‌گی یک گراف" است. با این‌حال، اگر این معیار در سطح کلی و برای شبکه‌هایی با چگالی بسیار متغیر و هسته‌های چندگانه[۳] محاسبه شود، نتایجی را نشان می‌دهد که تفسیر آن بسیار دشوار است.

ضریب خوشه‌گی برای شبکه‌های اجتماعی، معیار بسیار مهمی است زیرا سبب مشخص شدن نسبتی از دوستان شخص است که با یکدیگر هم دوست هستند. این نسبت فی‌المثل میزان اعتماد مردم به

---

[۱] Reduce Function

[۲] Clustering Coefficient

[۳] Multiple Cores

یکدیگر را نشان می‌دهد. با این معیار می‌توان فهمید که تا چه حد در شبکه‌های اعتماد، پیامها به‌راحتی می‌توانند پخش شده و پایدار بمانند.

مثال ۱

بیایید نگاهی به مثال زیر بیندازیم که مجدداً براساس گراف کایت کراکارت است:

```
In:  nx.clustering(g)
Out:
     {0: 0.6666666666666666,
      1: 0.6666666666666666,
      2: 1.0,
      3: 0.5333333333333333,
      4: 1.0,
      5: 0.5,
      6: 0.5,
      7: 0.3333333333333333,
      8: 0.0,
      9: 0.0}
```

تابع *nx.clustering* از کتابخانه *NetworkX*، ضریب خوشه‌گی برای هر گره در گراف را محاسبه می‌کند. هر قدر میزان ضریب خوشه‌گی یک گره بیشتر باشد، نشاندهنده درجه محفلی[۱] بزرگتر است که به معنای وجود بخشهایی با اتصال بالا در شبکه است.

بیایید تابع *nx.average_clustering()* را اجرا کنیم که "میانگین ضرایب خوشه‌گی" را برای گراف محاسبه می‌کند:

```
In:  nx.average_clustering(g)
Out: 0.5199999999999999
```

مثال ۲

محاسبه مقادیر ضریب خوشه‌گی هر گره برای شبکه قهرمان:

```
In:  nx.clustering(graph)
```

[۱] Cliquishness

```
Out:
     {u'QUESADA, JOE': 1.0,
     u'ZEITGEIST II/AXEL CL': 1.0,
     u'LOGA II': 0.7040169133192389,
     u'PIECEMEAL/GILBERT BE': 0.8768873403019745,
     u'BENWAY, DR.': 0.954248366013072,
     u'FRIEDLANDER, SHARON': 0.5136507936507937,
     u'DREAMING CELESTIAL/T': 0.7282282282282282,
     u'MODRED THE MYSTIC': 0.37714776632302405,
     u'THOMAS, DR. STAN': 0.9636363636363636,
     u'FIREWALL/THEARY/MIN ': 0.6688251618871416,
     u'DR. FAUSTUS': 0.6133333333333333,…]
```

اکنون، متوسط ضریب خوشه‌گی شبکه محاسبه می‌شود:

```
In:  nx.average_clustering(graph)
Out: 0.7746541217110626
```

بیایید ببینیم چگونه می‌توانیم ضریب خوشه‌گی را برای یک گره خاص محاسبه کنیم. کتابخانه *NetworkX* هیچ پیاده‌سازی مستقیمی برای این کار ندارد. ما ابتدا باید شبکه فردانه (فردمحور) را استخراج کنیم و سپس از تابع خوشه‌بندی متوسط، به‌صورت زیر استفاده می‌کنیم:

```
In:  ego_net = nx.ego_graph(graph, "QUESADA, JOE")
In:  len(ego_net)
Out: 40
In:  nx.average_clustering(ego_net)
Out: 1.0
```

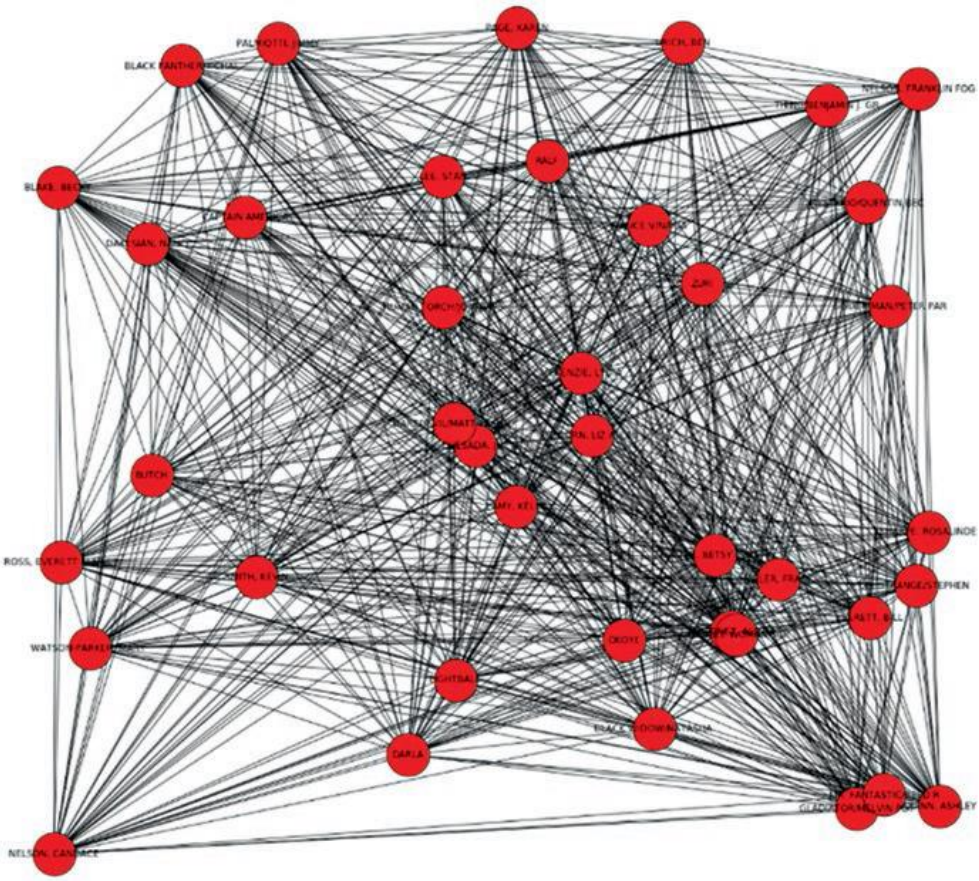
همانطور که مشاهده می‌کنید، شخص *Joe Quesada* دارای شبکه‌ای با ۴۰ گره است (یعنی ۴۰ همسایه دارد). اما چرا ضریب خوشه‌گی وی، یک است؟ بیایید شبکه محلی او را بررسی کنیم:

```
In:  nx.clustering(ego_net)
Out:
     {'QUESADA, JOE': 1.0, u'ZURI': 1.0, u'GLADIATOR/MELVIN POT': 1.0,
     u'SMITH, KEVIN': 1.0, u'OKOYE': 1.0, u'RALF': 1.0, u'PALMIOTTI,
```

```
JIMMY': 1.0, u'BLAKE, BECKY': 1.0, u'MYSTERIO/QUENTIN BEC': 1.0,
u'WATSON-PARKER, MARY ': 1.0, u'SPIDER-MAN/PETER PAR': 1.0,
u'MCKENZIE, LYDIA': 1.0, u'LAMY, KELLY': 1.0, u'URICH, BEN': 1.0,
u"BLACK PANTHER/T'CHAL": 1.0, u'MR. FANTASTIC/REED R': 1.0,
u'THING/BENJAMIN J. GR': 1.0, u'QUINN, ASHLEY': 1.0, u'CAPTAIN
AMERICA': 1.0, u'MALICE V/NAKIA': 1.0, u'NELSON, CANDACE': 1.0,
u'MARTINEZ, ALITHA': 1.0, u'BUTCH': 1.0, u'EIGHTBALL': 1.0,
u'DAREDEVIL/MATT MURDO': 1.0, u'ROSS, EVERETT KENNET': 1.0,
u'POTTER, BETSY BEATTY': 1.0, u'BLACK WIDOW/NATASHA ': 1.0,
u'EVERETT, BILL': 1.0, u'DAKESIAN, NANCI': 1.0, u'SHARPE,
ROSALINDE': 1.0, u'NELSON, FRANKLIN FOG': 1.0, u'MILLER, FRANK':
1.0, u'HUMAN TORCH/JOHNNY S': 1.0, u'OSBORN, LIZ ALLAN': 1.0, u'DR.
STRANGE/STEPHEN ': 1.0, u'DARLA': 1.0, u'INVISIBLE WOMAN/SUE ':
1.0, u'PAGE, KAREN': 1.0, u'LEE, STAN': 1.0}
```

این تابع، قابلیت محاسبه "ضریب خوشه‌گی" مربوط به هر گره (قهرمان) در "شبکه فردانه" را دارد. *Joe Quesada* توسط مجموعه‌ای از قهرمانان که هر کدام cc=۱ دارند، احاطه شده است (شکل۲-۶).

```
In:  plt.figure(figsize=(30,30))
     pos = nx.random_layout(ego_net)
     nx.draw_networkx(ego_net, pos, node_size=3000, with_labels=True)
```



شکل ۲-٦ شبکه قهرمان-قهرمان نشان‌دهنده ارتباطات شبکه فردانه (خودمحور) است.

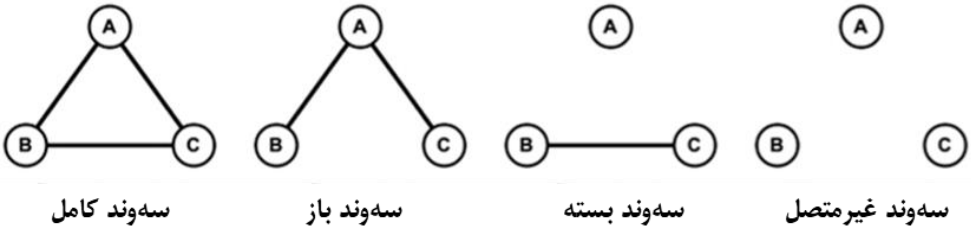
٦-٤ تجزیه و تحلیل سه‌وَندی

منظور از سه‌وَند (یا سه‌گاه)، زیرگرافی با سه گره است، یعنی: یک گره کانونی (فردانه)، یک گره همنشین (دگرانه) و یک گره سوم (ثالثانه یا وان‌دگرانه[۱]). البته این ساختار فقط شامل سه گره نیست بلکه همچنین

[۱] Third Party

شامل هر پیوندی است که بین این گره‌ها در آن زیرگراف وجود دارد. یک سه‌وَند، کوچکترین ساختار گراف است که بیش از دو گره دارد (شکل ۳-۶).

در شکل ۳-۶، چهار نوع ممکن از اشکال غیرجهتدار سه‌وَند وجود دارد: بسته، باز، متصل (همبند) و غیرمتصل (ناهمبند). گرافی که در سمت چپ شکل ۳-۶ قرار دارد، یک سه‌وَند کامل[۱] است، یعنی یک زیرشبکه کامل[۲] که متشکل از سه گره است.

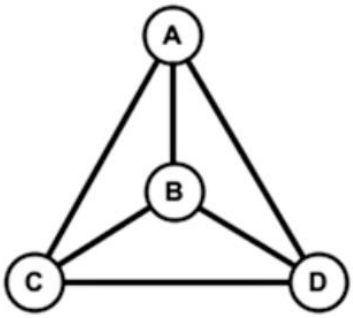


شکل ۳-٦ سه‌وَند در چهار حالت مختلف

گره‌های A، B، و C از طریق ارتباطات همتراز و یا هم‌قدرت با یکدیگر پیوند دارند. این سه‌وَندِ کامل، یک محفل با اندازه سه است. چرا چنین است؟ این بدان دلیل است که نمی‌توانیم یک گره دیگر از این گراف را به این زیرگراف اضافه کنیم به طوریکه همچنان کامل باقی بماند. با این‌حال، در گرافِ کاملِ غیرجهتدار شکل ۴-۶، گره‌های A، B و C یک محفل ایجاد نمی‌کنند، زیرا می‌توانیم گره D را اضافه کنیم و زیرشبکه همچنان کامل باقی بماند. گره‌های A، B، C و D یک محفل با اندازه چهار تشکیل می‌دهند که از چهار مثلث کامل تشکیل شده است (شکل ۴-۶).

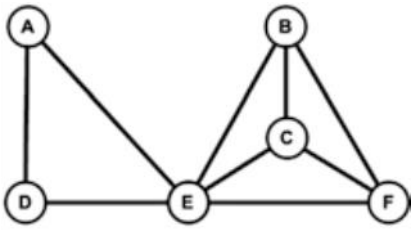
[۱] Complete Triad

[۲] Complete Subnetwork



سه‌وند کامل

شکل ٤-٦ سه‌وند کامل



سه‌وند همپوشان

شکل ٥-٦ سه‌وَند همپوشان یا متداخل[۱]

شکل ۵-۶ نشاندهنده یک ویژگی حیاتی سه‌وَندِ کامل است، یعنی سه‌وَندهای کامل می‌توانند با یکدیگر همپوشانی داشته باشند. سه‌وَند کامل A، D و E با سه‌وند کامل B، C و E، همپوشانی دارد چرا که هر دوی آنها در گره E مشترک هستند. در بستر شبکه‌های اجتماعی، محافل همپوشان[۲]، عملاً زیرگرافهای منسجمی هستند که نمایانگر همپوشانی حلقه‌های اجتماعی[۳] هستند. مثلاً در یک سه‌وَند کاملاً همبند، سه نفر اطلاعات و هنجارها را به اشتراک می‌گذارند، اعتماد لازم را با بازخورد ایجاد می‌کنند و مشکلات احتمالی با شخص ثالث (واندگرانه) را حل و فصل یا رتق و فتق می‌کنند. آنها مانند یک "تیم" رفتار

[۱] Overlapped Triad

[۲] Overlapping Cliques

[۳] Overlapping Social Circles

می‌کنند و نه همچون مجموعه‌ای از "افراد منفرد"[۱]. این وضعیت سه‌وَندها، با ساختار دووَندها (یا دوگاه‌ها) متفاوت است که در آنها دو نفر بطور انحصاری، اطلاعات و نظرات را مبادله کرده و رابطه خود را حفظ می‌کنند، در حالیکه، خود را به هنجارهای تیمی مقید نمی‌کنند (شکل ۵–۶).

برای یک گراف جهتدار ساده، ممکن است ۱۶ نوع سه‌وَند مختلف رخ دهد. این انواع در شکل ۶–۶ نشان داده شده‌اند. از آنجا که ما یالهای یک طرفه[۲] و نیز دو طرفه[۳] را در نظر می‌گیریم، لذا تنوع سه‌وندها به ۱۶ نوع متفاوت، بالغ می‌شود.

هر نوع سه‌وند با یک "کد سه‌بخشی" و گاهی چهاربخشی مشخص می‌شود که به آن شماره *M-A-N* می‌گویند. حرف *M*، که از حرف اول واژه *Mutual* اخذ شده، اولین بخش این کد و نشاندهنده تعداد دووندهای مثبت متقابل است. حرف *A*، که از حرف اول واژه *Asymmetric* اقتباس شده، بخش دوم کد و نشاندهنده تعداد دووندهای نامتقارن است. درنهایت حرف *N*، که از واژه *Null* گرفته شده، بخش سوم کد بوده، و نمایشگر تعداد دووَندهای خالی است. برای تشخیص سه‌وَندهایی که دارای یالهای جهتدار و شمارهٔ یکسان *M-A-N* می‌باشند، یک حرف الفبا اضافه می‌شود. *D*: پایین، *U*: بالا، *C*: چرخه[۴] و *T*: تراگذر (در سه‌وَند تراگذر، هر مسیری به طول دو، توسط کمانی که از گره شروع به سمت گره پایانی مسیر رفته، بسته شده است) (شکل ۶–۶).

تحقیقات نشان داده است که شناخت انواع سه‌وَندهایی که در شبکه جهتدار وجود دارند، می‌تواند به استنباط ما در مورد ساختار کلی شبکه کمک کند. به عبارت دیگر، با تجزیه و تحلیل سه‌وَندهای شبکه، می‌توانیم ساختار کلی شبکه را درک کنیم.

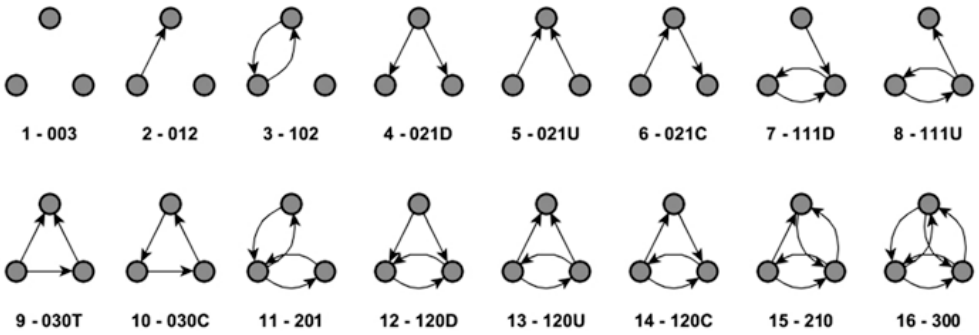
---

[۱] Set of Individuals

[۲] One Way

[۳] Bidirectional

[۴] Cyclic



شکل ٦-٦ انواع اتصالات در یک سه‌وَند جهتدار

٦-٥ حفره‌های ساختاری

یک حفره ساختاری، بر اساس تعریف، نوعی انفصال در شبکه فردانهٔ یک گره (بدلیل فقدان یک یال) است به گونه‌ای که دو همسایهٔ آن یعنی گره‌های دگرانه و وان‌دگرانه (شخص ثالث) نمی‌توانند به طور مستقیم بهم متصل شوند. شخصی که به این دو نفرِ منفصلِ شبکه (یعنی دگرانه و وان‌دگرانه) متصل است، در موقعیت واسطه‌گری[۱] بین دو طرف است و می‌تواند از این وساطت[۲] سود ببرد. این شخص ممکن است از موضع خود برای ایجاد اختلاف بین دو طرف دیگر بمنظور کنترل آنها یا حل و فصل اختلافات آنها استفاده کند. در سازمانها، این شخص (احتمالاً فرد خبره‌ای است) که می‌تواند بین کارهای دو یا چند بخش مختلف ارتباط برقرار کند و به این دلیل دارای اهمیت و اعتبار بیشتری است.

تئوری حفره‌های ساختاری[۳] توسط رونالد برت[۴] در کتابش به نام "حفره های ساختاری: ساختار اجتماعی رقابت[۵] " در سال ۱۹۹۲ بعنوان تلاشی برای مطالعه اهمیت موقعیت کاربران در شبکه‌های اجتماعی معرفی شد. این ایده و موارد دیگری که توسط رونالد برت توسعه داده شده‌اند، راه جدیدی را برای تفکر

[۱] Mediate
[۲] Mediation
[۳] Theory of Structural Holes
[۴] Ronald Burt
[۵] Structural Holes: The Social Structure of Competition

بیشتر درباره مباحث مربوط به رفتارهای اجتماعی باز کرد؛ مباحثی همچون اینکه چرا و چگونه کاربران به یکدیگر متصل شده‌اند و این امر چگونه می‌تواند بر محدودیتها، فرصتها و در نهایت رفتار آنها تأثیر بگذارد.

٦-٦ واسطه‌گری

موقعیتهای خاصی (همچون میانداری و یا واسطه‌گری) در یک شبکه وجود دارد که می‌تواند بشدت در تبادل اطلاعات، جابجایی کالاها، انتشار بیماری، شیوع شایعه یا ارائه خدمات دخیل باشند. این درحالی است که موقعیتهای معمولی در شبکه این ویژگیها را ندارند. با داشتن چنین موقعیتهای ویژه‌ای، واسطه‌گران[۱] (یا کارگزاران) سعی می‌کنند با تمرکز بر روابط اجتماعی موجود در شبکه‌ها، نقش کانالهای مبادله را بازی کنند. معیار واسطه‌گری در یک سه‌وند، عملاً پتانسیل فردانه را برای ایجاد و بهره برداری از رقابت بین کنشگران می‌سنجد. این معیار همچنین، پتانسیل آن شخص (یعنی فردانه) را برای ایفای نقشهای موذیانه یا خرابکارنه[۲] از طریق ایجاد مخاصمه و یا بهره‌برداری از تخاصمات موجود بین عاملان دیگر برای کنترل آنها، اندازه‌گیری می‌کند.

اینکه تا چه حد یک کنشگر می‌تواند در یک شبکه، نقش واسطه‌گری ایفا کند، بستگی زیادی به حضور حفره‌های ساختاری دارد: وجود حفره‌های ساختاری بیشتر، به معنای استراتژیهای واسطه‌گری بیشتر در عمل است.

مفهوم مهم دیگر که مربوط به واسطه‌گری (یا کارگزاری) است، اصطلاحاً "محدودیت تجمیعی[۳]" نامیده می‌شود که جمع "محدودیتهای دووَندی" در تمام پیوندهای مربوط به یک گره خاص است. هرچه محدودیت تجمیعی بیشتر باشد، یک واسطه‌گر نقش نسبتاً کمتری می‌تواند ایفا کند. (توجه داشته باشید که "وجود دسترسی" عملاً برعکس "وجود حفره ساختاری" است. در اینجا از کلمه "محدودیت" عملاً مفهوم "محدودیت برای واسطه‌گری" مستفاد می‌شود. یعنی وقتی دسترسی جمع همسایگان به همدیگر، یعنی بصورت دو به دو یا دووَندی فراهم است، آنگاه گره فردانه برای ایفای نقش واسطه‌گری بین گره

---

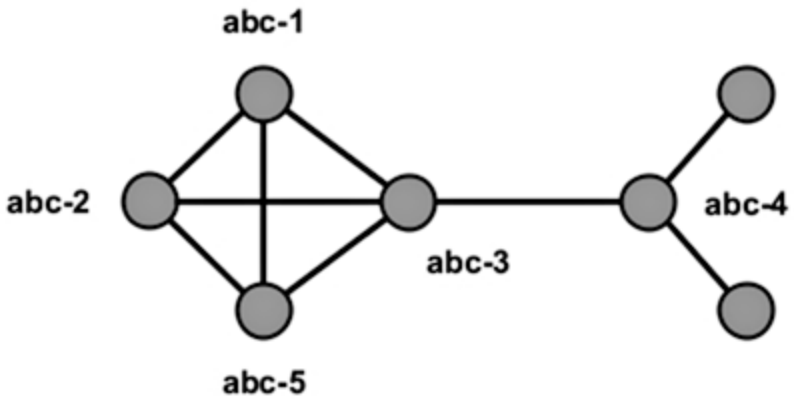
[۱] Brokerage

[۲] Subversive Roles

[۳] Aggregate Constraint

دگرانه و گره واندگرانه دچار محدودیت می‌شود). اثر محدودیت تجمیعی در بسیاری از وضعیتهای زندگی واقعی نیز مشهود است. بعنوان مثال، کارکنانی که دارای محدودیت کمتری هستند، شغلهای موفقیت‌آمیز بیشتری نسبت به دیگران دارند و یا بخشهای کسب و کارِ با قیود یا محدودیتهای کمتر، سودآورتر هستند.

در حالیکه حفره‌های ساختاری، فرصتهای عالی برای واسطه‌گری در "سه‌وند ناکامل" ایجاد می‌کنند، در "سه‌وَند کامل" موضوع برعکس بوده و واسطه‌گری دچار محدودیت جدی می‌شود. از آنجا که هیچ حفره ساختاری در سه‌وند کامل وجود ندارد، ایفای نقش واسطه‌گری توسط یک شخص خاص (فردانه) بسیار دشوار می‌شود و هر تلاشی از جانب او برای برهم‌زدن ارتباطات موجود در شبکه، باعث ایجاد حفره‌های ساختاری در اطراف خود او شده که برای خودش مضر بوده و برای هم‌نشینان او مفید خواهد بود. بیایید نگاهی به شکل ۷–۶ بیاندازیم.



شکل ۷–٦ شبکه فردانه (فردمحور) نشاندهنده فرصتهای واسطه‌گری

شکل ۷–۶ یک شبکه فردانه (یا فردمحور) را نشان می‌دهد. این گراف شامل گرهٔ فردانه (یعنی*abc-3*) و همسایگان گره فردانه و روابط میان آنها است. ما می‌توانیم گراف را از منظر سه‌وندهایش تجزیه و تحلیل کنیم. بعنوان مثال، فردانه *abc-3*، این فرصت را دارد تا بین *abc-4* و *abc-5* واسطه‌گری کند، زیرا تمام ارتباطات بین این دو گره باید از طریق گره فردانه صورت بگیرد (یعنی گره *abc-3* در کوتاهترین مسیر بین آن دو وجود دارد). این مطلب همچنین برای همه ارتباطات بین گره‌های *abc-4* و *abc-2* یا *abc-5* صادق است. با توجه به اینکه *abc-3* سه تا سه‌وَند دارد، این امکان را به او

قدرت می‌دهد تا برای *abc-4* واسطه‌گری کنند. البته، برای این فردانه در مورد واسطه‌گری بین *abc-1*، *abc-2* و *abc-3* محدودیت وجود دارد. زیرا این سه گره، یک سه‌وند کامل ایجاد کرده‌اند و بنابراین هیچ راه و فرصتی برای گرهٔ فردانه جهت نفوذ و واسطه‌گری بین آنها وجود ندارد.

با مقایسه دو حالت فوق، می‌توان گفت برای گرهٔ فردانه جهت واسطه‌گری برای گره *abc-4* محدودیت کمی وجود دارد، زیرا هم‌نشینانش به‌طور مستقیم با *abc-1*، *abc-2* یا *abc-5* مرتبط نیستند. لیکن، محدودیت بیشتری برای گره فردانه جهت واسطه‌گری میان *abc-1*، *abc-2* یا *abc-5* وجود دارد، زیرا این سه گره یک سه‌وندِ کامل را تشکیل می‌دهند.

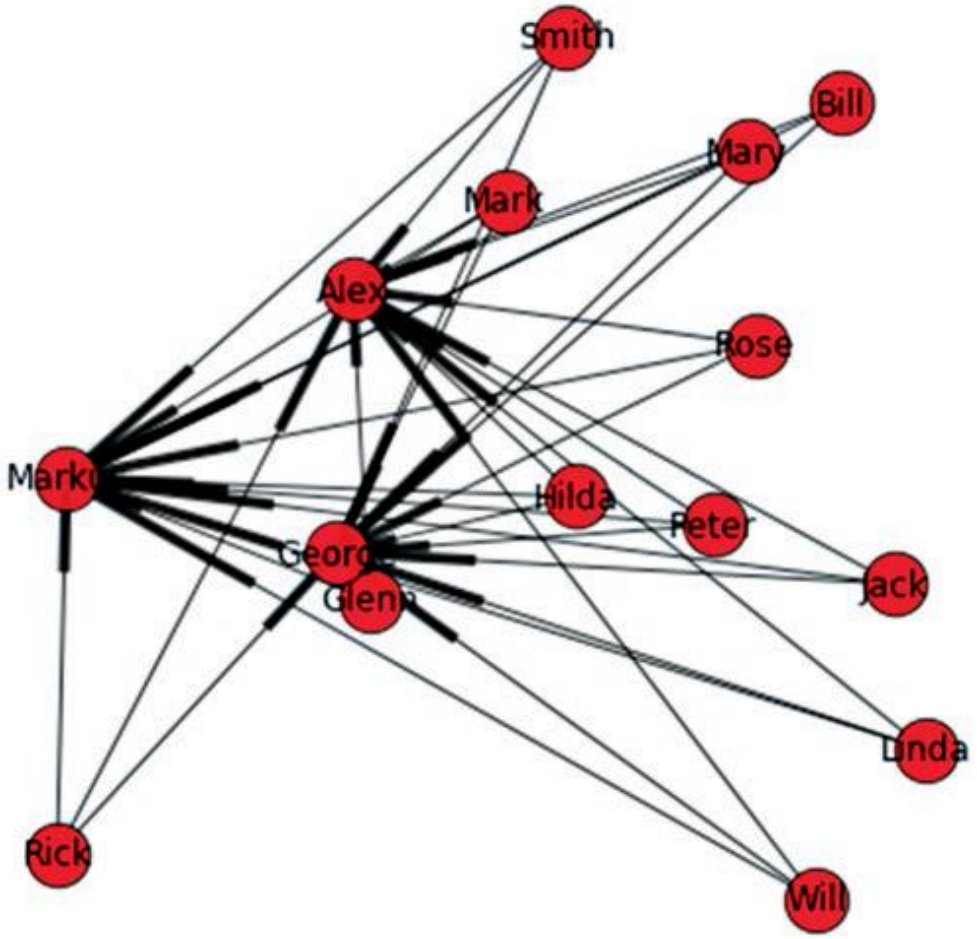
مثال

در کتابخانه *NetworkX* یک تابع بنام *(triadic_census)* وجود دارد که "تعداد سه‌وَندهای یک گراف جهتدار" را تعیین می‌کند. به عبارت دیگر، این تابع مشخص می‌کند چه تعداد از ۱۶ نوع ممکن سه‌وَند در یک گراف جهتدار موجود است. این تابع، یک پارامتر و یک گراف جهتدار *NetworkX* را دریافت کرده و در جواب، یک دیکشنری (که **کلید** آن: نام سه‌وَندها، و **مقدار** آن: تعداد رخداد هر سه‌وَند است) را باز می‌گرداند. این تابع دارای پیچیدگی *O(m)* است که در آن *m* تعداد یالها در گراف است.

در این مثال، قصد داریم با استفاده از توییتر ساختگی (یک گراف جهتدار) که قبلاً ایجاد کرده‌ایم، تابع سرشماری سه‌وَند[۱] را اعمال کنیم (شکل ۸-۶).

```
In:  print(nx.info(graph))
Out: Name:
     Type: DiGraph
     Number of nodes: 15
     Number of edges: 36
     Average in degree: 2.4000
     Average out degree: 2.4000
In:  plt.figure(figsize=(8,8))
     layout = nx.random_layout(graph)
     nx.draw_networkx(graph, layout, node_color='r', node_size=500,
     alpha = 0.8
```

[۱] Triadic Census Function



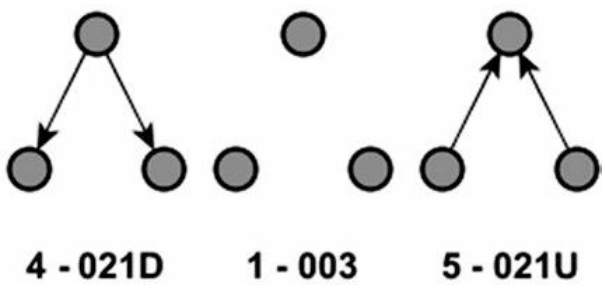
شکل ۸-٦ شبکه توییتر ساختگی با ۱۵ گره

اکنون تابع "سرشماری سه‌وَندی" را اجرا می‌کنیم:

```
In:  print(nx.triadic_census(graph))
Out: {'201': 0, '021C': 0, '021D': 36, '210': 0, '120U': 0, '030C': 0,
      '003': 221, '300': 0, '012': 0, '021U': 198, '120D': 0, '102': 0,
      '111U': 0, '030T': 0, '120C': 0, '111D': 0}
```

همانطور که می‌بینیم، این تابع یک دیکشنری شامل نام سه‌وَندها و تعداد رخداد هر یک از آنها را برمی‌گرداند. با مشاهده این نتایج و مقایسه آن با فهرست انواع سه‌وَندها، متوجه می‌شویم که سه نوع سه‌وَند در این گراف جهتدار وجود دارند:

سه‌وَند نوع *021D* (۳۶ نمونه)، **سه‌وَند نوع *003*** (۲۲۱ نمونه) و **سه‌وَند نوع *021U*** (۱۹۸ نمونه) (شکل ۹-۶).



شکل ۹-٦ انواع سه‌وند موجود در شبکه توییتر ساختگی

سه‌وَندهای ***021D*** و ***021U*** حاوی جفت گره‌های ناهمبند هستند که سبب ایجاد دو نوع حفرهٔ ساختاری شده‌اند. با این حال، به دلیل پیوندهای دووَند نامتقارن[۱] (یا یکسویه)، گره فردانه نمی‌تواند از موقعیت خود بعنوان یک واسطه بین دو گره دیگر، بهره‌مند شود.

نوع دیگر سه‌وَند موجود در این ساختار، یعنی **۰۰۳**، در شبکه‌های خوشه‌پذیر[۲] یافت می‌شود، زیرا حاوی گره‌هایی هستند که به سه خوشه متفاوت تعلق دارند.

داشتن دو نوع مختلف سه‌وند (یعنی حفره‌های ساختاری و سه‌وَندهای ناهمبند[۳]) به این معنی است که این شبکه، ساختار یگانه و سازگاری ندارد. به بیان دیگر، از سویی، مناطقی در شبکه وجود دارند که تعداد زیادی حفره ساختاری داشته (۲۳۴ = ۱۹۸ + ۳۶) و از دگر سو، مناطق دیگری در شبکه وجود دارند که دارای تعداد زیادی سه‌وَند ناهمبند (یعنی ۲۲۱ مورد) هستند. وضعیت دیگری که در اینجا شاهد آن

[۱] Asymmetric Dyad Ties

[۲] Clusterable Networks

[۳] Unconnected Triads

هستیم این است که سه‌وَندهای بسته (کد **۳۰۰**) که نشاندهنده مناطقی با بستار[۱] بالاست، در ساختار این شبکه وجود ندارند.

نسبت بین سه‌وَندهای ناهمبند (غیرمتصل) و حفره ساختاری حدود (۰,۹) است، به این معنی که مناطق متعلق به هرکدام از آنها، تقریباً برابر هستند. این مناطق را می‌توان از تصویر نمونه نیز مشاهده کرد، اما برای شبکه‌های بزرگتر، ممکن است خیلی آشکار نباشد.

نوع تجزیه و تحلیل در این مثال، یعنی "سرشماری سه‌وَندها"، نمونه‌ای از یک استراتژی تحقیق است که بر ساختارهای محلی شبکه متمرکز است، زیرا تنها یالهای درون سه‌وندها را در نظر می‌گیرد. انجام "تجزیه و تحلیل سه‌وَندی یا سه‌وَندانه"[۲] که بر تحلیل سه‌وندها و روابط میان آنها در گرافهای جهتدار متمرکز می‌شود، بر اساس مدلهای آماری و اجرای آزمون فرضیه است. این مدلهای آماری در واقع، بدنبال کشف انگیزه‌هایی هستند که باعث می‌شوند برخی گره‌ها، از میان همه گره‌های شبکه، گره‌های خاصی را انتخاب کرده و با آنها اتصال برقرار کنند. این تحلیل همچنین به شناسایی نقش گره در ساختار شبکه کمک می‌کند.

٦-۷ تراگذری

تراگذری یا انتقال‌پذیری، یک ویژگی مربوط به یالها است که به ما اجازه می‌دهد تا "بستار سه‌وندی" (سه‌گانه‌های بسته[۳]) را ایجاد کنیم و پیش‌بینیهایی درباره همسایگان گره مربوطه انجام دهیم. اگر گره A به گره B وصل باشد و گره B نیز به گره C وصل باشد، آنگاه احتمال دارد که گره A نیز به گره C متصل باشد. مقدار تراگذری بالاتر، نشاندهنده بالا بودن چگالی محلی (یا موضعی) گره است.

مثلثهای کاملاً بسته[۴]، عملاً محافلی را ایجاد می‌کنند که می‌توانند بعنوان "انجمنها و یا اجتماعات" شناسایی شوند. تابع *nx.transitivity()* از طریق تابع *nx.triangles ()* ، تعداد مثلثهای موجود در گراف را محاسبه کرده و آن را بر کل تعداد مثلثهای ممکن در شبکه تقسیم می‌کند.

---

[۱] Closure

[۲] Triadic Analysis

[۳] Closed Triples

[۴] Completely Closed Triangles

مثال ۱

در این مثال، از شبکه قهرمانی استفاده خواهیم کرد.

```
In:  nx.transitivity(graph)
Out: 0.19453974709267596
```

مقادیر بالاتر ضرایب خوشه‌گی و تراگذری، ممکن است بدان معنا باشد که گراف موردنظر، اثر دنیای کوچک[۱] را به نمایش گذاشته است. "اثر دنیای کوچک"، یکی از خصوصیات شبکه‌های اجتماعی است. این ویژگی پیش‌بینی می‌کند که هر دو نفر، با بیش از شش پیوند از یکدیگر دور نیستند (یعنی حداکثر با شش پیوند یا یال، می‌توانند با هم ارتباط داشته باشند).
اکنون **شعاع** (یعنی کمترین طول مسیر بین هر دو گره) و **قطر** (یعنی بیشترین طول مسیر بین هر دو گره) را محاسبه می‌کنیم:

```
In:  nx.radius(graph)
Out: NetworkXError: Graph not connected:    infinite path length
In:  nx.diameter(graph)
Out: NetworkXError: Graph not connected: infinite path length
```

این نتیجه واضح بود؛ زیرا گراف ما، منفصل (یا ناهمبند) است. برای حل این مشکل، یک مولفه یا سازند متصل در گراف را به‌صورت زیر درنظر می‌گیریم:

```
In: sub_graph = nx.connected_component_subgraphs(graph)
```

بیایید یک تابع برای پیدا کردن "بزرگترین مولفه یا سازند" در این فهرست، ایجاد کنیم:

```
In:
    cur_graph = graph
    if not nx.is_connected(cur_graph):
        # get a list of unconnected networks
        sub_graph = nx.connected_component_subgraphs(cur_graph)

        main_graph = sub_graph[0]
```

[۱] Small-World Effect

```
# find the largest network in that list
for sg in sub_graph:
    if len(sg.nodes()) > len(main_graph.nodes()):
        main_graph = sg

cur_graph = main_graph
```

تابع *nx.connected_component_subgraphs (graph)* یک تابع مهم برای مجزا کردن مولفه‌های متصل (یا سازندهای همبند)[۱] است. این تابع یک آرایه از "اشیاء گراف" را باز می‌گرداند به گونه‌ای که هر شی مربوط به یکی از این مولفه‌های متصل است:

```
In:  len(graph)
Out: 6426
In:  [len(s) for s in sub_graph]
Out: [6408, 9, 7, 2]
In:  print(nx.number_connected_components(graph))
Out: 4
```

این نتیجه نشان می‌دهد که گراف ما شامل ۶۴۲۶ گره است، اما شبکه به چهار مولفه غیرمتصل (یا سازند ناهمبند) تقسیم شده‌است که بزرگترین مولفه آن، ۶۴۰۸ گره دارد. در واقع هر یک از این چهار بخش، یک زیرگراف را تشکیل می‌دهد که شامل مجموعه‌ای از گره‌ها و نیز یالهایی است که این گره‌ها را بهم پیوند می‌دهند. هیچ محدودیتی برای تعداد گره‌های تشکیل‌دهنده یک زیرگراف وجود ندارد، زیرا هر تعدادی از گره‌ها می‌توانند یک زیرگراف را ایجاد کنند. حالا بیایید "توزیع اندازه مولفه‌ها"[۲] را ببینیم (شکل ۱۰-۶).

سه مولفه از چهار مولفه، اندازه‌ای کوچکتر از ۱۰ دارند. هیچ سازند مجزا (یا مولفهٔ ایزوله‌ای)[۳] در این گراف وجود ندارد (بنا به تعریف، سازند ایزوله، عملاً مولفه‌ای با درجه صفر است)، البته سه مولفه با درجه

---

[۱] Connected Components

[۲] Distribution of the Component Sizes

[۳] Isolated Components

پایین داریم (۷,۹ و ۲). در این گراف، ما یک زیرگراف غول‌پیکر[۱] داریم؛ یعنی یک مولفه یا سازند بزرگی که بیشتر شبکه را پر می‌کند. در اینجا زیرگراف غول‌پیکر حدود ۹۹,۷ درصد از شبکه را اشغال کرده است.

در برنامه‌های کاربردی شبکه واقعی، مانند اینترنت، داشتن یک مولفه که تقریباً همه شبکه را پر می‌کند، امر مهمی است. در نتیجهٔ وجود این سازند گسترده، یک مسیر اتصالی در سراسر شبکه اینترنت بگونه‌ای فراهم می‌شود که امکان ارتباطات کامپیوتر به کامپیوتر را تقریباً به همه کاربران ارائه دهد. چنین ساختار شبکه‌ای، که در آن، شبکه به یک سازند بزرگ و چندین سازند کوچک دیگر تقسیم می‌شود، معمولاً در شبکه‌های غیرجهتدار یافت می‌شود.

بیایید ببینیم چگونه می‌توانیم آن دسته از زیرگرافها را استخراج کنیم که شرط خاصی مثل داشتن حداقل تعداد درجه گره را مد نظر دارند:

```
In:  [len(s) for s in nx.connected_component_subgraphs(graph) if len(s)
     > 5]
Out: [6408, 9, 7]
```

خروجی این کد نشان می‌دهد که سه مولفه (یا سازند) با اندازه بزرگتر از پنج وجود دارند. در این مورد خاص، می‌توانیم بزرگترین مولفه یعنی مولفه غول‌پیکر را بعنوان کل شبکه در نظر بگیریم.

شکل۱۰-٦ توزیع اندازه‌های مولفه مربوط به شبکه قهرمانی

[۱] Giant Subgraph

اکنون بیایید قطر و شعاع این بزرگترین مولفه را محاسبه کنیم:

```
In:  sub_graph = sub_graph[0]
In:  nx.diameter(sub_graph)
Out:
In:  nx.radius(sub_graph)
Out:
```

از سوی دیگر، پیاده سازی تابع *nx.connected_components (graph)* فهرستی از گره‌ها را برای هر مولفه گراف ایجاد می‌کند.

اکنون محاسبه معیار برون‌مرکزی (یا گریز از مرکز) امکانپذیر است: یعنی بیشینه کوتاهترین فاصله مسیر از هر گره در شبکه. محاسبه این معیار در زمانی که گراف بیش از یک مولفه داشت، امکان‌پذیر نبود.

```
In: ecc = nx.eccentricity(sub_graph)
Out:
```

مثال ۲

در این مثال، ما شبکه فیسبوک را بررسی خواهیم کرد.

```
In:  fb = nx.read_edgelist("…/facebook_combined_minimized.txt",
     create_using=nx.Graph(), nodetype=int)

In:  nx.radius(fb)
Out: 3

In:  nx.diameter(fb)
Out: 5

In:  print(nx.info(fb))
Out: Name:
     Type: Graph
     Number of nodes: 1820
     Number of edges: 7846
     Average degree: 8.6220
```

```
In:  nx.number_connected_components(fb)
Out: 1

In:  len(fb)
Out: 1820

In:  sub_graph = nx.connected_component_subgraphs(fb)

In:  [len(s) for s in sub_graph]
Ou:  [1820]
```

٦-٨ هسته‌ای بودن

در فصل یک گفته شد که در گرافهای بدون جهت، یک k -هسته‌ای، یک زیرگراف منتجه بیشین همبند است که مقدار بیشینه آن بزرگتر یا مساوی k است. این بدان معنی است که هر گره، با حداقل k گره دیگر ارتباط دارد.

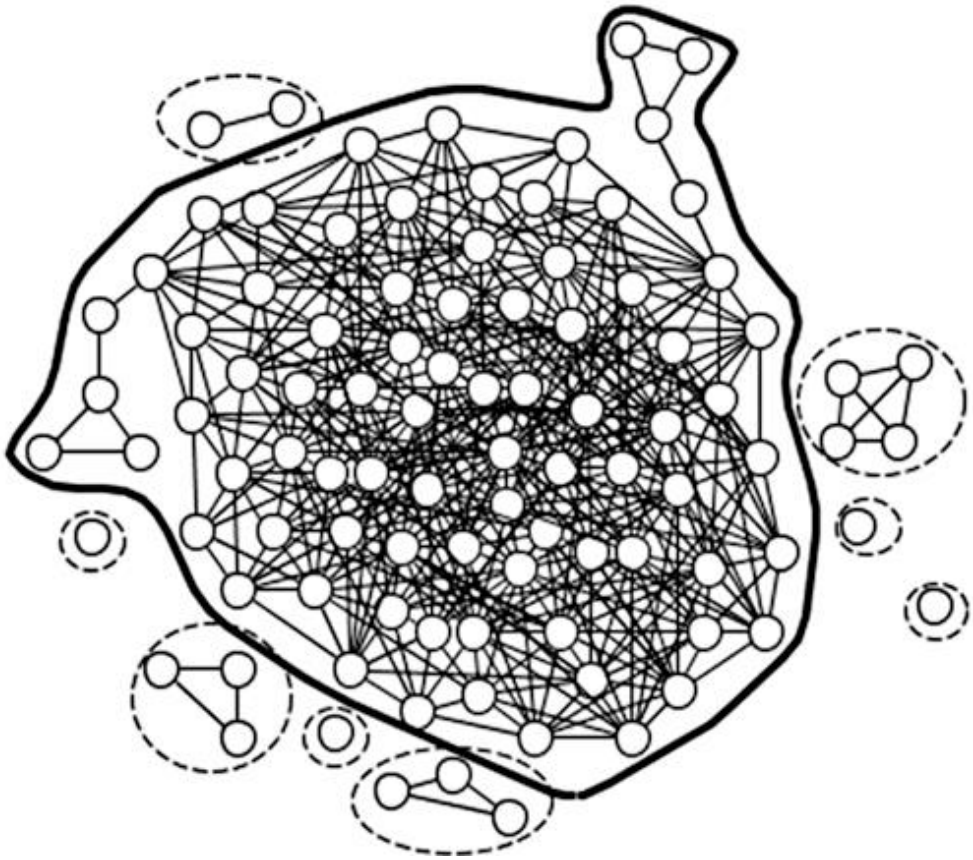
این زیرگرافها لزوماً گروه منسجم نیستند (و یا انسجام گروهی ندارند)، اما می‌توانند مناطقی را مشخص کنند که دارای ساختارهای محفل-مانند هستند. تعیین k-هسته‌ایها را می‌توان بعنوان راهی برای یافتن زیرگروههای منسجم در گرافها به کار برد.

حال، طبق تعریف، میزان هسته‌ای بودن یک گره[۱] (یا درجه هسته[۲]) را می‌توان c درنظر گرفت اگر آن گره، متعلق به c-هسته‌ای بوده، اما به $(c+1)$-هسته‌ای متعلق نباشد.

کاربرد این معیار آنست که دریابیم یک گره با چه قدرتی به شبکه متصل است. محاسبه میزان هسته‌ای بودن، منجر به طبقه‌بندی گره‌ها در چندین لایه می‌شود که برای تجسم مفید است (شکل ۱۱-۶).

[۱] Node's Coreness

[۲] Core Number



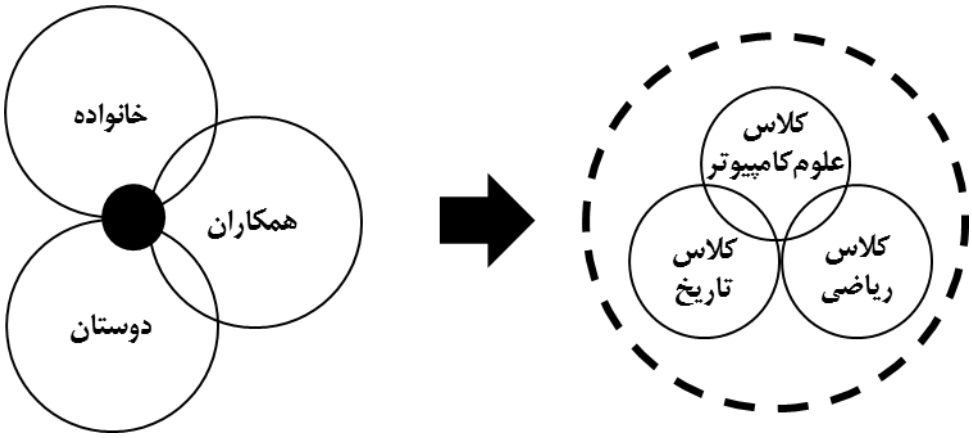
شکل۱۱-٦ هسته

٦-۹ اجتماعات همپوشان

آیا "اجتماعات متمایز غیرهمپوشان"[۱] در شبکه‌های‌اجتماعی تجربی معنایی دارند؟ اجتماعات همپوشان ممکن است در سطوح مختلف دانه‌بندی[۲] وجود داشته باشند (شکل ۱۲-۶).

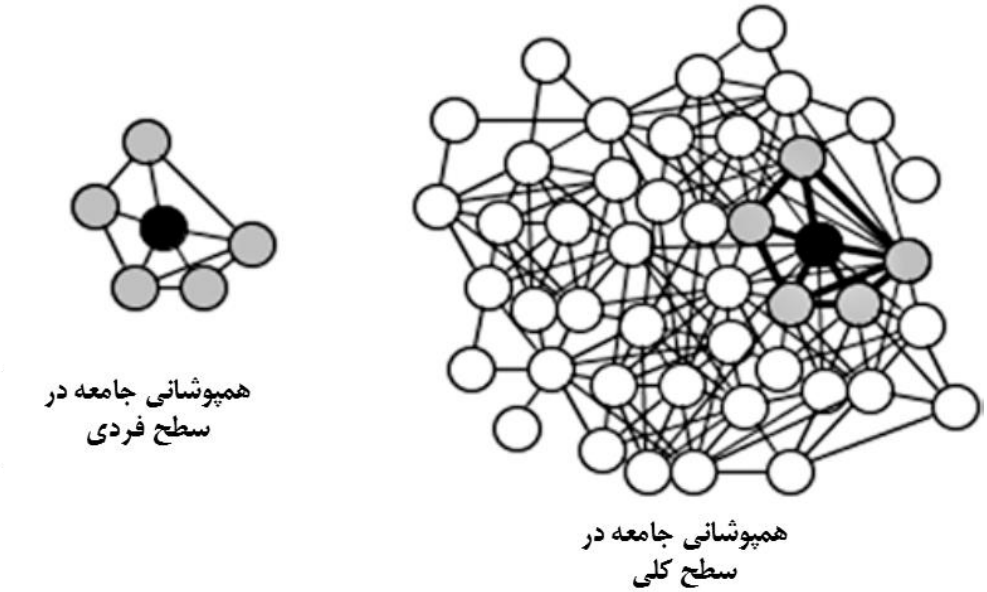
[۱] Disjoint Non-Overlapping Communities

[۲] Granularity



شکل ۱۲-٦ اجتماعات همپوشان

"اجتماعات متمایز غیرهمپوشان" به ندرت در مقیاسهای بزرگ مثلاً در بسیاری از شبکه‌های تجربی وجود دارند. اجتماعات به‌طور فراگیری همپوشانی دارند و این امر بخش‌بندی شبکه‌ها را بدون جداسازی اجتماعات، غیرممکن می‌کند (شکل ۱۳-۶).

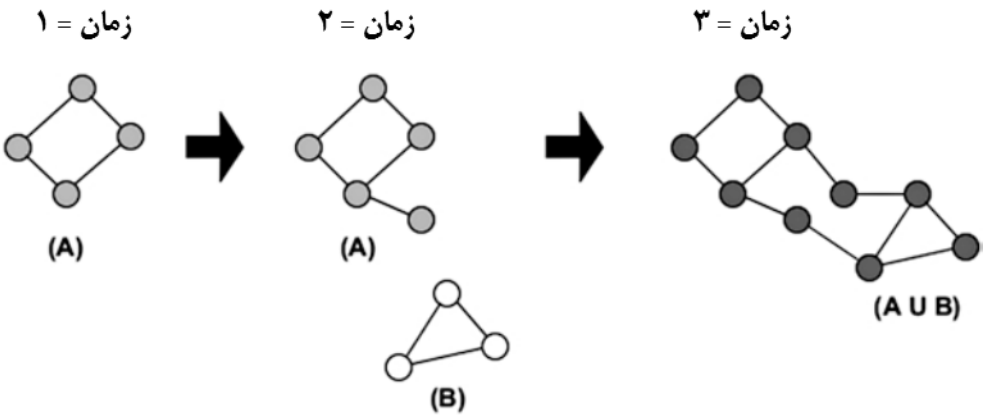


شکل ۱۳-٦ همپوشانی جامعه در سطح فردی (فردانه) و جمعی (جمعانه)

هیچ پشتیبانی درون–سازی[۱] برای الگوریتمهای همپوشان وجود ندارد، اما ما می‌توانیم از ابزار *MOSES* برای تجزیه و تحلیل گرافها که بعنوان "فهرست یال" نشان‌داده شده، استفاده کنیم. *MOSES* رویکردی مقیاس‌پذیر[۲] برای شناسایی اجتماعاتی با همپوشانی بالا است. در این الگوریتم: (۱) به طور تصادفی یک یال را انتخاب کرده و در اطراف این یال، یک اجتماع را به طرز حریصانه‌ای گسترش دهید بگونه‌ای که یک تابع هدف را بهینه کنید. (۲) اجتماعات با کیفیت پایین را حذف کنید. (۳) با تخصیص مجدد گره‌ها، اجتماعات را با دقت تنظیم کنید.

٦–۱۰ یافتن اجتماع پویا

در بسیاری از اوقات هدف ما از تحلیل شبکه‌های اجتماعی (*SNA*) آنست که می‌خواهیم تحلیل کنیم چگونه اجتماعات یا انجمنها در شبکه شکل گرفته و در طول زمان تکامل می‌یابند. ما غالباً این تحلیل را به صورت "آفلاین" و با بررسی عکسهای متوالی[۳] شبکه انجام می‌دهیم (شکل ۱۴–۶).



شکل ١٤–٦ جامعه کوچک در سه مرحله تکامل یافته است

[۱] Built-in Support

[۲] Scalable Approach

[۳] Successive Snapshots

"اجتماعات پویا" را می‌توان براساس رویدادهای کلیدی چرخه حیات آنها دسته‌بندی کرد. رویدادهای چرخه حیات عبارتند از: (۱) تولد و مرگ اجتماعات، (۲) انبساط و انقباض اجتماعات[۱] و (۳) ترکیب و تجزیه اجتماعات[۲]. این مکانیزم به شرح زیر است: الگوریتم اجتماع‌یابی را برای هر عکس فوری از گراف بکار ببرید، سپس اجتماعات تازه تولید شده در هر مرحله را با اجتماعاتی که در گذشته شناسایی شده‌اند، مطابقت دهید.

٦-۱۱ *m*–قطاع

m–قطاع یک زیرشبکه بیشین است که دارای خطوطی با چندگانگی یا کثرتی[۳] برابر (یا بیشتر) با *m* بوده و نیز رئوسی که محل تلاقی این خطوط هستند. هدف اینکار شناسایی "زیرگروههای منسجم" است که البته به کثرت (یا تعدد) خطوط بستگی دارند. این معیار اولین بار توسط جان اسکات[۴] معرفی شد، که آن را ابتدائاً *m*–هسته[۵] نامید. بر اساس این معیار، گره‌های منزوی (ایزوله) به صورت ۰–قطاع تعریف می‌شوند، زیرا چیزی به اشتراک نمی‌گذارند. با این وجود، *m*–قطاع تضمین نمی‌کند که تمام راسها در مولفهٔ *m*–قطاع با حداقل سطح چندگانگی (یا کثرت) خطوط، متصل شده باشند. در عوض باید توجه داشت این مولفه‌های *m*–قطاع هستند که زیرگروه منسجم تلقی می‌شوند و نه خود *m*–قطاع.

اگر بخواهیم زیرگروههای منسجم را در درون یک *m*–قطاع بیابیم یعنی بدنبال رئوسی باشیم که با خطوطی در حداقل کثرت معین، بهم متصل شده‌اند، باید همه خطوط با کثرت یا شدتی کمتر از *m* و نیز رئوسی که متعلق به *m*–قطاع نیستند را حذف کرده و مولفه‌های ضعیف در زیرگروه منتجه را شناسایی کنیم. حال هر مولفه شامل رئوسی است که مسقیماً یا غیرمستقیم با یالهایی پیوند خورده‌اند که دارای کثرت یا شدت موردنظر هستند.

---

[۱] Expansion and Contraction of Communities

[۲] Merging and Splitting of Communities

[۳] Multiplicity

[۴] John Scott

[۵] M-Core

٦-۱۲ k-هسته‌ها

k-هسته یک مولفه متصل (یا سازند همبند) در شبکه است که با حذف تمام گره‌هایی که درجه آنها کمتر از k است، بدست می‌آید؛ البته این حذف مکرراً ادامه می‌یابد تا زمانی که چنین گره‌هایی دیگر وجود نداشته باشند. این فرآیند، باعث می‌شود تا تمام گره‌های k-هسته‌ای دارای حداقل درجه k باشند. معیار k-هسته کمک می‌کند تا مکانِ خوشه‌های هسته‌ای[۱] شناسایی شود. بزرگترین مقدار k، که برای آن یک k-هسته وجود دارد، "فرو اُفت یا تباهیدگی شبکه"[۲] نامیده می‌شود.

٦-۱۳ اجتماع‌یابی

اجتماع یا انجمن، زیرگرافی از یک شبکه است که در آن اتصال گره‌ها به یکدیگر، چگالتر از اتصالشان در فضای خارج است.

پیدا کردن اجتماعات[۳] در شبکه‌ها (یا اتجمن‌یابی) به معنای شناسایی مکانهایی در شبکه است که در آنها، راسها در درون خوشه‌ها به‌گونه‌ای سازماندهی می‌شوند که بسیاری از یالها، راسهای داخل خوشه‌های خاصی را بهم متصل کرده و یالهای کمتری، راسهای موجود در سایر خوشه‌ها را بهم وصل می‌کنند. (در واقع هر یک از این خوشه‌های خاص، نشاندهنده یک اجتماع هستند). این موضوع تبدیل به یک حوزه پژوهشی فعال برای بسیاری از دانش پژوهان و محققان شده است.

الگوریتمهای بسیاری برای شناسایی اجتماعات یا انجمن‌یابی ارائه شده اند که از جمله آنها (۱) الگوریتم دوبخشی طیفی[۴] است (که براساس خواص طیفی از ماتریس لاپلاس[۵] عمل می‌کند) و (۲) الگوریتم *Kernighan-Lin* (که یکی از اولین رویکردهای پیشنهادی است و هنوز هم اغلب در ترکیب با سایر

---

[۱] Core Clusters

[۲] Degeneracy of the Network

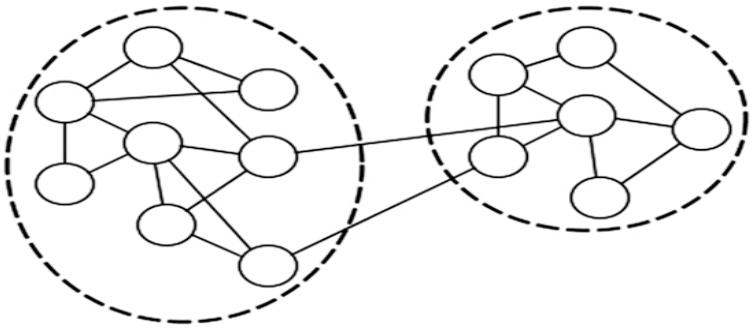
[۳] Finding Communities

[۴] Spectral Bisection Algorithm

[۵] Laplacian Matrix

روشها، مورد استفاده قرار می‌گیرد. این الگوریتم بر مبنای ایدهٔ بخش‌بندی مدارات الکترونیکی[۱] بر روی بردها است). سایر روشهای محبوب برای بخش‌بندی گراف شامل الگوریتم هندسی، بخش‌بندی سطح–ساختار[۲] و الگوریتمهای چندسطحی[۳] است.

انجمن‌یابی شامل شناسایی گروهی از گره‌هایی است که خواص مشترک دارند و/یا نقش مشابهی در گراف دارند (فرتناتو[۴] ۲۰۱۰). چنین گروههایی دارای یک اتصال گره–گره متراکم یا چگال هستند، در حالیکه اتصالات گره–گره با تراکم یا چگالی کمتری با دیگر گروهها دارند (شکل ۱۵–۶).



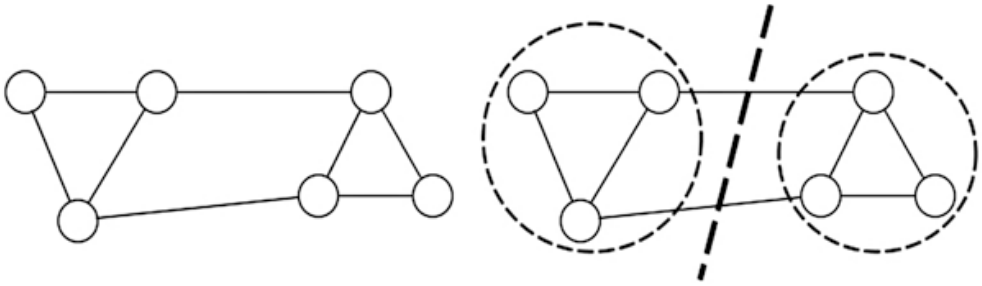
شکل ۱۵–٦ شناسایی اجتماعات در شبکه‌ها

٦–۱۳–۱ بخش‌بندی گراف

بخش‌بندی گراف[۵] با هدف تقسیم گره‌های گراف به تعداد معینی از "گروه گره‌ها"، بمنظور بهینه‌سازی معیار مربوط به تعداد "برش یالها" انجام می‌شود. برش کمینه[۶] یک رویکرد سنتی برای بخش‌بندی گراف است که شامل کمینه‌کردن تعداد (یا وزن) "برش یالها" در بخش‌بندی آنها می‌شود. روشهای

[۱] Partitioning Electronic Circuits
[۲] Level-Structure Partitioning
[۳] Multilevel Algorithms
[۴] Fortunato
[۵] Graph Partitioning
[۶] Min-Cut

پیشرفته‌تر از برشهای نرمال[۱] استفاده می‌کنند و استراتژیهای چندسطحی را برای مقیاس به نمودارهای بزرگ اعمال می‌کنند (شکل ۱۶-۶).



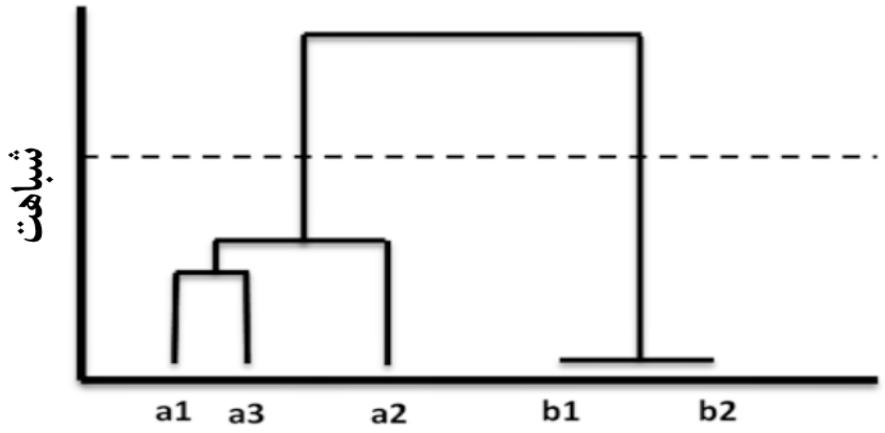
شکل ۱۶-۶ بخش‌بندی گراف

۶-۱۳-۲ خوشه‌بندی سلسله‌مراتبی

علوم اجتماعی، رویکرد متفاوتی را برای یافتن اجتماعات یا شناسایی انجمنها اتخاذ کرده‌اند که بر پایهٔ اندازه‌گیری شباهت بین زوج راسها با کمک برخی از معیارهای سنجش شباهت است.

خوشه‌بندی سلسله‌مراتبی یک روش خوشه‌بندی آماری است که از آن برای خوشه‌کردن زوج گره‌هایی استفاده می‌شود که بر اساس وضعیت مورد نظر، ساختار تقریباً یکسانی دارند. این روش، امکان بررسی ویژگیهای ساختاری در شبکه‌های اجتماعی را فراهم می‌کند. الگوریتم مذکور در ابتدا، گره‌هایی را گروه‌بندی می‌کند که بسیار مشابه هستند. سپس، جفت گره‌ها یا گروه‌هایی را خوشه‌بندی می‌کند که بیشترین شباهت را دارند. این امر آنقدر تکرار می‌شود تا همه گره‌ها کنار هم قرار گرفته و یا اصطلاحاً بهم ملحق شوند. الگوریتم مذکور با ایجاد یک ساختار داده مبتنی بر درخت دودویی به نام نمودار دندانه‌ای یا دندروگرام ، مسئله را حل می‌کند (شکل ۱۷-۶).

[۱] Normalized Cuts



شکل ۱۷-٦ خوشه‌بندی سلسله‌مراتبی

در شکل ۱۷-۶ یک دندروگرام، نتایج خوشه‌بندی سلسله‌مراتبی را به تصویر کشیده است. همانگونه که مشهود است گره‌های *b1* و*b2* برای وهله نخست، به هم ملحق می‌شوند زیرا آنها کاملاً مشابه هستند (عدم تشابه = صفر). سپس، گره‌های *a1* و *a3* با هم خوشه می‌شوند، زیرا عدم شباهت[۱] آنها کم است. سپس گره *a2* به خوشه *a1* و *a3* اضافه می‌شود. در مرحله چهارم و نهایی فرآیند خوشه‌بندی، این خوشه با خوشهٔ گره‌های *b1* و*b2* ادغام می‌شود. در یک نمودار دندانه‌ای، هر چه شاخه بالاتر باشد، نشان می‌دهد که دو گره یا خوشه (در لحظه‌ای که آنها خوشه‌بندی می‌شوند)، تفاوت بیشتری دارند.

اگر بخواهیم یک نمودار دندانه‌ای را به گره‌های اولیه‌اش تقسیم کنیم، لازم است دستوراتی که براساس آنها دندروگرام ساخته شده را بصورت معکوس اجرا کنیم. یعنی ابتدا خوشه بزرگی که از الحاق دو خوشه (*a1*، *a2*، *a3*) و (*b1* و*b2*) شکل گرفته بود را به اجزایش یعنی دو خوشه مذکور تقسیم می‌کنیم. سپس، خوشه‌ای را که حاوی گره‌های *a1*، *a2*، *a3* است، تجزیه می‌کنیم. در نهایت، خوشه‌ای را که حاوی گره‌های *a1* و *a2* است، جداسازی می‌کنیم. یک قانون کلی برای تقسیم دندروگرام این است که خوشه‌ها در مکان یا مکانهایی تجزیه شوند که این انشعاب یا شاخه‌بندی جهش قابل توجهی را بدست می‌دهد. یعنی ما ابتدا باید خوشهٔ گره‌هایی را شناسایی کنیم که بطور ساختاری کاملاً معادل یا تقریباً معادل هم هستند.

[۱] Dissimilarity

مثال ۱

برنامه بازگشتی ساده زیر نشان می‌دهد که چگونه می‌توان یک سلسله مراتب از گرافها (یک ساختار دندروگرام–مانند) را با استفاده از *NetworkX* ساخت. کد مربوطه که توسط *Joel* آماده شده و اینک با برخی اصلاحات، ارائه می‌شود، در آدرس زیر آمده است.

[https://stackoverflow.com/users/2966723/joel].

```
In:
    import networkx as nx
    def hierarchy_pos(G, root, width=1., vert_gap = 0.2, vert_loc = 0,
    xcenter = 0.5 ):
        def h_recur(G, root, width=1., vert_gap = 0.2, vert_loc = 0,
    xcenter = 0.5,
                    pos = None, parent = None, parsed = [] ):
            if(root not in parsed):
                parsed.append(root)
                if pos == None:
                    pos = {root:(xcenter,vert_loc)}
                else:
                    pos[root] = (xcenter, vert_loc)
                neighbors = G.neighbors(root)
                if parent != None:
                    neighbors.remove(parent)
                if len(neighbors)!=0:
                    dx = width/len(neighbors)
                    nextx = xcenter - width/2 - dx/2
                    for neighbor in neighbors:
                        nextx += dx
                        pos = h_recur(G,neighbor, width = dx, vert_gap =
    vert_gap, vert_loc = vert_loc-vert_gap, xcenter=nextx, pos=pos,
    parent = root, parsed = parsed)
            return pos

        return h_recur(G, root, width=1., vert_gap = 0.2, vert_loc = 0,
    xcenter = 0.5)
```

توضیحاتی پیرامون تعریف تابع بالا:

۱. اگر یک چرخه‌ای وجود دارد که از ریشه قابل دسترسی است، آنگاه نتیجه یک سلسله مراتب نخواهد بود.

۲. G: گراف.

۳. ریشه (*root*): گره ریشه شاخه جاری

۴. عرض (*width*): فضای افقی اختصاص یافته به این شاخه – از همپوشانی با شاخه‌های دیگر اجتناب می‌کند.

۵. *vert_gap* : فاصله بین سطوح سلسله‌مراتبی.

۶. *vert_loc* : موقعیت عمودی ریشه[۱].

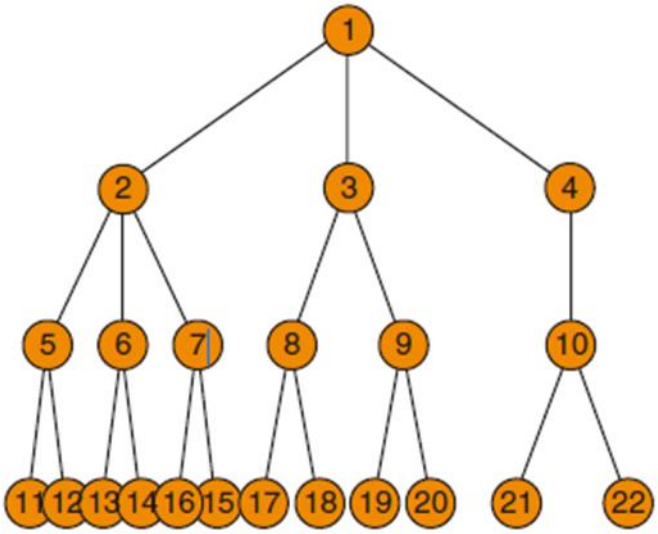
۷. *Xcenter* : موقعیت افقی ریشه[۲].

این مثالی است که نشان می‌دهد چگونه می‌توان از کد بالا استفاده کرد (شکل ۱۸–۶):

```
In:
     G=nx.Graph()
     G.add_edges_from([(1,2), (1,3), (1,4), (2,5), (2,6), (2,7), (3,8),
     (3,9), (4,10), (5,11), (5,12), (6,13), (6, 14), (7, 15), (7, 16), (8,
     17), (8, 18), (9, 19), (9, 20), (10, 21), (10, 22)])
     pos = hierarchy_pos(G,1)
     nx.draw(G, pos=pos, with_labels=True, node_size= 500,
     node_color='orange')
Out:
```

[۱] Vertical Location of Root

[۲] Horizontal Location of the Root



شکل ۱۸-٦ سلسله‌مراتب گراف

با این حال، چگونه می‌توانیم فاصله بین دو خوشه را محاسبه کنیم؟ در عمل، روشهای متعددی برای حل این مسئله وجود دارد. بیایید سه مورد از آنها را بررسی کنیم:

- **پیوند تکی**[۱]: این روش آن دو خوشه‌ای را با هم ترکیب می‌کند که کوچکترین حداقل فاصله بین جفت گره‌های دو خوشه را دارند.

- **پیوند متوسط**[۲]: این روش آن دو خوشه‌ای را با هم ترکیب می‌کند که کوچکترین میانگین فاصله بین جفت گره‌های مربوط به هر دو خوشه را دارند.

- **پیوند کامل**[۳]: این روش آن دو خوشه‌ای را با هم ترکیب می‌کند که کوچکترین حداکثر فاصله بین جفت گره‌های دو خوشه را دارند.

[۱] Single Linkage

[۲] Average Linkage

[۳] Complete Linkage

خوشه‌بندی سلسله‌مراتبی را می‌توان از دو روش مختلف بدست آورد: خوشه بندی از پایین به بالا (همجوشی یا ترکیبی) و از خوشه بندی بالا به پایین (جدایشی یا تجزیه‌ای). با وجود اینکه هر دو رویکرد از مفهوم دندروگرام برای انجام خوشه‌بندی استفاده می‌کنند، ممکن است بدلیل معیار استفاده شده در طی فرایند خوشه‌بندی، نتایج کاملاً متفاوتی داشته باشند.

۱. همجوشی یا ترکیبی: در این روش از یک رویکرد پایین به بالا برای ساخت درخت استفاده می‌شود. یعنی الگوریتم از داده‌های منفرد شروع می‌کند (در واقع هر داده در ابتدا بعنوان یک خوشه در نظر گرفته می‌شود) و با ادغام خوشه‌ها، فرآیند ادامه پیدا می‌کند تا اینکه همه اشیا در یک خوشه (خوشه نهایی) قرار گیرند. ادغام خوشه‌ها با استفاده از یکی از تکنیکهای ادغام به‌طریقی‌که توازنی میان بهره‌وری و کیفیت ایجاد شود، صورت می‌گیرد. نمونه‌ای از تکنیکهای ادغام عبارتند از: پیوند تکی، پیوند نمونه‌گیری[۱]، پیوند همه‌جانبه[۲] و پیوند مرکزی[۳]. با شروع از هر گره بعنوان یک خوشه جداگانه، رویکرد پایین به بالا، شبیه‌ترین جفت خوشه‌ها را در هر مرحله ادغام می‌کند.

۲. جدایشی یا تجزیه‌ای: در این روش از یک رویکرد بالا به پایین برای بخش‌بندی داده‌ها به مولفه‌های کوچکتر استفاده می‌شود تا نهایتاً یک ساختار درخت–مانند[۴] ایجاد شود. کار با یک خوشهٔ تکی که حاوی تمام گره‌های شبکه است، آغاز می‌شود، با بکارگیری رویکرد بالا به پایین، آن خوشه انتخاب شده به دو زیرخوشه[۵] در هر مرحله تقسیم می‌شود.

مثال ۲

بیایید ببینیم چگونه خوشه‌بندی سلسله‌مراتبی در این مثال با استفاده از داده‌های ساده کار می‌کند. از آنجا که تاکنون کتابخانه *NetworkX* هیچ پیاده‌سازی مستقیمی برای خوشه‌بندی سلسله مراتبی ندارد، ما از خوشه‌بندی سلسله‌مراتبی موجود در بسته *SciPy* استفاده خواهیم کرد. کد اصلی توسط تاروساتو[۶] *[http://okomestudio.net/]* نوشته شده است (شکل۱۹–۶).

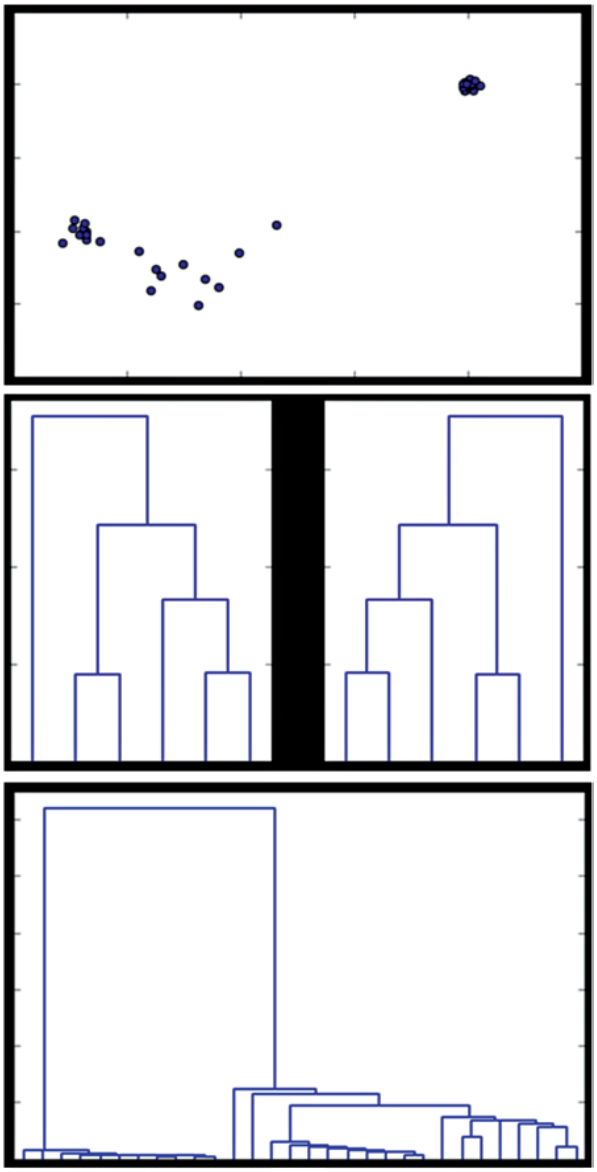
```
In:
```

[۱] Sampled Linkage
[۲] All-Pairs Linkage
[۳] Centroid Linkage
[۴] Tree-Like Structure
[۵] Sub-Clusters
[۶] Taro Sato

```
 import matplotlib.pyplot as plt
 from scipy.cluster.hierarchy import dendrogram, linkage
 from numpy import array
 import numpy as np

 mat = array([184, 222, 177, 2162,3 1,
              45, 123, 128, 200,
              129, 121, 203,
              46, 83,
              83])
 dist_mat = mat

 linkage_matrix = linkage(dist_mat, 'single')
 print linkage_matrix

 plt.figure(101)
 plt.subplot(1, 2, 1)
 plt.title("ascending")
dendrogram(linkage_matrix,
           color_threshold=1,
           truncate_mode='lastp',
           labels=array(['a', 'b', 'c', 'd', 'e', 'f']),
           distance_sort='ascending')
plt.subplot(1, 2, 2)
plt.title("descending")
dendrogram(linkage_matrix,
         color_threshold=1,
         truncate_mode='lastp',
         labels=array(['a', 'b', 'c', 'd', 'e', 'f']),
         distance_sort='descending')
def make_fake_data():
    amp = 1000.
    x = []
    y = []
    for i in range(0, 10):
        s = 20
        x.append(np.random.normal(30, s))
        y.append(np.random.normal(30, s))
```

```
        for i in range(0, 20):
            s = 2
            x.append(np.random.normal(150, s))
            y.append(np.random.normal(150, s))
        for i in range(0, 10):
            s = 5
            x.append(np.random.normal(-20, s))
            y.append(np.random.normal(50, s))
        plt.figure(1)
        plt.title('fake data')
        plt.scatter(x, y)

        d = []
        for i in range(len(x) - 1):
            for j in range(i+1, len(x) -1):
                d.append(np.sqrt(((x[i]- x[j])**2 + (y[i]-y[j])**2)))
        return d
    mat = make_fake_data()

    plt.figure(102)
    plt.title("Three Clusters")

    linkage_matrix = linkage(mat, 'single')
    print "three clusters"
    print linkage_matrix

    dendrogram(linkage_matrix,
               truncate_mode='lastp',
               color_threshold=1,
               show_leaf_counts=True)
    plt.show()
Out:
    [[ 1. 2. 45. 2.]
     [ 3. 4. 46. 2.]
     [ 5. 7. 83. 3.]
     [ 6. 8. 121. 5.]
     [ 0. 9. 177. 6.]]
    three clusters
```

```
[[ 15. 22. 0.33632604 2. ]
 [ 20. 26. 0.39356172 2. ]
 [ 10. 18. 0.62552494 2. ]
 [ 17. 29. 0.80644649 2. ]
 [ 25. 42. 0.85853812 3. ]
 [ 16. 40. 0.8586993 3. ]
 [ 11. 12. 0.91381892 2. ]
 [ 39. 44. 1.06065447 5. ]
 [ 21. 27. 1.08475839 2. ]
 [ 28. 43. 1.12965956 4. ]
 [ 19. 46. 1.26167457 6. ]
 …
```



شکل ۱۹-٦ خوشه‌بندی سلسله‌مراتبی با استفاده از *Scipy*

در این مثال از برنامه‌های زیر استفاده شده است:

- *matplotlib.pyplot*، برای رسم نتایج

- *dendrogram* و *linkage* از *scipy.cluster.hierarchy*، برای ساخت دندروگرامها و تعریف روش پیوند
- *array* از *numpy*، برای تعریف ماتریس فاصله (مسافت)[۱]
- بسته *numpy*، برای تولید اعداد تصادفی

این مثال از روش "پیوند تکی" برای تعریف فاصله میان خوشه‌ها استفاده کرده و آنرا با دو نوع شیوه: صعودی[۲] و نزولی[۳] پیاده‌سازی کرده است. البته می‌توان گزینه‌های دیگر مانند پیوند متوسط، پیوند کامل و یا پیوند وزنی را امتحان کرد. هرچند تعداد خوشه‌ها که ۳ است به همین صورت باقی خواهد ماند، ولیکن نتایج کاملاً متفاوتی بدست خواهند آمد.

دستور *make_fake_data()* یک تابع است که داده‌های تصادفی را برای این مثال تولید می‌کند. بیایید نگاهی بیندازیم به داده‌های نمونه‌ای که از اجرای این تابع تولید شده است:

```
In:  mat
Out: [45.245034231790527,
      52.318030166215507,
      62.669312651126731,
      61.023428681366973,
      44.312064158396517,
      58.228649487002663,
      74.700976436821762,
      35.956497691214871,
      33.423710737357311,
      206.50594326367511,
      206.46813737653895,
      207.32596327946229,
      210.9789592576071,
      206.63628599779912,
      …]
```

[۱] Distance Matrix

[۲] Ascending

[۳] Descending

```
In: len(mat)
Out: 741
```

٦-١٤ مدلهای بلوکه‌ای

مدل بلوکه‌ای، نسخه کوچکتری از یک شبکه بزرگتر است که در آن گره‌ها بر اساس یک مدل بخش‌بندی خاص، بر روی هم فروریخته[۱] می‌شوند. هر دسته یا بخش از گره‌ها (اصطلاحاً بلوک)، در این شبکه جدید، بعنوان یک گره نمایش داده می‌شود. به طور مشابه، ارتباط بین گره‌ها در شبکه اصلی، با هم تجمیع شده تا اتصال بین بلوکها در شبکه جدید را نشان دهد. رویکرد مدل بلوکه‌ای مهم است زیرا ابزار موثری برای توصیف ساختار کلی شبکه و موقعیت هر گره در این ساختار است. فرایند تولید شبکه‌ای کوچکتر از یک شبکه بزرگتر، مدلسازی بلوکه‌ای[۲] نامیده می‌شود.

هدف مدلسازی بلوکه‌ای تبدیل یک شبکه بزرگ بالقوه نامتجانس و ناسازگار به یک ساختار کوچکتر و قابل درک بگونه‌ای است که بسادگی قابل تفسیر باشد. این رویکرد براساس یک رویه تجربی و بر این مبناست که واحدها یا گره‌ها در یک شبکه را می‌توان بصورتی گروهبندی کرد که بر اساس برخی تعاریف معنادار از هم‌ارزی[۳]، همچنان معادل شبکه اصلی باشد. فارغ از تعریفی که برای هم‌ارزی ارائه می‌شود، دو رویکرد مبنایی برای هم‌ارزی واحدها در یک شبکه خاص وجود دارد:

- گره‌ها یا واحدهای هم‌ارز الگوی ارتباط یکسان با همسایگان یکسان دارند.
- گره‌ها یا واحدهای هم‌ارز دارای الگوی ارتباط یکسان و یا مشابه اما با همسایگان متفاوت دارند.

نوعِ نخست هم‌ارزی با ایده "هم‌ارزی ساختاری"[۴] فرموله می‌شود و نوع دوم آن با ایده "هم‌ارزی منظم"[۵] فرموله می‌شود. در واقع نوعِ دوم هم‌ارزی، حالت تعمیم یافته نوع اول است.

---

[۱] Collapsed
[۲] Blockmodeling
[۳] Equivalence
[۴] Structural Equivalence
[۵] Regular Equivalence

بدین ترتیب، "مدلسازی بلوکه‌ای" روشی است که بسیار شبیه "خوشه‌بندی سلسله‌مراتبی" است. این نحوه مدلسازی بدین دلیل توسعه یافته تا برخی از ویژگیهای ساختاری شبکه‌های اجتماعی مانند انسجام، هسته/حاشیه و رتبه‌بندی را شناسایی کند. "فرایند مدلسازی بلوکه‌ای" عبارتست از گروهبندی گره‌ها به خوشه‌ها و تعیین روابط بین این خوشه‌ها. برای انجام این فرایند معمولاً از ماتریسها (بعنوان مثال، ماتریسهای مجاورت) جهت انجام محاسبات و تجسم داده استفاده می‌شود. این روش، یک تکنیک انعطاف‌پذیر و کارآمد برای تجزیه و تحلیل شبکه‌های اجتماعی متراکمِ کوچک است. با این روش، کنشگرانی که بر مبنای روابط هم‌ارزی دارای مشخصات ساختاری مشترک هستند در کلاسها و یا خوشه‌های معادل (هم‌ارز) قرار می‌گیرند و روابطشان (با قرار گرفتن) در درون بلوکها، بخش‌بندی می‌شود؛ به همین دلیل آنرا مدلسازی بلوکه‌ای گویند. خوشه‌های کنشگران هم‌ارز عملاً موقعیتهای شبکه را تعریف کرده و بلوکه‌ها، نشاندهنده موقعیتها[۱] و نقشها هستند.

در حالیکه روشهایی مانند "معیارهای مرکزیت" براساس موقعیتهای ساختاری گره‌ها هستند، اما مدلسازی بلوکه‌ای، نما و چشم‌انداز متفاوتی و بر اساس مفاهیم دیگری فراهم می‌کند. این روش بر مبنای "مفاهیم ساختاری" همچون هم‌ارزی و موقعیت (که با مفاهیم نظری همچون نقش اجتماعی و مجموعه‌های نقش[۲] مرتبط می‌باشد) بنا شده است و عملاً نقشهای اجتماعی و الگوهای مربوط به ارتباطات در شبکه را توصیف می‌کند. با این حال، تکنیکهای دیگری بجز مدلسازی بلوکه‌ای همچنان برای کار بر روی شبکه‌های بزرگ و یا تُنُک معتبر هستند.

پیاده سازی مدلسازی بلوکه‌ای به دو شیوه امکانپذیر است. در روش اولِ پیاده‌سازی، لازم است که ما از قبل مدل بلوکه‌ای شبکه را بشناسیم (یعنی دو چیز را بدانیم: یکی، نحوه تخصیص "گره‌ها به کلاسها" و دیگری، "ماتریس تصویر"[۳] که مشخص کننده انواع بلوک مجاز است). در روش دوم پیاده‌سازی، فرض آنست که یک شبکه وجود دارد و ما می‌خواهیم مدل بلوکه‌ای که دربردارنده ساختار آن شبکه است را پیدا کنیم. در مثالهای زیر فرض می‌کنیم که مدل بلوکه‌ای این شبکه‌ها را می‌دانیم.

---

[۱] Position

[۲] Role Sets

[۳] Image Matrix

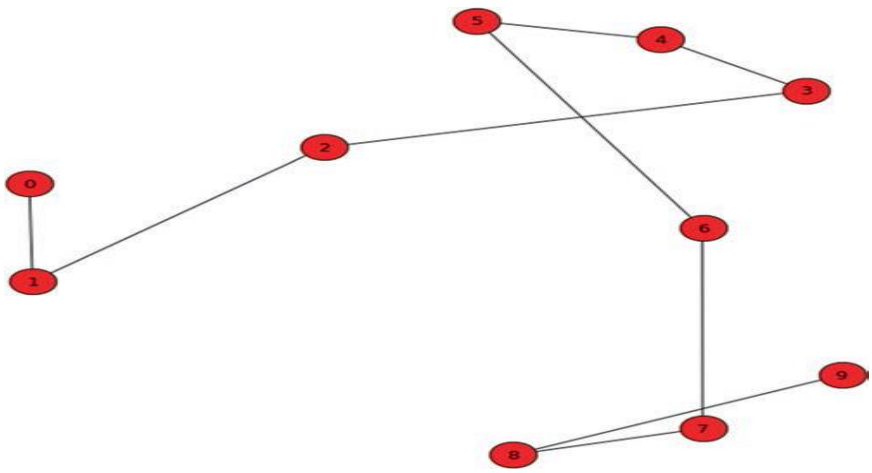
مثال ۱

یک روش پیاده سازی مناسب برای مدلهای بلوکه‌ای (*nx.blockmodel(G,partition)*) در کتابخانه *NetworkX*، وجود دارد. این تابع با استفاده از روش "مدلسازی بلوکه‌ای تعمیم‌یافته" یک گراف کاهش‌یافته را باز می‌گرداند. پارامترهای تابع عبارتند از:

- *G* (گراف یا دایگراف (گراف جهتدار) کتابخانه *NetworkX)*
- بخش یا افراز[۱] (افراز غیرهمپوشانی گره‌ها)
- گراف چندگانه (که این پارامتر اختیاری بوده و ارزش بولین[۲] را مشخص می‌کند.)

از طرف دیگر تابع مدل بلوکه‌ای، شیئ گراف *NetworkX*، را باز می‌گرداند.

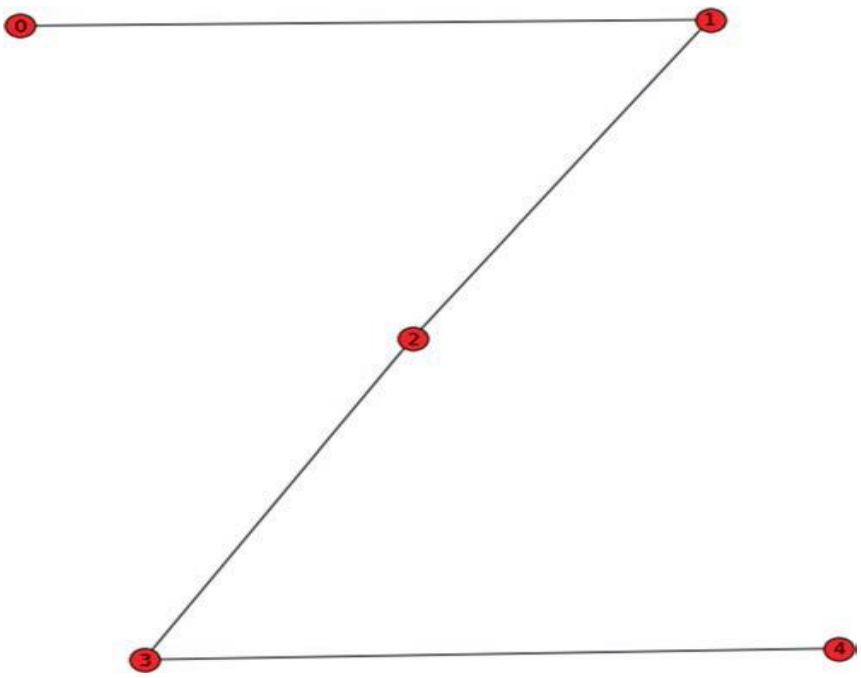
```
In:  G=nx.path_graph(10)
     nx.draw_networkx(G)
Out:
```



شکل ۲۰-٦ گراف مسیر ساده با ده گره

[۱] Partition

[۲] Boolean Value

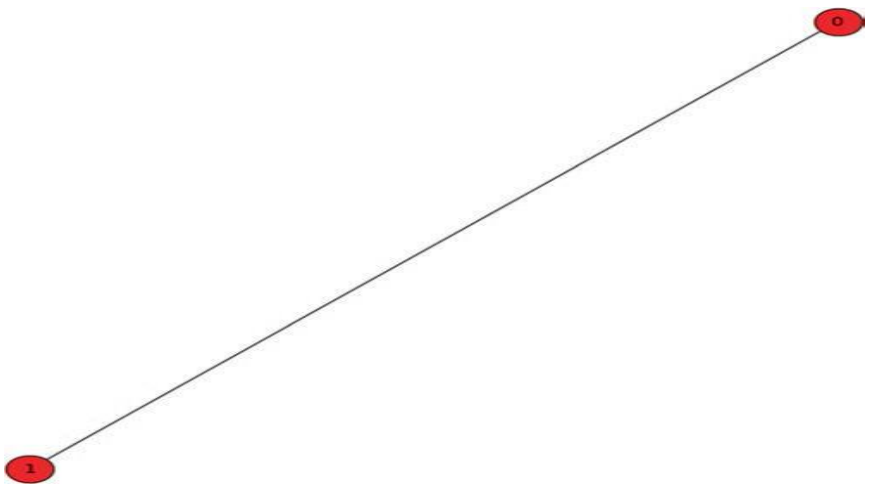


شکل ۲۱-٦ نمایش مدل بلوکه‌ای گراف مسیر

```
In:  partition = [[0,1],[2,3],[4,5], [6, 7], [8, 9]]
     M = nx.blockmodel(G,partition)
In:  nx.draw_networkx(M)
Out:
```

خوشبختانه این انعطاف وجود دارد که بتوان فرایند مدل بلوکه‌ای را تغییر داد، به طوریکه مثلاً، به جای تولید پنج بلوک، دو بلوک تولید نماید (شکل ۲۲-۶).

```
In:  partition = [[0,1,2,3],[4,5, 6, 7, 8, 9]]
     M = nx.blockmodel(G,partition)
In:  nx.draw_networkx(M, node_size=800)
Out:
```

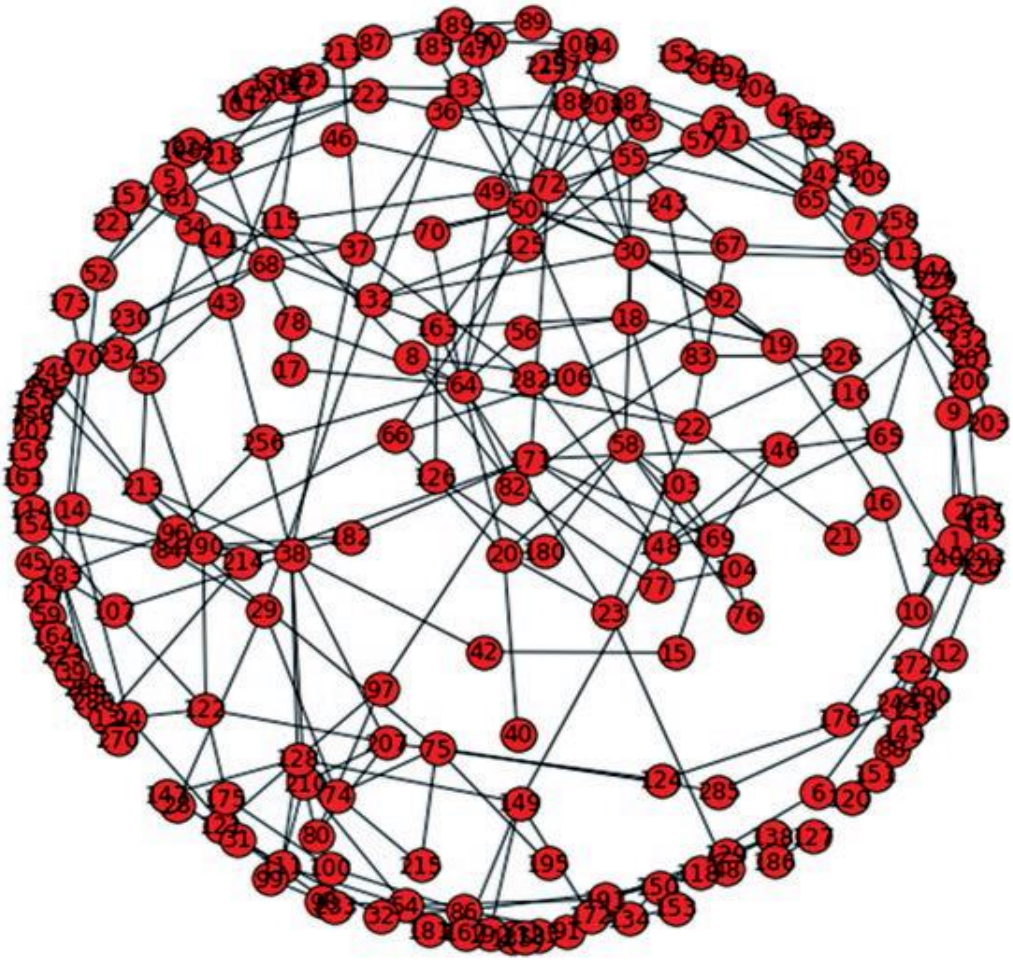


شکل ۲۲-٦ نمایش مدل بلوکه‌ای گراف مسیر (با دو بلوک)

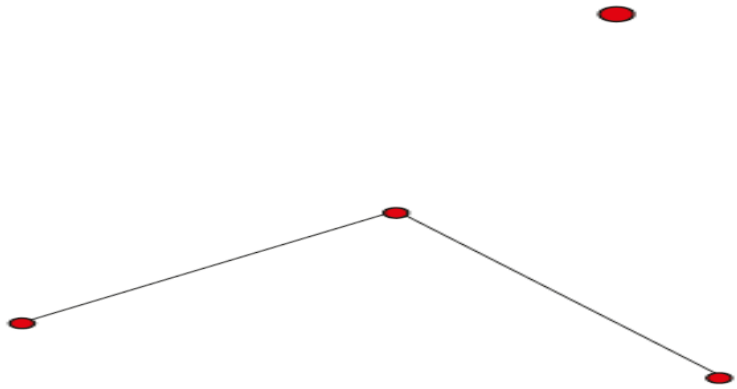
مثال۲

کدی که در ادامه آورده شده است از مجموعه دادهٔ شبکه مصرف‌کنندگان داروی CT در شهر هارتفورد، استفاده می‌کند (این داده‌ها از سایت https://networkx.github.io گرفته شده است). در ابتدا داده‌ها وارد شده و سپس ترسیم می‌شوند (شکلهای ۲۳-۶ و ۲۴-۶).

```
In:  print(nx.info(G))
Out: Name:
     Type: Graph
     Number of nodes: 212
     Number of edges: 284
     Average degree: 2.6792
In:  nx.draw_networkx(G)
Out:
```



شکل ۲۳-۶ شبکه مصرف‌کنندگان داروی CT در شهر هاتفورد



شکل ٦-٢٤ نمایش مدل بلوکه‌ای شبکه مصرف‌کنندگان داروی CT در شهر هاتفورد

```
In:
   #!/usr/bin/env python
   # encoding: utf-8
   """
   Example of creating a block model using the blockmodel
   function in
   NX. Data used is the Hartford, CT drug users network:
   @article{,
    title = {Social Networks of Drug Users in {High-Risk} Sites:
   Finding the Connections},
    volume = {6},
    shorttitle = {Social Networks of Drug Users in {High-Risk}
    Sites},
    url = {http://dx.doi.org/10.1023/A:1015457400897},
    doi = {10.1023/A:1015457400897},
    number = {2},
    journal = {{AIDS} and Behavior},
    author = {Margaret R. Weeks and Scott Clair and Stephen P.
    Borgattiand Kim Radda and Jean J. Schensul},
    month = jun,
    year = {2002},
```

```
  pages = {193--206}
}
"""
# Authors: Drew Conway <drew.conway@nyu.edu>, Aric Hagberg
<hagberg@lanl.gov>
from collections import defaultdict
import networkx as nx
import numpy
from scipy.cluster import hierarchy
from scipy.spatial import distance
import matplotlib.pyplot as plt
def create_hc(G):
      """Creates hierarchical cluster of graph G from distance
matrix"""
      path_length=nx.all_pairs_shortest_path_length(G)
      distances=numpy.zeros((len(G),len(G)))
      for u,p in path_length.items():
            for v,d in p.items():
                  distances[u ][v]=d
      # Create hierarchical cluster
      Y=distance.squareform(distances)
      Z=hierarchy.complete(Y) # Creates HC using farthest
      point
linkage
      # This partition selection is arbitrary, for illustrive
      purposes
      membership=list(hierarchy.fcluster(Z,t=1.15))
      # Create collection of lists for blockmodel
      partition=defaultdict(list)
      for n,p in zip(list(range(len(G))),membership):
            partition[p].append(n)
      return list(partition.values())
if __n ame__ == '__main__':
G=nx.read_edgelist("C:/Users/Zuhair/Desktop/
```

```
  hartford_drug.edgelist")
        # Extract largest connected component into graph H
        H = next(nx.connected_component_subgraphs(G))
        # Makes life easier to have consecutively labeled
        integer nodes
        H=nx.convert_node_labels_to_integers(H)
        # Create parititions with hierarchical clustering
        partitions=create_hc(H)
        # Build blockmodel graph
        BM=nx.blockmodel(H,partitions)
        # Draw original graph
        pos=nx.spring_layout(H,iterations=100)
        fig=plt.figure(1,figsize=(6,10))
        ax=fig.add_subplot(222)
        nx.draw(H,pos,with_labels=False,node_size=10)
        plt.xlim(0,1)
       plt.ylim(0,1)
       # Draw block model with weighted edges and nodes sized by
       Number of internal nodes
       node_size=[BM.node[x]['nnodes']*10 for x in BM.nodes()]
       edge_width=[(2*d['weight'])       for       (u,v,d)       in
BM.edges(data=True)]
       # Set positions to mean of positions of internal nodes from
       original graph
       posBM={}
       for n in BM:
           xy=numpy.array([pos[u] for u in BM.node[n]['graph']])
           posBM[n]=xy.mean(axis=0)
       ax=fig.add_subplot(666)
  nx.draw(BM,posBM,node_size=node_size,width=edge_width, =Fal
with_labels se)
      plt.xlim(0,1)
      plt.ylim(0,1)
      plt.axis('off')
```

٦-١٤-١ بهینه‌سازی پیمانه‌ای

مدولار بودن[۱](یا پیمانه‌ای بودن) معیاری مرتبط با ساختار گراف است که گروه–محور (و نه گره–محور) می‌باشد. پیمانه‌ای بودن یا پیمانگی، کمیتی است که نشاندهنده قدرت یا کارایی تقسیم یک گراف یا شبکه به پیمانه‌ها است (پیمانه‌ها را گاهی گروه، خوشه یا انجمن نیز می‌نامند). شبکه‌هایی که درجه پیمانگی بالاتری دارند، از یکسو دارای گره‌های بسیار مرتبط و متراکم در درون پیمانه‌ها بوده و از دگرسو دارای ارتباطات پراکنده و نامتراکم میان گره‌های پیمانه‌های مختلف هستند. معمولاً از پیمانگی برای بهینه‌سازی روش تشخیص ساختار اجتماعات (یا انجمن‌یابی) استفاده می‌شود. درعین حال مشخص شده است که روش پیمانگی از مشکل محدودیت دانه‌بندی و یا ضعف دقت، رنج می برد زیرا قادر نیست اجتماعات کوچک را شناسایی کند.

معیار پیمانگی به صورت رابطه ۱–۶ محاسبه می‌شود:

$$Q = \frac{|E_{in}| - |E_{in-R}|}{|E|} \qquad ۱–۶$$

در رابطه ۱–۶، $|E_{in}|$، تعداد پیوندهای مربوط به گره‌هایی است که در داخل یک اجتماع هستند. $|E_{in-R}|$، تخمینِ $|E_{in}|$، درصورت تصادفی بودن پیوندهاست.

پیمانگی که اولین بار توسط نیومن وگیروان[۲] در سال ۲۰۰۳ ارائه شد، از روشهایی است که زیاد مورد استفاده قرار می‌گیرد. این روش موجب ایجاد یک درک اولیه از خوشه‌بندی می‌شود و از دو مرحله تشکیل شده است:

- مرحله اول جدایشی یا تجزیه‌ای است که در واقع به طور تکراری آنقدر یال از شبکه حذف می‌کند که نهایتاً شبکه را تبدیل به اجتماعات می‌کند.
- مرحله دوم محاسبه مجدد[۳] است، یعنی هنگامی که یالی حذف می‌شود، مجدداً امتیازات بینابینی محاسبه می‌شود.

---

[۱] Modularity

[۲] Newman and Girvan

[۳] Recalculation

این دو مرحله آنقدر تکرار می‌شوند تا بیشترین امتیاز مدولار کلی به دست آید. نتیجه این کار، مجموعه‌ای از زیرشبکه‌ها خواهد بود به طوریکه، همانگونه که ذکر شد گره‌ها در درون زیرشبکه‌ها بسیار متراکم و منسجم بوده ولی میان زیرگرافها اصطلاحاً توخالی یا تُنُک است و انسجام کمی میان آنها وجود دارد.

٦-١٥ روش لوون

روش لوون[۱]، الگوریتمی برای یافتن انجمنها یا اجتماعات در شبکه‌های بزرگ است که توسط وینسنت بلاندل[۲]، ژان لوپ گیولم[۳]، رنولابیوت[۴] و اتین لفور[۵] توسعه داده شده است. این روش نشان می‌دهد که چگونه یک شبکه می‌تواند به‌صورت طبیعی به "گروههایی از گره‌ها" تقسیم شود به گونه‌ای که ارتباطات متراکم (و چگال) در داخل هر گروه شکل گرفته و ارتباطات غیرمتراکم (و ناچگال) بین گروهها باشد. براساس مفهوم بیشنه‌سازی پیمانگی، روش لوون یک رویکرد حریصانه[۶] اثربخش است که می‌تواند در گرافهای با مقیاس بزرگ، مثلاً با بیش از $۱۰^۹$ یال (یا یک میلیون گره) بکار رود. زمان اجرای الگوریتم $o(nlogn)$ است. مراحل اجرای الگوریتم بشرح زیر است:

۱. با "بهینه‌سازی پیمانه‌ای محلی" بر روی تمامی گره‌ها، "اجتماعات کوچکی از گره‌ها" ایجاد می‌شوند.

۲. "اجتماعات کوچک" سپس در "**یک گره**" گروه‌بندی شده (یعنی هر اجتماع کوچک یک گره درنظر گرفته می‌شود) و مجدداً مرحله یک تکرار می‌شود.

برای گراف وزندار، پیمانگی به صورت رابطه ۲-۶ تعریف می‌شود:

---

[۱] Louvain Method

[۲] Vincent Blondel

[۳] Jean-Loup Guillaume

[۴] Renaud Lambiotte

[۵] Etienne Lefebvre

[۶] Greedy Approach

$$Q = \frac{1}{2m}.\sum_{i,j}\left[a_{ij} - \frac{k_i k_j}{2m}\right].\delta(c_i c_j) \qquad ۲-۶$$

در رابطه فوق، a_{ij} وزن یال بین گره i و j است و k_i و k_j بترتیب جمع اوزان یالهای بین گره‌های i و j هستند. m جمع وزن همهٔ گره‌ها در گراف است و c_i و c_j اجتماعات مربوط به گره‌هاست. δ در اینجا دلتای کرونکر[۱] است که بررسی می‌کند آیا هر جفت نمونه، متعلق به یک اجتماع (یا انجمن) یکسان هستند یا خیر. مقدار Q بین مثبت یک و منفی یک است که به لحاظ تئوری، با بهینه کردن این مقدار می‌توان، گروهبندی بهتری از گره‌ها را ایجاد نمود. با توجه به اینکه انجام تمام تکرارهای ممکن برای گره‌های داخل گروهها عملاً امکانپذیر نیست، از الگوریتمهای هیوریستیک[۲] استفاده می‌شود.

منبع

Fortunato S *(2010)* Community detection in graphs. *Phys Rep* 486(3):75–174

[۱] Kronecker Delta

[۲] Heuristic Algorithms

فصل هفتم

تحلیل درسطح شبکه

در این فصل قصد داریم گرافها و شبکه‌ها را بعنوان یک کل و در سطح کلان مطالعه و بررسی کنیم (تحلیل جمعانه). این تحلیل با آنچه در فصلهای پیشین انجام دادیم، یعنی گرافها را در سطح گره بررسی کردیم (تحلیل فردانه) و یا در سطح گروه مطالعه کردیم (تحلیل بخشانه)، متفاوت است. لذا، در این فصل مفاهیمی مانند مولفه‌ها (سازندها) و گره‌های مجزا (ایزوله‌ها)، هسته‌ها و حاشیه (پیرامون)، چگالی (تراکم) شبکه، کوتاهترین مسیرها، دوسویگی (ارتباط متقابل)، شبکه‌های وابستگی (وابستاری)[۱] و شبکه‌های دو-حالته[۲] (دو-بخشی) و همریختی (همگونی) مورد توجه قرار می‌گیرند.

[۱] Affiliation Networks

[۲] Two-Mode Networks

۷-۱ مولفه‌ها/ایزوله‌ها

یک مولفه (یا سازند) در واقع سازه یا مجموعه‌ای از کنشگران است که در درون سازه به همدیگر متصل (همبند) بوده، اما در بیرون از سازه از سایر راسهای موجود در شبکه، منفصل (ناهمبند) هستند. هنگامیکه یک راس از دیگر راسها جدا باشد، در واقع ایزوله می‌شود.

۷-۲ هسته/حاشیه

در هسته‌های یک شبکه، اعضا به طور متراکم با یکدیگر همبند شده‌اند. اما، در حاشیهٔ شبکه، گره‌ها بیشتر با اعضای هسته، و نه با یکدیگر، پیوند دارند.

برخی از محققان، سیستم اقتصاد جهانی[۱] را در سال ۱۹۷۴ مطالعه کردند و از مفاهیم مرتبط با ساختار شبکه مانند هسته و حاشیه برای تشریح ساختار اقتصادی جهان بهره بردند. آنها نشان دادند که کشورهایی که در وسط مدل بودند (یعنی کشورهای هسته مانند ایالات متحده آمریکا، ژاپن و کشورهای غربی)، کشورهای ثروتمندی بودند. از سوی دیگر، کشورهای حاشیه‌ای مدل (مانند بولیوی، پاناما و نیکاراگوئه) کشورهای توسعه نیافته بودند. محدوده‌های دیگر، کشورهای نیمه‌حاشیه‌ای[۲] نامیده می‌شوند که نقش واسطه را برای ارتباطات میان هسته و حاشیه ایفا می‌کنند.

محققان دریافتند که کنشگران با ارتباط زیاد (یعنی کنشگران موجود در هستهٔ شبکه[۳]) و کنشگران با تنها چند ارتباط (یعنی کنشگران موجود در حاشیهٔ شبکه[۴]) الگوهای رفتاری محدود و قابل پیش‌بینی دارند. ولی، کنشگرانی که ارتباطات متوسط دارند (نه کم و نه زیاد)، رفتار قابل پیش‌بینی ندارند. رفتار این گروه عمدتاً به کسانی که با آنها ارتباط دارند، بستگی دارد.

---

[۱] World Economic System

[۲] Semi-Periphery

[۳] Network Core

[۴] Network Periphery

۷-۳ چگالی

چگالی (یا تراکم) به‌صورت تعداد یالهای موجود در شبکه تقسیم بر حداکثر تعداد یالهای ممکن بین گره‌ها در شبکه تعریف می‌شود. مقادیر چگالی بین ۰ و ۱ است. این معیار برای کاوش برخی از پویاییهای شبکه[۱] مانند سرعت انتشار اطلاعات بین گره‌ها و نیز سطح سرمایه اجتماعی[۲] و یا سطح محدودیتهای اجتماعی[۳] که گره‌های شبکه دارند، مفید است.

مثال

```
In:  import networkx as nx
     %matplotlib inline
     import matplotlib.pyplot as plt
```

در ابتدا شبکه خود را با استفاده از گراف *Karate Club* موجود در کتابخانه *NetworkX*، می‌سازیم و سپس چگالی آنرا محاسبه می‌کنیم:

```
In:  g = nx.karate_club_graph()
     nx.density(g)
Out: 0.13903743315508021
```

اجازه دهید، یک شبکه فردانه (فردمحور) بسازیم و مجدداً چگالی را محاسبه کنیم:

```
In:  ego = nx.ego_graph(g, 8)
     nx.density(ego)
Out: 0.6666666666666666
```

همانطور که تصور می‌شد، شبکه *Karate Club* در کل (و در مقایسه با شبکه هشت گره‌ای که خیلی متراکم و چگال است) بسیار متراکم نیست.

---

[۱] Network Dynamics

[۲] Social Capital

[۳] Social Constraints

اکنون، بیایید چگالی شبکه *hero* را محاسبه کنیم:

```
In:  nx.density(graph)
Out: 0.00810031232553549
```

۷-٤ کوتاهترین مسیر

فرض کنید $L(i,j)$ طول کوتاهترین مسیر(ها) بین گره‌های i و j باشد. معمولاً، محاسبه کوتاهترین مسیرها با الگوریتم دایکسترا انجام می‌شود. طولانی‌ترین فاصله در یک گراف "قطر" نامیده می‌شود که طولانی‌ترین مسیر کوتاه بین تمامی زوج گره‌های موجود در شبکه (یا حداکثر برون‌مرکزی[۱] در شبکه) را نشان می‌دهد. هدف از معیار قطر شبکه، نشان دادن گستردگی[۲] شبکه است، به این معنی که فاصله دورترین گره‌ها در شبکه از یکدیگر چقدر است. "شعاع" کمترین برون‌مرکزی در شبکه است. در شبکه‌های اجتماعی، شبکه‌های کم‌تراکم در مقایسه با شبکه‌های متراکم، قطر بزرگتری دارند. علاوه بر این، میانگین کوتاهترین مسیرِ میان تمامی زوج گره‌ها معیار جالبی است، زیرا می‌تواند مشخص کند جامعه یا جهانی که با این شبکه نمایش داده شده تا چه حد بزرگ است؛ در همین راستا یک طول کوتاه برای این میانگین نشان می‌دهد که شبکه در کل، بخوبی همبند و متصل است.

مثال

بیایید بر روی گراف *Karate Club* برخی محاسبات مربوط به شعاع و قطر را انجام دهیم:

```
In:  nx.shortest_path(g, 0, 8)
Out: [0, 8]
In:  nx.shortest_path_length(g, 0, 8)
Out: 1
In:  nx.average_shortest_path_length(g)
Out: 2.408199643493761
```

---

۱ Eccentricity Maximum

۲ Extensiveness

تابع *nx.all_pairs_shortest_path()* به ما اجازه می‌دهد که فهرست کوتاهترین مسیرها برای هر گره در گراف را ببینیم:

```
In:  print(nx.all_pairs_shortest_path(g))
Out: {0: {0: [0], 1: [0, 1], 2: [0, 2], 3: [0, 3], 4: [0, 4], 5: [0, 5],
     6: [0, 6], 7: [0, 7], 8: [0, 8], 9: [0, 2, 9], 10: [0, 10], 11: [0,
     11], 12: [0, 12], 13: [0, 13], 14: [0, 2, 32, 14], 15: [0, 2, 32,
     15], 16: [0, 5, 16], 17: [0, 17], 18: [0, 2, 32, 18], 19: [0, 19],
     20: [0, 2, 32, 20], 21: [0, 21], 22: [0, 2, 32, 22], 23: [0, 2, 32,
     23], 24: [0, 31, 24], 25: [0, 31, 25], 26: [0, 8, 33, 26], 27: [0,
     2, 27], 28: [0, 2, 28], 29: [0, 2, 32, 29],…}
```

همچنین می‌توانیم کوتاهترین مسیرها را بین یک گره خاص و تمام گره‌ها در گراف پیدا کنیم:

```
In:  print(nx.all_pairs_shortest_path(g)[1])
Out: {0: [1, 0], 1: [1], 2: [1, 2], 3: [1, 3], 4: [1, 0, 4], 5: [1, 0,
     5], 6: [1, 0, 6], 7: [1, 7], 8: [1, 0, 8], 9: [1, 2, 9], 10: [1, 0,
     10], 11: [1, 0, 11], 12: [1, 0, 12], 13: [1, 13], 14: [1, 2, 32,
     14], 15: [1, 2, 32, 15], 16: [1, 0, 5, 16], 17: [1, 17], 18: [1, 2,
     32, 18], 19: [1, 19], 20: [1, 2, 32, 20], 21: [1, 21], 22: [1, 2,
     32, 22], 23: [1, 2, 32, 23], 24: [1, 2, 27, 24], 25: [1, 0, 31,
     25], 26: [1, 13, 33, 26], 27: [1, 2, 27], 28: [1, 2, 28], 29: [1,
     2, 32, 29], 30: [1, 30], 31: [1, 0, 31], 32: [1, 2, 32], 33: [1,
     13, 33]}
```

محاسبه قطر گراف در شبکه‌های اجتماعی بسیار مهم است، زیرا گره‌ها، پیامهای ارتباطی[۱] را ارسال و دریافت می‌کنند و از این رو محاسبه قطر به درک نحوه انتقال یا انتشار اطلاعات کمک می‌کند. در گرافهای نرمال[۲]، تمام یالها فرصت یکسان برای دریافت اطلاعات دارند، زیرا فاصله برابری با منبع اطلاعات دارند. از سوی دیگر، در گرافهای وزندار که یالها می‌توانند فرکانس ارتباطات فردی (و موارد

[۱] Communication Messages

[۲] Normalized Graphs

دیگری) را نشان دهند، اوزان یال[۱] می‌تواند اطلاعاتی از قبیل سرعت جریان اطلاعات از یک گره به گره دیگر را نشان بدهد.

بیایید ابتدا قطر گراف را محاسبه کنیم، که عملاً همان حداکثر برون‌مرکزی در شبکه است (برون‌مرکزی یا گریز از مرکز یک گره v حداکثر فاصله از v تا تمام گره‌های دیگر در گراف g است):

```
In:  nx.diameter(g)
Out: 5
In:  print(nx.eccentricity(g))
Out: {0: 3, 1: 3, 2: 3, 3: 3, 4: 4, 5: 4, 6: 4, 7: 4, 8: 3, 9: 4, 10: 4,
     11: 4, 12: 4, 13: 3, 14: 5, 15: 5, 16: 5, 17: 4, 18: 5, 19: 3, 20:
     5, 21: 4, 22: 5, 23: 5, 24: 4, 25: 4, 26: 5, 27: 4, 28: 4, 29: 5,
     30: 4, 31: 3, 32: 4, 33: 4}
```

روش *nx.eccentricity ()* از کتابخانه *Packet NetworkX* یک دیکشنری از مقادیر گریز از مرکز ایجاد می‌کند که گره‌ها، کلید دیکشنری هستند.

راه دیگر برای محاسبه قطر گراف بصورت زیر است:

```
In:  eccentricity = nx.eccentricity(g)
     print(max(eccentricity.values()))
Out: 5
In:  radius = nx.radius(g)
     print(radius)
Out: 3
In:  print(min(eccentricity.values()))
Out: 3
```

[۱] Edge Weights

۷-۵ دوسویگی

روابط متقابل و یا دوسویگیِ یک گراف جهتدار، نسبت تعداد یالهای (روابط) دوجهته، به تعداد کل روابط در گراف (یا شبکه‌اجتماعی) است.

۱-۷

$$R = \left|(u,v) \in G \middle| (v,u) \in G\right| / \left|(u,v) \in G\right|$$

دوسویگیِ یک گره تنها همچون u نیز به طور مشابه تعریف می‌شود. یعنی، نسبت تعداد یالهای دوجهته آن به تعداد کل یالهای متصل به گره u است. این بدان معناست که دوسویگی برای گره‌های ایزوله تعریف نمی‌شود. در چنین مواردی، تابع محاسبه دوسویگی، درخروجی خود، چیزی را نشان نمی‌دهد.
این معیار تنها می‌تواند برای گرافهای جهتدار (مانند توییتر) محاسبه شود که در آن شما می‌توانید دیگران را دنبال کنید، هرچند ممکن است دیگران لزوماً شما را دنبال نکنند. از سوی دیگر، شبکه دوستی فیسبوک همیشه غیرجهتدار است.
محاسبه دوسویگی گراف *hero*:

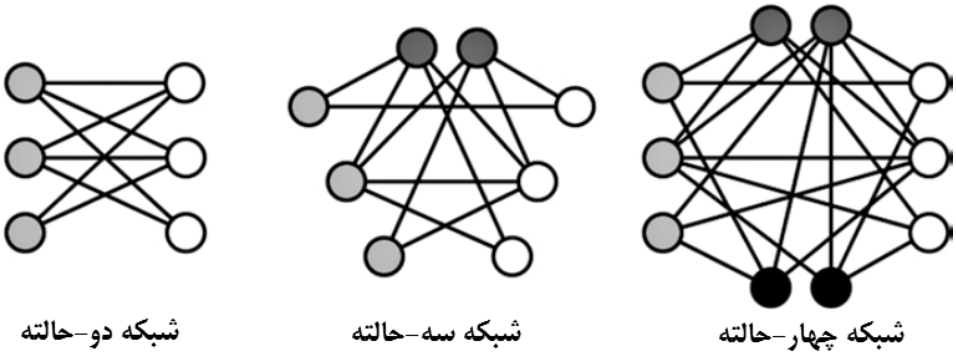
```
In:  und_graph = graph.to_undirected()
     reciprocity =
     float(nx.number_of_edges(und_graph)/nx.number_of_edges(graph))
     print(reciprocity)
Out: 1.0
```

باتوجه به اینکه شبکه *hero*، غیرجهتدار است، نتایج دوسویگی ۱,۰ می‌شود.

۷-۶ شبکه‌های وابستاری

شبکه‌های وابستگی یا وابستاری، شبکه‌هایی هستند که حداقل دو مجموعه مختلف از گره‌ها (تحت عنوان کنشگران و رویدادها) را دارند به‌گونه‌ای‌که یالها (که همچنین وابستگی نامیده می‌شود) فقط گره‌های متعلق به مجموعه‌های ناهمنام (و نه گره‌های مجموعه خودی) را بهم متصل می‌کنند. این نوع

شبکه‌ها، شبکه‌های وابستگی (سازمانی) یا وابستاری نامیده می‌شوند تا چند عضویتی بودن گره[۱] یا چند مشارکتی بودن[۲] آن را بیان کنند (شکل ۱-۷).



شکل ۱-۷ انواع شبکه‌های وابستاری (یا وابستگی سازمانی)

افراد در این نوع شبکه‌ها می‌توانند عضو بیش از یک گروه باشند، که معمولاً بعنوان عضو مشترک (یا هم-عضو)[۳] شناخته می‌شوند. وجود هم-عضو در یک گروه یا یک رویداد می‌تواند شاخصی برای پیوند اجتماعی[۴] باشد. در واقع شبکه‌های هم-شرکتی[۵] به نوعی روابط اجتماعی را توسعه می‌دهند که آنهم به نوبه خود باعث تسهیل ترویج و انتشار نوآوری، اطلاعات، اخبار و امثال آن در شبکه می‌شود.

عضویت مشترک (یا هم-عضوی) در یک گروه یا یک رویداد، داده‌های وابستگی یا وابستاری[۶] تولید می‌کند که از آن می‌توان برای نشان دادن ارتباط بین دو موجودیت یا نهاد از دو مجموعه مختلف استفاده شود. در این راستا از یک مقدار باینری بهره می‌گیرند. مقدار باینری (۰ یا ۱) به معنای یک "رابطه باینری"[۷] است.

---

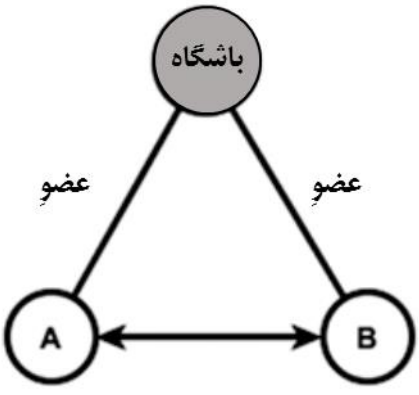
[۱] Node Multi-Membership
[۲] Multi-Participation
[۳] Co-Membership
[۴] Social Tie
[۵] Co-Participation Networks
[۶] Affiliation Data
[۷] Binary Relationship

بعنوان مثال، یک فرد ممکن است در برخی از انجمنها، عضویت داشته یا در برخی از رویدادها، مشارکت داشته یا در هیچکدام از آنها عضویت و یا مشارکت نداشته باشد. جدول ۱-۷، نمونه‌ای از داده‌های وابستاری را نمایش می‌دهد؛ که به گونه‌ای نشاندهنده روابط میان کنشگرانی است که در سه رویداد حضور داشته‌اند. اگر کنشگری در یک رویداد حضور داشته باشد، عدد یک را وارد می‌کنیم؛ در غیر اینصورت مقدار ۰ را وارد می‌کنیم. هر چه تعداد دفعاتی که افراد در یک رویداد شرکت می‌کنند (که به آن میزان مشارکت نیز می‌گویند) بیشتر باشد، آنگاه احتمال اینکه آنان با یکدیگر ارتباط برقرار کرده و سپس روابط را توسعه دهند، بیشتر می‌شود (جدول ۱-۷).

جدول ۱-۷ رابطه بین کنشگر-رویداد

	رویداد ۱	رویداد ۲	رویداد ۳
کنشگر ۱	۱	۰	۰
کنشگر ۲	۱	۰	۱

در جدول بالا، کنشگر ۱ و کنشگر ۲ هر دو شرکت‌کننده در رویداد ۱ هستند. این امر باعث می‌شود که احتمالاً آنها در طول زمان، روابط خود را توسعه دهند. این احتمال قابل تزاید است اگر تعداد رویدادهایی که آنها در آن شرکت می‌کنند (یعنی میزان مشارکت) افزایش یابد (شکل ۲-۷).



شکل ۲-۷ شبکه ساده شده افراد-به-باشگاه

در شبکهٔ شکل ۲-۷، که عملاً نسخهٔ ساده شدهٔ شبکهٔ "افراد-به-باشگاه" می‌باشد، نفرات A و B عضو یک باشگاه هستند. اساساً، این گراف، یک "سه‌وَند باز" با یک حفره ساختاری است. از آنجا که A و B عضو یک باشگاه هستند، می‌توانیم با کمی مسامحه چنین استنتاج کنیم که آنها یکدیگر را می‌شناسند و نتیجتاً این گراف، یک "سه‌وَند بسته" می‌شود. این استنتاج می‌توانست کاملاً متقن و مستدل شود، اگر قادر بودیم اطلاع حاصل کنیم که A و B همزمان عضو یک باشگاه بوده، در ساعت کاری یکسانی حضور داشته یا فعالیت مشابهی را انجام می‌دادند[۱].

۷-۷ شبکه‌های دو-حالته

گرافهای وابستاری را گاهی اوقات شبکه‌های دو-حالته[۲]، دو-بخشی[۳] یا دو-مُد[۴] می‌نامند. اینگونه شبکه‌ها توسط گرافهای دو-بخشی[۵]، یعنی $B=(V1,V2,E)$، نشان داده می‌شوند. $V1$ و $V2$ دو مجموعه مختلف از گره‌ها هستند و E مجموعه‌ای از یالها است که فقط گره‌های از مجموعه‌های ناهمنام یا مخالف[۶] را بهم وصل می‌کنند. بعنوان مثال، یک مجموعه داده می‌تواند شامل اطلاعات مربوط به ارتباط کارمندان با مدیران یا ارتباط مدیران با هیئت مدیره باشد. علاوه بر سازمانها، این مطلب در مورد باندها[۷]، احزاب سیاسی، باشگاههای اجتماعی و بسیاری موارد دیگر نیز صادق است.

---

[۱] توجه کنید که در شبکه‌های وابستاری (یا وابستگی سازمانی) از واژه‌های هم-نهادی همچون: هم-باشگاهی، هم-عضوی، هم-قطاری، هم-زمانی، هم-فعالیتی، هم-مشارکتی، هم-برنامه‌ای، هم-رویدادی، هم-ورزشی و امثال آن استفاده می‌شود که دلالت بر انجام کار مشابه توسط دو عضو داشته اما نباید فوراً تعبیر به شناخت متقابل آن دو نفر شود.

[۲] Two-Mode

[۳] Bipartite

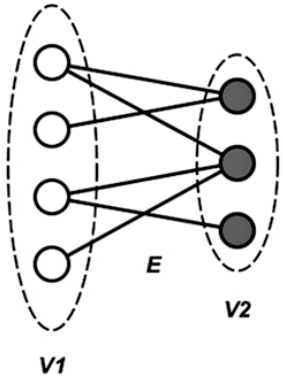
[۴] Bimodal Network

[۵] Bipartite Graphs

[۶] Opposite Sets

[۷] Gangs

به بیان دیگر، می‌توان گفت شبکه‌های دو–حالته، شبکه‌هایی هستند که تنها از دو مجموعه گره تشکیل شده‌اند. از لحاظ ساختاری، این شبکه‌ها با شبکه‌های یک–حالته[۱] که در آن تمام گره‌ها به یک مجموعه تعلق دارند و هر گره می‌تواند با گره‌های دیگر مرتبط باشد، متفاوت هستند (شکل ۳–۷).



شکل ۳–۷ شبکه دو–حالته

در نمودار شکل ۳–۷، ما دو مجموعه گره، یعنی مجموعه $V1$ (دارای چهار گره) و مجموعه گره $V2$ (دارای سه گره) را داریم. این دو مجموعه گره از طریق مجموعه‌ای از یالها، E (شش یال)، بهم متصل می‌شوند. یالهای مجموعه E فقط گره‌های متعلق به مجموعه‌های مخالف (یعنی ناهمنام) را بهم وصل می‌کنند.

برخی معیارهای شبکه همچون چگال بودن و کامل بودن (به عبارت دیگر تراکم و تمامیت[۲]) برای شبکه‌های دو–حالته باید به روشی متمایز از آن روشی که برای شبکه‌های یک–حالته اعمال می‌شود، محاسبه شوند. فلذا بعنوان مثال، معیار کامل بودن، که بعنوان بیشینه تعداد خط (یا یال) در یک شبکه تعریف می‌شود، مقدار بسیار بیشتری را در شبکه‌های تک–حالته، در مقایسه با شبکه‌های دو–حالته، داراست. زیرا هر گره در شبکه‌های یک–حالته می‌تواند به برخی یا همه گره‌های دیگر موجود در شبکه متصل شود. در مقابل، گره‌های یک مجموعه در شبکه‌های دو–حالته، فقط می‌توانند به گره‌های مجموعه دیگر متصل شوند. همچنین، به دلیل مشابهی، چگالی یا تراکم، که بعنوان تعداد یالهای موجود

---

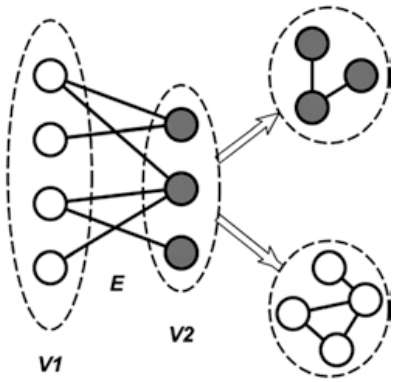
[۱] One-Mode Networks

[۲] Completeness

در یک شبکه تقسیم بر حداکثر تعداد ممکن یالها در همان شبکه محاسبه می‌شود، مقدار بیشتری را برای شبکه‌های تک–حالته نسبت به شبکه‌های دو–حالته داراست. از این رو، نمی‌توان تکنیکهایی را که برای تجزیه و تحلیل شبکه‌های یک–حالته استفاده می‌شود، برای شبکه‌های دو–حالته اعمال کرد. پس چه می‌توانیم بکنیم؟ یک راه حل خوب، روشی است که معنای شبکه‌های دو–حالته را دگرگون نکرده و یا تغییر ندهد اما تکنیک‌پذیر باشد.

خوشبختانه تغییر شبکه دو–حالته به شبکه تک–حالته امکانپذیر است. این کار با تقسیم گره‌ها به دو مجموعهٔ مجزا انجام می‌شود، بگونه‌ای که یالها در این روش، گره‌های متعلق به دو مجموعه مختلف را بهم متصل نمی‌کنند. در نتیجه "دو شبکهٔ یک–حالته" ایجاد می‌شود. در این صورت تکنیکهای تجزیه و تحلیل استاندارد[۱] می‌تواند برای تجزیه و تحلیل شبکه جدید استفاده شود (شکل ۴–۷).

در شکل ۴–۷، دو مجموعه گره *V1* و *V2*، از طریق مجموعه‌ای از یالها، *E*، بهم متصل شده‌اند. تبدیل این گراف به دو گراف جداگانه، یک جفت شبکهٔ یک–حالته ایجاد می‌کند. یالها اکنون بین گره‌های موجود در یک مجموعه (به جای گره‌های موجود در مجموعه‌های مختلف) ارتباط برقرار می‌کنند. ایجاد یالهای جدید بر اساس هم–عضویتی (عضویت توام) و یا هم–مشارکتی (مشارکت توام) هر گره در شبکه اصلی انجام می‌شود.



شکل ٤–۷ تبدیل شبکه دو–حالته به دو شبکه یک–حالته

[۱] Standard Analysis Techniques

شبکهٔ یک–حالتهٔ منتجه، معمولاً متراکم است و شامل محافل بسیاری است. روشهای تحلیل محافل یا زیرشبکه‌های کامل (مانند مدلهای بلوکه‌ای، خوشه‌بندی‌سلسله‌مراتبی، جزایر[۱]، k–هسته‌ای و m–قطاع) را می‌توان در اینجا برای شناسایی "گروههای منسجم" استفاده کرد. این بدان دلیل است که شبکه‌های منتجه، عملاً شبکه‌های شباهت[۲] یا شبکه‌های همبستگی[۳] هستند.

در بسیاری از موارد، شبکه مشتق‌شده[۴] یک شبکه ساده نیست، یعنی ممکن است حاوی حلقه‌هایی باشد. خطوط چندگانه[۵] (یعنی وجود بیش از یک یال یا کمان برای اتصال میان هر جفت گره) را می‌توان با یک خط واحد جایگزین کرد تا یک شبکه ارزش‌دار (یا ارزش‌گذاری شده)[۶] بدست آید. شبکه ارزش‌دار، شبکه‌ای است که در آن، هر خط، یک مقدار یا ارزش دارد. ارزشی که روی خط واحد جدید وجود دارد قدرت ارتباط میان یک زوج گره را نشان می‌دهد و از طریق شمارش تعداد خطوط قبلی محاسبه می‌شود. از این مقدار یا ارزش همچنین بعنوان "چندگانگی یا کثرت خطوط"[۷] یاد می‌کنند. هرچه ارزش خطوط چندگانه (که دو گره را بهم چفت و بست می‌کند) بیشتر باشد، انسجام و قدرت پیوند آن دو گره بیشتر است.

بدست آوردن یک شبکه یک–حالته از یک شبکه دو–حالته به کشف روابط اجتماعیِ مابین کنشگران هر یک از آن دو مجموعه، کمک می‌کند؛ موضوعی که تا پیش از این هرگز قادر به کشف آن نبودیم[۸].

۷-۸ همریختی

عضویت مشترک در یک انجمن اجتماعی، اغلب بدان معناست که علائق مشابهی بین اعضا وجود دارد. بعنوان مثال، افرادی که عضو یک باشگاه شطرنج هستند احتمالاً دارای حرفه، علاقه یا حتی وضعیت

---

[۱] Islands

[۲] Networks of Similarities

[۳] Networks of Correlations

[۴] Derived Network

[۵] Multiple Lines

[۶] Valued Network

[۷] Line Multiplicity

[۸] یک مثال واقعی در این خصوص، در فصل نهم تشریح شده است.

اجتماعی مشابه هستند. در واقع این تعداد و شدت موضوعات مشترک بین افراد است که موجب استنتاج در مورد درجه شباهت بین آنان می‌شود.

یکی از مزایای بدست آوردن شبکه‌های یک–حالته از یک شبکه دو–حالته، شناسایی الگوهای تعاملی بین افراد مشابه است. این پدیده بعنوان همریختی[۱] شناخته می‌شود.

این پدیده (که توسط لازارسفلد و مرتون[۲] در سال ۱۹۵۴ شناسایی و تبیین شد)، فراتر از الگوها و معیارهای مرسوم تحلیل شبکه‌ها همانند درجه، فاصله، تراکم و معیارهای دیگر است. این معیارهای مرسوم فقط مربوط به ساختار شبکه می‌شوند. به طور دقیق، در شبکه‌های اجتماعی، گره‌ها (مثلا مردم) اغلب تمایل بیشتری دارند که با گره‌های مشابه خودشان (در مقایسه با گره‌های نامشابه) ارتباط داشته باشند. این الگوی اختلاط (یا جذابیت[۳]) براساس ویژگیهای شخصی مانند سن، حرفه، جنسیت، نوع سرگرمی، دیدگاه سیاسی و موارد بسیار دیگر است.

همگونی می‌تواند رفتار و رفاه را به طرق مختلفی تحت تأثیر قرار دهد. ایده‌ها، نگرشها و ارتباطات اجتماعی مردم می‌تواند تحت تأثیر عضویت در برخی از انجمنهای اجتماعی، فرهنگی، تفریحی و یا علمی قرار گیرد. بعنوان مثال، تصمیمات کارگران در مورد اینکه مثلاً آیا از شمول قوانین نیروی کار خارج شوند یا خیر، تحت تاثیر همگونی قرار می‌گیرد. در واقع اتخاذ چنین تصمیماتی عملاً به میزان زیادی به طرز فکر سایر دوستان و همکاران آن کارگران بستگی دارند.

به مثال قبلی یعنی شبکه افراد–به–باشگاه بازگردید، یک نسخه توسعه یافته از آن شبکه، یک شبکه دو–حالته (یا شبکه وابستاری) است که افراد را به باشگاههایی که عضو آنها هستند، متصل می‌کند. با استخراج شبکه یک-حالته از شبکه اصلی، یک شبکه فرد–به–فرد[۴] و یک شبکه باشگاه–به–باشگاه[۵] ایجاد می‌شود. شبکه اول برای ما جالبتر است؛ زیرا نشان می‌دهد که چه‌کسی احتمالاً به چه کسی متصل می‌شود. از روی این شبکهٔ فرد–به–فرد و براساس ارتباطات موجود در آن، می‌توان فهرستی از

---

[۱] کلمه هموفیلی، که ما از واژه‌های همریختی، همگونی و یک‌ریختی بعنوان معادلش استفاده کرده‌ایم، یک کلمه یونانی به معنی عشق و علاقه میان افراد مشابه است.

[۲] Lazarsfeld and Merton

[۳] Assortativity

[۴] Person-to-Person Network

[۵] Club-to-Club Network

دوستان پیشنهادی[۱] را برای هر فرد ایجاد کرده و به او معرفی کرد. همچنین می‌توان فهرستی از ترجیحات پیشنهادی[۲] نیز ایجاد کرد تا به افراد گفته شود دوست و همتای آنان چه چیزی را ترجیح می‌دهد.

مثال ۱

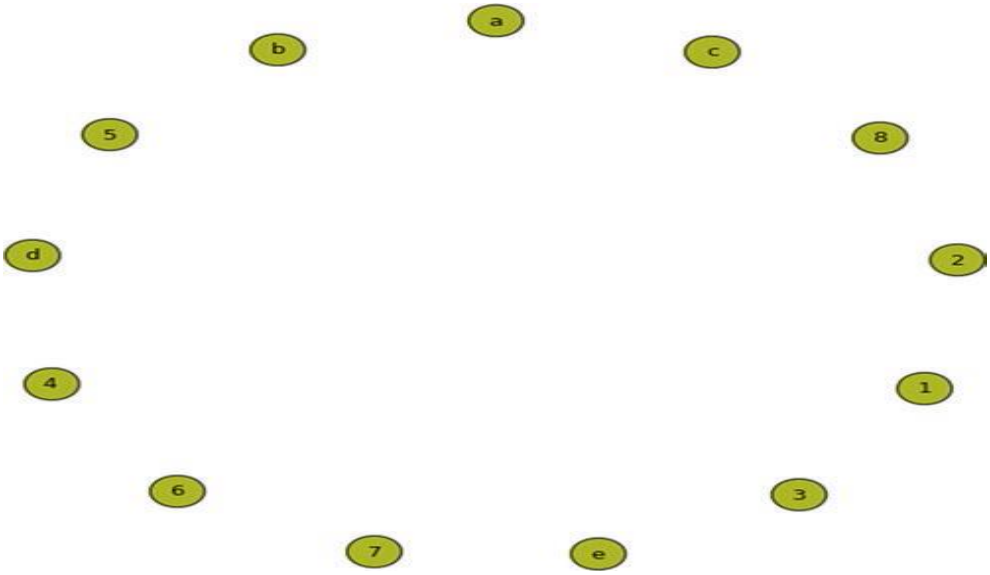
در این مثال، قصد داریم یک گراف بسیار ساده دو–حالته (یا دو–بخشی) بسازیم. باید *NetworkX* و *bipartite* را از *networkx.algorithms* وارد کنیم. برای *bipartite* نیاز است نسخه ۱٫۵ یا جدیدتر *NetworkX* نصب شود.

با یک گراف خالی بدون جهت شروع می‌کنیم (گراف جهتدار نیز می‌تواند در نظر گرفته شود). هنگام اضافه‌کردن یالها به گره‌ها، باید بررسی کنیم که کدام گره متعلق به کدام مجموعه است و اطمینان حاصل کنیم که یالهای بین گره‌های متعلق به یک مجموعه به یکدیگر متصل نمی‌شوند. معمولاً از یک مشخصه گره به نام "دو–بخشی" با مقادیر ۰ یا ۱ برای شناسایی هر مجموعه گره، استفاده می‌کنند (شکل ۵–۷).

```
In:  import networkx as nx
     from networkx.algorithms import bipartite
In:  B = nx.Graph()
     B.add_nodes_from([1,2,3,4,5,6,7,8], bipartite=0)
     B.add_nodes_from(['a','b','c','d','e'], bipartite=1)
In:  nx.draw_networkx(B, node_color='y', node_size=800)
Out:
```

[۱] Suggested Friends

[۲] Suggested Preferences



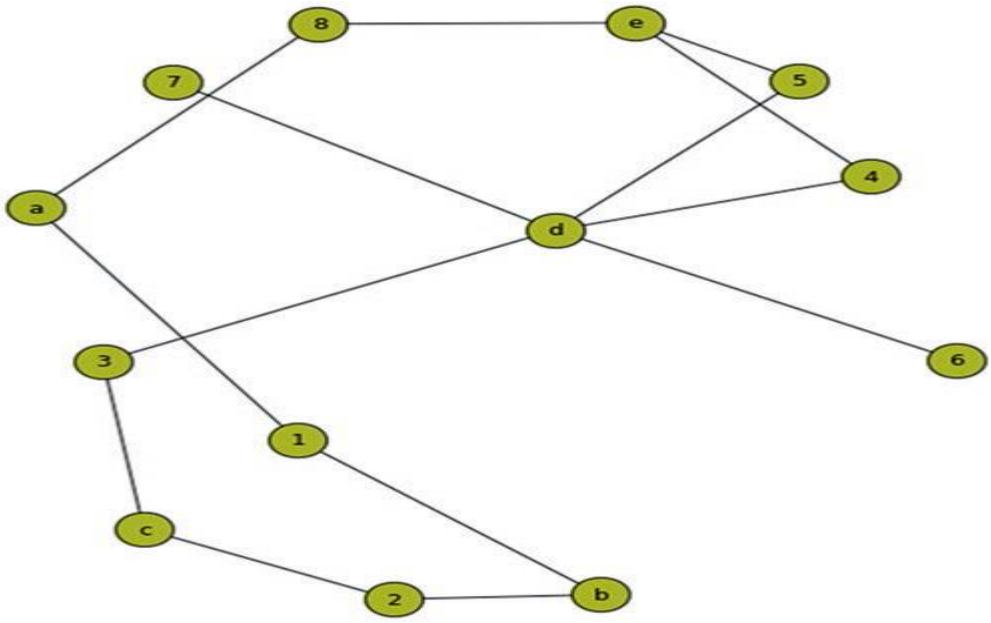
شکل ۵-۷ گراف دو-بخشی بدون یال بین گره‌ها

اجازه بدهید برخی از یالهای بین گره‌ها را از دو مجموعه مختلف (ناهمنام) اضافه کنیم:

```
In:  B.add_edges_from([(1,'a'), (1,'b'), (2,'b'), (2,'c'), (3,'c'),
     (3,'d'), (4, 'd'), (4, 'e'),(5, 'd'), (5, 'e'), (6, 'd'), (7,
     'd'),(8, 'e'), (8, 'a')])
```

حال می‌توانیم گراف را رسم کنیم (شکل ۶-۷):

```
In:  nx.draw_networkx(B, node_color='y', node_size=800)
Out:
```



شکل ٦-٧ گراف دو-بخشی به‌همراه یالهای بین گره‌ها

بیایید بررسی کنیم که آیا این گراف، یک گراف دو-بخشی (یا به بیان دیگر: یک شبکه دو-حالته) است یا خیر:

```
In:  bipartite.is_bipartite(B)
Out: True
```

حال اجازه دهید بررسی کنیم آیا این گراف، همبند (متصل) است یا خیر:

```
In:  nx.is_connected(B)
Out: True
```

حال بیایید مجموعه گره‌های این گراف دو-بخشی را بررسی کنیم:

```
In:  bipartite.sets(B)
Out: ({'a', 'b', 'c', 'd', 'e'}, {1, 2, 3, 4, 5, 6, 7, 8})
In:  bottom_nodes, top_nodes = bipartite.sets(B)
```

```
     list(bottom_nodes)
Out: ['a', 'c', 'b', 'e', 'd']
In:  list(top_nodes)
Out: [1, 2, 3, 4, 5, 6, 7, 8]
```

روش دیگری برای بدست آوردن دو مجموعه گره وجود دارد که براساس ویژگی گره دو–بخشی است.

```
In:  top_nodes = set(n for n,d in B.nodes(data=True) if
     d['bipartite']==0)
     bottom_nodes = set(B) - top_nodes
In:  list(top_nodes)
Out: [1, 2, 3, 4, 5, 6, 7, 8]
In:  list(bottom_nodes)
Out: ['a', 'c', 'b', 'e', 'd']
```

اجازه دهید چگالی گراف را بررسی کنیم:

```
In:  bipartite.density(B, nodes)
Out: 0.3333333333333333
```

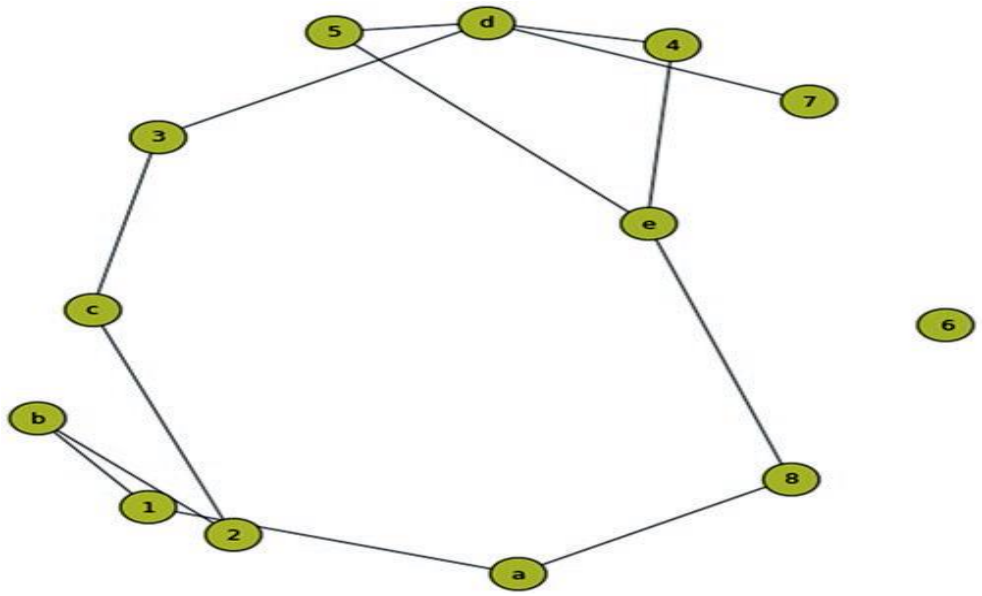
روش محاسبه چگالی گردشده[۱](دو عدد اعشار بعد از ممیز) گراف بصورت زیر است:

```
In:  round(bipartite.density(B, bottom_nodes),2)
Out: 0.35
```

بگذارید ببینیم چه اتفاقی می‌افتد اگر یکی از یالها را حذف کنیم (شکل ۷–۷):

```
In:  B.remove_edge(6,'d')
     nx.is_connected(B)
Out: False
In:  nx.draw_networkx(B, node_color='y', node_size=800)
Out:
```

[۱] Rounded Density



شکل ۷-۷ گراف دو-بخشی با یک یال حذف‌شده

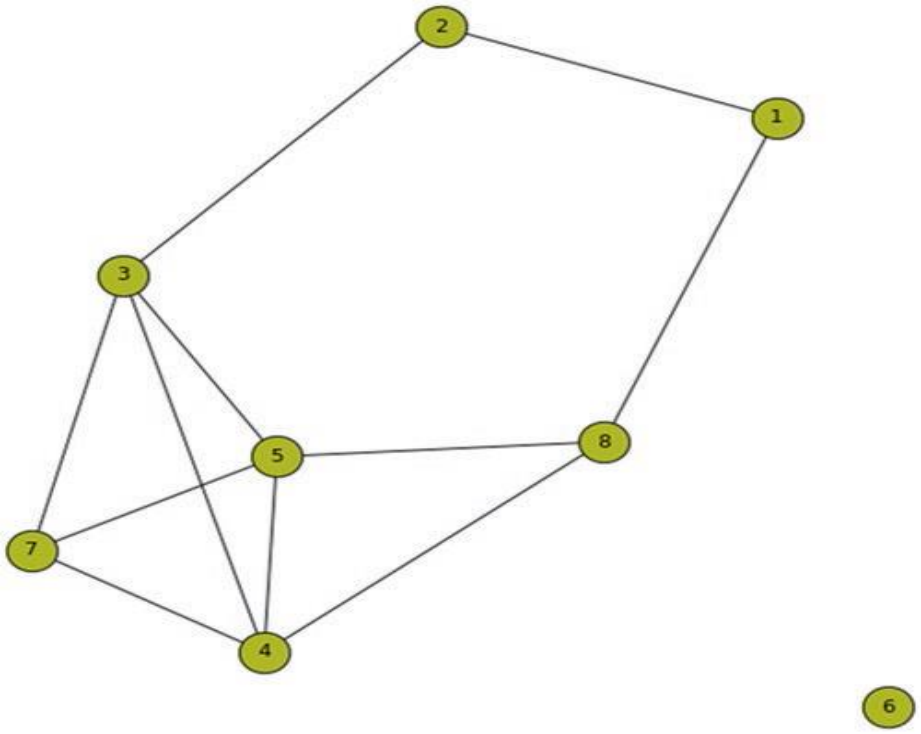
مثال ۲

در این مثال، ما کار خود را با مثال قبلی و با تبدیل گراف دو-بخشی به دو شبکهٔ یک-حالته، ادامه خواهیم داد. تابع *bipartite.projected_ graph ()* تصویر B را بر روی یکی از مجموعه گره‌های خود برمی‌گرداند:

```
In:  G = bipartite.projected_graph(B, top_nodes)
```

شبکهٔ تک-حالته جدید G نام دارد. این شبکه شامل اولین مجموعه از گره‌ها به نام *top_nodes* می‌باشد. بیایید برخی از جزئیات این گراف را بررسی کنیم (شکل ۸-۷):

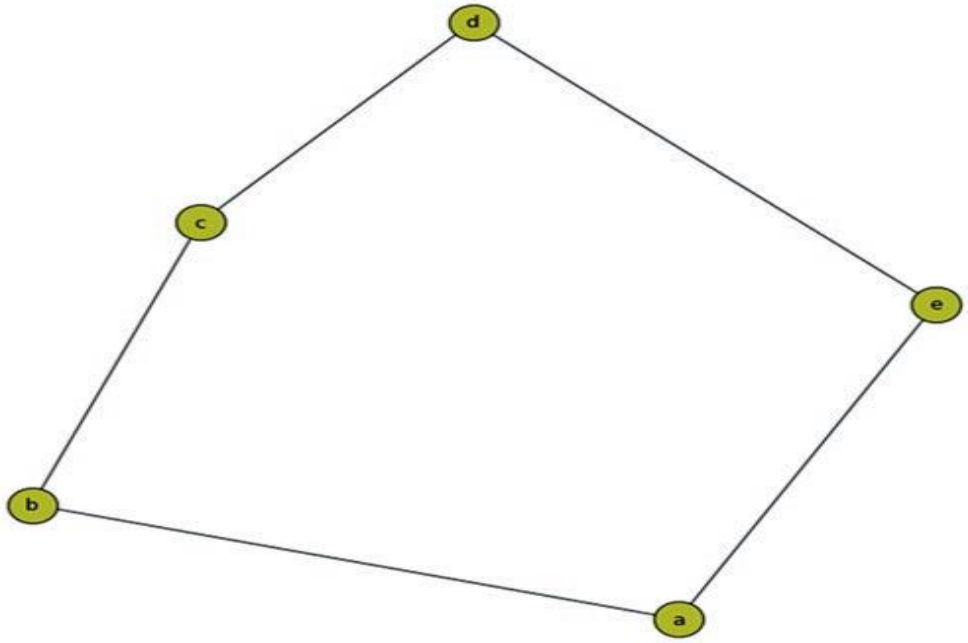
```
In:  G.edges()
Out: [(1, 8), (1, 2), (2, 3), (3, 4), (3, 5), (3, 7), (4, 8), (4, 5),
     (4, 7), (5, 8), (5, 7)]
In:  G.nodes()
Out: [1, 2, 3, 4, 5, 6, 7, 8]
In:  nx.draw_networkx(G, node_color='y', node_size=800)
Out:
```



شکل ۸-۷ شبکهٔ یک-حالته از اولین مجموعه گره‌ها

دومین شبکهٔ یک-حالته شامل گره‌هایی از مجموعه دیگر (*bottom nodes*) است (شکل ۹-۷).

```
In:  G = bipartite.projected_graph(B, bottom_nodes)
     G.nodes()
Out: ['a', 'c', 'b', 'e', 'd']
In:  print(G.edges())
Out: [('a', 'b'), ('a', 'e'), ('c', 'b'), ('c', 'd'), ('e', 'd')]
In:  nx.draw_networkx(G, node_color='y', node_isze=800)
Out:
```



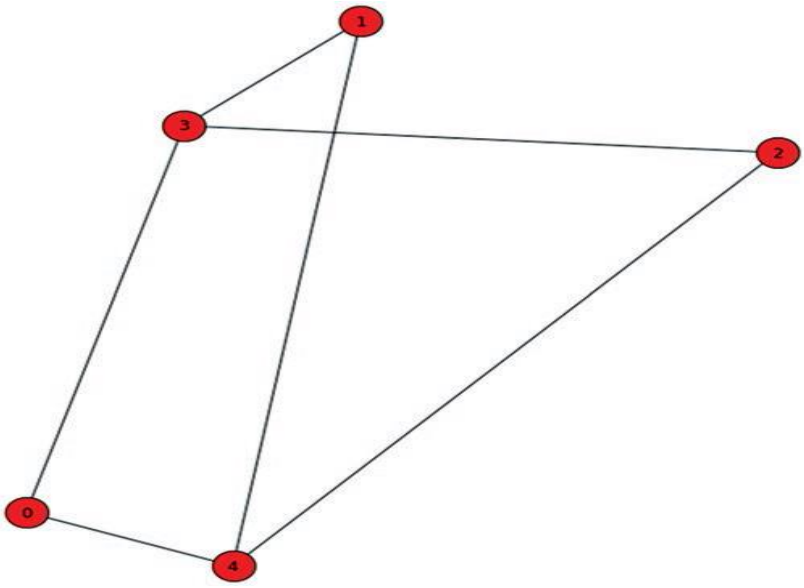
شکل ۹-۷ شبکهٔ یک-حالته از مجموعه دوم گره‌ها

پس از به دست آوردن شبکه یک-حالته از روی شبکه دو-حالته، می‌توانیم فراتر رفته و از روشهای استاندارد تحلیل شبکه استفاده کنیم. مثلاً می‌توانیم برای کشف گروههای منسجم از خوشه‌بندی سلسله‌مراتبی، شناسایی محافل یا مدلهای بلوکه‌ای استفاده کنیم. همچنین از روشهای مرکزیت میانی یا سرشماری سه‌وندها برای یافتن مهره‌های مرزی (پلها) استفاده کنیم. معیارهای مرکزیت نیز برای تعیین گره‌های مرکزی می‌توانند به‌کارگرفته شوند.

مثال ۳

روش دیگری نیز برای ایجاد گراف دو-بخشی کامل وجود دارد که مستقیماً از کتابخانه *NetworkX*، استفاده می‌کند (شکل ۱۰-۷):

```
In:  B = nx.complete_bipartite_graph(3, 2)
In:  nx.draw_networkx(B, nx.random_layout(B), node_size = 800)
Out:
```



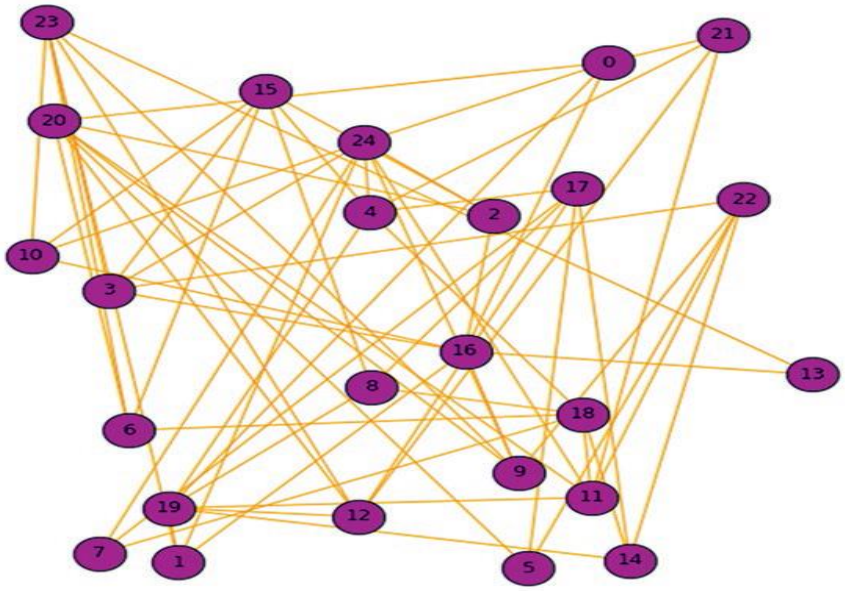
شکل ۱۰-۷ گراف کامل دو-بخشی با دو مجموعه گره (۲، ۳)

```
In:  X=set([0,1,2])
     bipartite.density(B,X)
Out: 1.0
In:  Y=set([3,4])
     bipartite.density(B,Y)
Out: 1.0
```

مثال ٤

در این مثال، یک گراف تصادفی دو-بخشی (شکل ۱۱-۷) می‌سازیم:

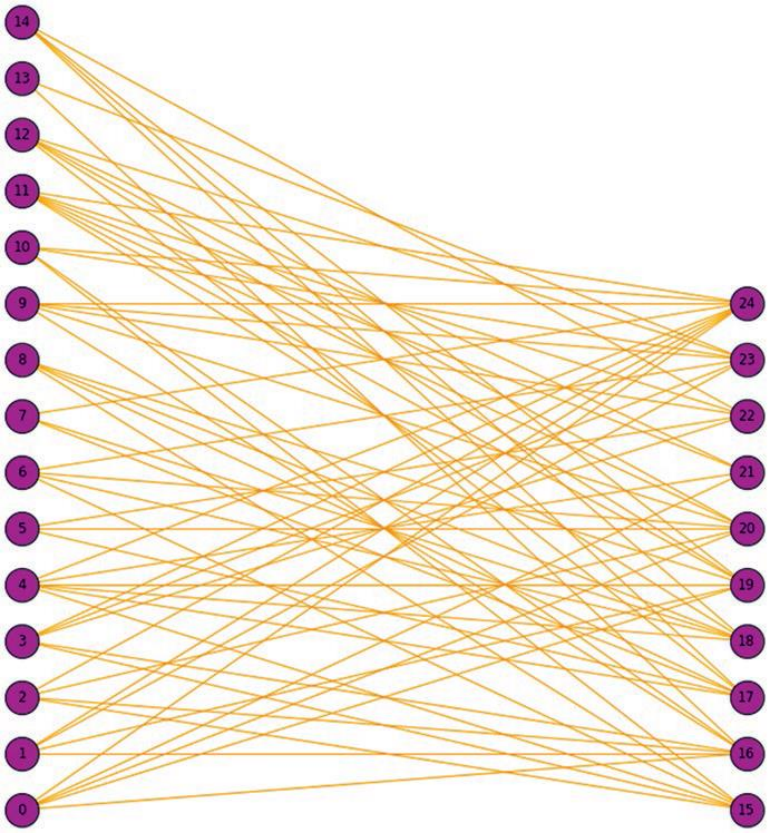
```
In:  randB = bipartite.random_graph(15, 10, 0.5)
     nx.draw_networkx(randB, nx.random_layout(randB), node_size=800,
     node_color='magenta', edge_color='orange')
Out:
```



شکل ۱۱-۷ گراف تصادفی دو-بخشی با دو مجموعه گره (۱۰، ۱۵)

احتمالاً با این نمایش، شکل ۱۱-۷ شبیه به یک گراف دو-بخشی نیست. حال بیایید برای یک نمایش بهتر، کاری بر روی آن انجام دهیم (شکل ۱۲-۷):

```
In:  X, Y = bipartite.sets(randB)
     position = dict()
     position.update( (n, (1, i)) for i, n in enumerate(X) ) # put
nodes
     from X at x=1
     position.update( (n, (2, i)) for i, n in enumerate(Y) ) # put
nodes
     from Y at x=2
     nx.draw(randB, pos=pos, with_labels=True, node_size=800,
     node_color='magenta')
     plt.show()
```



شکل ۱۲-۷ تغییر مکان گره‌های گراف دو-بخشی

اجازه دهید اطلاعاتِ این گراف را مشاهده کنیم:

```
In:  print(nx.info(randB))
Out: Name: fast_gnp_random_graph(30,50,0.5)
     Type: Graph
     Number of nodes: 80
     Number of edges: 726
     Average degree: 18.1500
```

اجازه بدهید بررسی کنیم آیا این گراف، واقعا دو-بخشی است یا خیر:

```
In:  nx.is_bipartite(randB)
```

```
Out: True
```

حال بیایید گره‌های هر دو مجموعه را بررسی کنیم:

```
In:  randB_top = set(n for n,d in randB.nodes(data=True) if
     d['bipartite']==0)
     randB_bottom = set(randB) - randB_top
In:  print(list(randB_top))
Out: [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18,
     19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29]
In:  print(list(randB_bottom))
Out: [30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45,
     46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62,
     63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79]
```

در این گراف، ما دو مجموعه گره یعنی گروه ۱ و ۰ داریم. بگذارید ببینیم چگونه می‌توان گروهِ هر گره را پیش‌بینی کرد. برای این منظور، قصد داریم از یک تابع دیگر، یعنی تابع *color()*، استفاده کنیم. این تابع یک دیکشنری ایجاد می‌کند که **کلیدش** شمارهٔ گره بوده و **مقدار** یک یا صفر را برای تعیین رنگ گره می‌دهد:

```
In:  c = bipartite.color(randB)
     print(c)
Out: {0: 1, 1: 1, 2: 1, 3: 1, 4: 1, 5: 1, 6: 1, 7: 1, 8: 1, 9: 1, 10: 1,
     11: 1, 12: 1, 13: 1, 14: 1, 15: 1, 16: 1, 17: 1, 18: 1, 19: 1, 20:
     1, 21: 1, 22: 1, 23: 1, 24: 1, 25: 1, 26: 1, 27: 1, 28: 1, 29: 1,
     30: 0, 31: 0, 32: 0, 33: 0, 34: 0, 35: 0, 36: 0, 37: 0, 38: 0, 39:
     0, 40: 0, 41: 0, 42: 0, 43: 0, 44: 0, 45: 0, 46: 0, 47: 0, 48: 0,
     49: 0, 50: 0, 51: 0, 52: 0, 53: 0, 54: 0, 55: 0, 56: 0, 57: 0, 58:
     0, 59: 0, 60: 0, 61: 0, 62: 0, 63: 0, 64: 0, 65: 0, 66: 0, 67: 0,
     68: 0, 69: 0, 70: 0, 71: 0, 72: 0, 73: 0, 74: 0, 75: 0, 76: 0, 77:
     0, 78: 0, 79: 0}
```

اجازه دهید گروه یکی از گره‌ها را بررسی کنیم:

```
In:  nx.set_node_attributes(randB, 'bipartite', c)
In:  print(randB.node[30]['bipartite'])
```

```
Out: 0
```

فصل هشتم

انتشار اطلاعات در شبکه‌های اجتماعی

در این فصل، درباره مفاهیم انتشار اطلاعات[1] در شبکه‌های اجتماعی بحث خواهیم‌کرد. ما علاقه‌مندیم بدانیم، چطور اطلاعات (یا دانش) از طریق شبکه‌ها، نشر و گسترش می‌یابد. مثالهای دیگری از این دست عبارتست از چگونگی انتشار ویروسهای کامپیوتری در اینترنت یا در شبکه‌ای از رایانه‌ها، اپیدمی یا واگیری بیماریها از طریق شبکه اجتماعی (حقیقی) و یا پراکنش و شیوع شایعات و ایده‌ها از طریق یک شبکه اجتماعی (مجازی). از مدلها یا روشهای انتشار اطلاعات معمولاً در بازاریابی ویروسی[2]، در سیستمهای فیلترینگ مشارکتی[3]، در مدیریت بحران[4]، در انجمن‌یابی و در مطالعه شبکه‌های استنادی استفاده می‌شود.

---

[1] Information Diffusion
[2] Viral Marketing
[3] Collaborative Filtering Systems
[4] Emergency Management

در این فصل، درمورد دو نوع کلی انتشار اطلاعات در شبکه‌های اجتماعی، مفصلاً بحث خواهیم کرد: انتشار نوآوری[۱] و همه‌گیری (اپیدمی)[۲]. انتشار نوآوری در بسیاری از زمینه‌ها مورد مطالعه قرار گرفته است، لذا در این فصل، ما تنها از دیدگاه شبکه اجتماعی آن را در نظر می‌گیریم. سایر انواع انتشار اطلاعات که در این فصل گنجانده نشده‌اند عبارتند از: رفتار گروهی (یا اصطلاحاً رفتار گله‌ای یا توده‌ای)[۳] و آبشار اطلاعات[۴].

۸-۱ انتشار

انتشار، فرآیندی است که از طریق آن اطلاعات از یک مکان به مکان دیگر و از طریق تعاملات مختلف، انتقال و گسترش می‌یابد. حوزه انتشار اطلاعات یا شیوع بیماریها، زمینه‌ای از دانش است که در آن از تکنیکهای بسیاری استفاده می‌شود. تکنکهایی که متعلق به رشته‌های مختلف علمی از قبیل جامعه‌شناسی، همه‌گیری‌شناسی (اپیدمیولوژی)[۵] و قوم‌نگاری (مردم‌شناسی)[۶] هستند. این مبحث مهم است چرا که مطمئناً هیچ‌کس علاقه‌مند به مبتلا شدن به یک بیماری مسری نیست!!

فرآیند انتشار شامل سه عنصر اصلی به شرح زیر است :

۱. **فرستنده**[۷]: یک فرستنده (یا گروهی از فرستندگان) مسئول ایجاد یا شروع فرآیند انتشار است.

۲. **گیرنده**[۸]: یک گیرنده (یا گروهی از گیرندگان) اطلاعات منتشره را از فرستنده دریافت می‌کند. معمولاً تعداد گیرندگان بیشتر از تعداد فرستندگان است.

۳. **رسانه (واسطه)**[۹]: کانالی است که از طریق آن اطلاعات منتشره از فرستنده به گیرنده ارسال می‌شود. این کانال می‌تواند تلویزیون، روزنامه، رسانه‌های اجتماعی (بعنوان مثال یک توییت در توییتر)، روابط اجتماعی، هوا (در مورد روند گسترش بیماری) و غیره باشد.

---

[1] Diffusion of Innovation
[2] Epidemics
[3] Herd Behavior
[4] Information Cascades
[5] Epidemiology
[6] Ethnography
[7] Sender
[8] Receiver
[9] Medium

از دیدگاه شبکه، سوال این است که فرایند انتشار چگونه انجام می‌شود؟ بی‌شک، روابط اجتماعی نقش مهمی ایفا می‌کنند. آنها کانالهایی هستند که منجر به اشاعه (سرایت)[۱] و اقناع (ترغیب)[۲] اجتماعی می‌شوند. به طور خاص، موقعیتهای ساختاری افراد و خصوصیات شخصی آنها، موجب می‌شود برخی افراد آمادگی بیشتری برای پذیرش نوآوری[۳] نسبت به دیگران داشته باشند. شبکه‌ها با الگوهای مختلفِ اتصال، دارای ویژگیهای متفاوتی در مورد چگونگی انتشار (اطلاعات) هستند که البته اثرات و پیامدهای قابل توجهی بر روی گسترش آن دارند. مثالی از این دست، فرایند پخش شایعات[۴] است.

عمل انتشار با یک پذیرنده[۵] (یا پذیرندگان) شروع می‌شود که موجب گسترش نوآوری به سمت دیگران می‌شود. مفهوم نوآوری، هرچند به طور معمول، معنای تازگی و جدید بودن را می‌دهد، اما همانند اختراع نیست. نوآوری از یک طرف، هم فرایند و هم خروجی (فرایند) است و از طرف دیگر، شامل تغییر دائمی[۶] است.

کسانی که زودتر یا ابتدائاً یک نوآوری جدید را می‌پذیرند، اغلب بسیار خلاقانه و مبتکرانه عمل کرده تا بتوانند بر دیگران در یک شبکه محلی، نفوذ و اثرگذاری داشته باشند. آنها مخاطبین خود را متأثر می‌کنند و به همین ترتیب مخاطبین آنها، مخاطبین جدیدی را متأثر می‌کنند. هر قدر فردی با افراد بیشتری ارتباط و پیوند داشته باشد، احتمال پذیرش نوآوری توسط وی، بیشتر است. در مقیاس وسیعتر و از آنجا که اجتماعات به یکدیگر متصل و مرتبط هستند، احتمال دارد که یک نوآوری از یک اجتماع با استفاده از مهره‌های مرزی (یا پلها) به اجتماع دیگر نفوذ کرده و مجدداً شروع به نشر کند. این وضعیت، در واقع از ویژگیهای مهم شبکه‌های اجتماعی است.

معمولاً این امکان وجود دارد که بتوان فرآیند انتشار را سرعت بخشید، به تأخیر انداخت و یا حتی متوقف کرد. مثلاً اگر مشخص شود یک محصول فرهنگی (مثلاً یک ویدیو، یک صوت، یک کتاب و یا امثال آن) نامطلوب و ناصواب است آنگاه آن محصول قاعدتاً باید اصلاح شده و سپس مجدداً نشر یابد. این فرآیند را اصطلاحاً مداخله[۷] می‌نامند. "فرایند مداخله" می‌تواند با روشهای مختلفی نظیر توقف تولید محصول، محدود کردن توزیع محصول، ایجاد محدودیت برای مواجهه با محصول، کاهش علاقه به

[1] Social Contagion
[2] Persuasion
[3] Adopt the Innovation
[4] Rumor Propagation
[5] Adopter
[6] Discontinuous Change
[7] Intervention

محصول یا کاهش تعاملات در میان جمعیت انجام شود. اما درعین حال باید دانست که، فرایندهای مداخله می‌توانند باعث آسیب به کار شرکتهای کوچک شوند، زیرا بسیاری از مشتریان، از آن پس، محصولات تولیدی این شرکتها را نخواهند پذیرفت.

۸-۲ واگیری یا سرایت

یکی دیگر از مفاهیم مهمی که گاهی اوقات در کنار انتشار استفاده می‌شود، سرایت یا واگیری[1] است که چگونگی گسترش سریع یک بیماری در یک شبکه را توضیح می‌دهد. از منظر بیولوژیکی، واگیری نیاز به تماس فیزیکی نزدیک، برای انتشار دارد. با این حال، گاه تنها حضور داشتن در همان محلی که در آن عفونت یا میکروب وجود دارد، برای گسترش بیماری کافی است. حتی در این موارد، عوامل محدودکننده‌ای (همانند سن، جنس، ایمنی، قد، وزن، سلامت کلی، قدرت ویروس و زمان تماس با افراد آلوده) وجود دارد که نشان می‌دهد آیا فرد در معرض بیماری، بیمار می‌شود یا خیر.
هر چند مدلهای واگیری[2] شبیه به مدلهای انتشار (یا شیوع اجتماعی[3]) هستند اما تفاوتهایی هم دارند:

- **واگیری** اغلب از طریق لنز بیماریهای عفونی دیده می‌شود (البته گاهی اوقات از این واژه برای توصیف پدیده‌های مختلف در بازاریابی و فضاهای اجتماعی استفاده می‌شود). از سوی دیگر، **انتشار**، فرآیند یا حالتِ گسترش بسیار وسیع چیزی (مانند ایده، نوآوری، شایعه، دارایی دیجیتالی، و غیره) است.

- **واگیری** معمولاً با یک عفونت آغاز می‌شود، در حالی که **انتشار** به طرق گوناگونی آغاز می‌شود؛ شیوه‌هایی همچون استفاده از رسانه‌های سنتی، صحبت کردن، تبلیغات و یا رویدادهای صنعتی.

- **واگیری** نیاز به تماس مستقیم بین قربانی و فرد آلوده ندارد، اما **انتشار**، بدنبال گسترش ایده‌ها، نوآوریها و مفاهیم دیگری است که نیاز به ارتباط مستقیم (و نه لزوماً یک ارتباط فیزیکی) دارد.

[1] Contagion
[2] Contagion Models
[3] Social Contagion

- **واگیری** وابسته به مجاورت فیزیکی فرد بیمار با افراد غیر مبتلاست و این موضوع، جدا از این حقیقت است که آیا افراد سالم، بخشی از شبکه اجتماعی یا حرفه‌ای فرد مبتلا هستند یا خیر. این مورد با **انتشار** متفاوت است که در آن به تماس اجتماعی (و نه ضرورتاً مجاورت فیزیکی) یا نفوذ ایده به درون ذهن فرد نیاز است.

- مدلهای **واگیری** شامل فرایندهایی مانند تصمیم‌گیری نیستند، که در آن فرد تصمیم بگیرد که مبتلا شود یا خیر، زیرا واگیری بعنوان یک فرآیند طبیعی تصادفی عمل می‌کند. اما در مدلهای **انتشار**، معمولاً انتشاردهنده نیز دخیل است. بعنوان مثال، زمانی که یک شایعه گسترش می‌یابد، افرادی که این شایعه را دریافت می‌کنند تصمیم می‌گیرند که آیا آنها علاقه‌مند به گسترش آن شایعه به همسایگان خود هستند یا خیر.

در بخش زیر، ما در مورد انتشار نوآوری بحث خواهیم کرد که توضیح می‌دهد چگونه ایده‌ها و شیوه‌های جدید در داخل و بین اجتماعات گسترش می‌یابند.

۸-۳ انتشار نوآوری

انتشار نوآوری، بعنوان یک فرآیند اجتماعی مهم، نحوهٔ دست بدست شدن (یا دهان به دهان شدن) یک نوآوری (مثلاً محصول، موسیقی، ویدیو، محبت، افکار یا نگرش) از یک گره (فرد) به گره دیگر در یک شبکه اجتماعی را در طول زمان توصیف می‌کند. این فرآیند، عملاً یک نظریه است که اصول و دیدگاههای شبکه را به طور گسترده‌ای مورد استفاده قرار داده است. بعلاوه مبانی نظری خوبی را برای تحقیق درباره اینکه چگونه شبکه‌ها بر رفتار (و تغییر و تثبیت آن) تاثیر می‌گذارند، ارائه نموده است.
منشاء ایده‌ها، الگوها و یا شیوه‌های جدیدی که به اجتماعات وارد می‌شوند معمولاً منابع بیرون از اجتماعات نظیر رسانه‌های جمعی، حضور برخی کارکنان جدید، نوآوریهای فناورانه، ارتباطات جهانی و بسیاری موارد دیگر است. همچنین ایده‌های نو می‌توانند از درون همان انجمن یا اجتماعی که در آن

منتشر می‌شوند، سرچشمه بگیرند. در همه این موارد، انتشار از طریق شبکه‌های "تماس میان–فردی"[1] (یا ارتباطات بین افراد) انجام می‌شود.

در حوزه‌های مختلف انسان‌شناسی، اقتصاد، جغرافیا، جامعه‌شناسی، بازاریابی، اپیدمیولوژی (همه‌گیر شناسی) و غیره، انتشار نوآوری برای بسیاری از گروهها حیاتی است. بعنوان مثال، سازمانها علاقه‌مند به انتشار اطلاعات خود هستند و ایده‌پردازان علاقه‌مند به پذیرش محصولات جدید هستند.

صدها مطالعه در دهه ۱۹۵۰ و اوایل دهه ۱۹۶۰ انجام شد تا به چرایی و چگونگی گسترش نوآوری، دلایل روند انتشار و سوالات دیگر پاسخ دهند. سوالاتی همچون اینکه چرا برخی افراد، یک ایده جدید را می‌پذیرند؟ چرا بعضی از مردم، قبل از دیگرانی که زمان قابل توجهی را صرف بررسی و پذیرش یک ایده جدید می‌کنند، آن ایده را می‌پذیرند؟ و بالاخره میزان نرخ گسترش یک نوآوری تا چه حد است؟ در سالهای اخیر با ظهور مدلهای پیچیده‌تر شبکه و فناوری مرتبط با آن، تحقیقات در این زمینه از نو آغاز شده است.

لازم به ذکر است که انتشار نوآوری به طور معمول زمان زیادی طول می‌کشد. بعنوان مثال، چندین دهه طول کشید تا تلفن در ایالات متحده آمریکا محبوب شود و نیز مدت زیادی طول کشید تا ضبط کننده ویدئویی (یا VCR) همه‌گیر شود. دلیل کوتاه یا بلند بودن زمانِ انتشار، اغلب به وضعیت ساختار شبکه برمی‌گردد، که بعضاً به یک یا چندین دلیل، ممکن است فرایند انتشار، کُند یا تند شود. برخلاف مثالهای قبل، ظهور فنآوریهای کامپیوتری و ارتباطات تلفن همراه، سرعت انتشار اطلاعات و پذیرش سایر محصولات را افزایش داده‌اند. برای مثال، تنها چند سال طول کشید تا فیسبوک میلیونها کاربر را به دست آورد.

۸–٤ پذیرش نوآوریها

پذیرشِ نوآوری، یک تصمیم شخصی (یا گروهی) است که به معنای استفاده کامل از نوآوری بعنوان بهترین راهکار موجود است. نکته مهم این است که فعل پذیرش، درست بعد از اینکه شخص برای اولین بار در مورد یک محصول جدید اطلاعات بدست می‌آورد، صورت نمی‌گیرد. بلکه در نظریه انتشار

[1] Interpersonal Contact Networks

نوآوریها، رویه پذیرشِ نوآوری مشتمل بر یک فرایند پنج مرحله‌ای است. این مراحل را می‌توان همچنین برای استفاده در بخش‌بندی بازار[1] و یا برای سنجش پیشرفت در تغییر رفتار، بکار گرفت. این مراحل عبارتند از:

۱. آگاهی یافتن از وجود محصول (اما با اطلاعات محدود)
۲. شروع به یافتن اطلاعات بیشتر در مورد محصول
۳. تصمیم‌گیری جهت پذیرش
۴. امتحان محصول
۵. پذیرش محصول به طور کامل

۸-۵ مدلهای انتشار نوآوری

شبکه‌های اجتماعی، امکان گسترش بسیاری از ایده‌ها، آداب، رسوم، شیوه‌ها، رویه‌ها و الگوهای جدید را از طریق "تماسهای میان-فردی" فراهم می‌نمایند. این تماسها عمدتاً از جنس تعاملات، معاشرتها و یا ارتباطات میان افراد هستند. متخصصان معتقدند نحوه گسترش بسیاری از ایده‌ها را می‌توان با روشهای مشابه‌ای مدلسازی کرد. از این رو، دانستن اینکه چگونه بیماریها، مثلاً، از طریق شبکه‌ها شیوع می‌یابند، به ما کمک خواهد کرد که بدانیم چطور بقیه موضوعات همه‌گیر می‌شوند. در بازاریابی کالاها، در سطح خرده‌فروشی[2]، بعنوان مثال، اطلاعات لازم، نظیر نتیجه بررسیها و نظرسنجیها، بدون ایجاد هیچ هزینه‌ای برای فروشنده، نشر یافته و به سمع و نظر مشتریان بالقوه و جدید می‌رسد. این روش گسترشِ اطلاعاتِ محصولات را اصطلاحاً "بازاریابی ویروسی" می‌نامند. این مکانیزمِ انتشار اطلاعات، مهم است، زیرا فرصت آگاهی از اینکه مردم چگونه به یک محصول خاص می‌نگرند را فراهم کرده و فروشنده، همزمان می‌تواند نتایج را بررسی کند.

رسانه‌های اجتماعی در اکثر اجتماعات، نقش اساسی و مهمی در گسترش اطلاعات دارند. از اینرو مدلهای بسیاری برای نشان دادن فرایندی که در آن رسانه‌های اجتماعی مسئول گسترش اطلاعات (یا

[1] Market Segmentation
[2] Retail Marketing

ارتباط جمعی[1]) هستند، ارائه شده است. در این بخش، یکی از این مدلها که یک مدل جریان دو مرحله‌ای[2] است (و بعضاً مدل جریان چند مرحله‌ای[3] نیز نامیده می‌شود) را بررسی می‌کنیم. این مدل توسط الیو کاتز[4]، استاد ارتباطات در دانشگاه پنسیلوانیا پیشنهاد شد. پروفسور کاتز ضمن همکاری با پاول لازارسفلد[5] و رابرت مارتون[6]، پیشگام بسیاری از نوآوریها در خصوص نحوه اثرگذاری رادیو و تلویزیون بر انبوه مخاطبان بود.

٦-٨ مدل جریان دو مرحله‌ای

مدل "جریان دو مرحله‌ای" یکی از ساده‌ترین مدلها برای مدلسازی فرایند انتشار نوآوری است. این مدل تاثیر رسانه‌های جمعی بر انواع رفتارهای مردم از جمله رفتار مشتریان و الگوهای رای‌گیری[7] را مورد بررسی قرار می‌دهد. این مدل برخلاف دیدگاه قبلی که معتقد بود رسانه‌های جمعی به طور مستقیم مردم را تحت تاثیر قرار می‌دهند، براساس این ایده است که، اثرات رسانه‌ها بواسطه نفوذ بین–فردی، نافذ می‌شود.

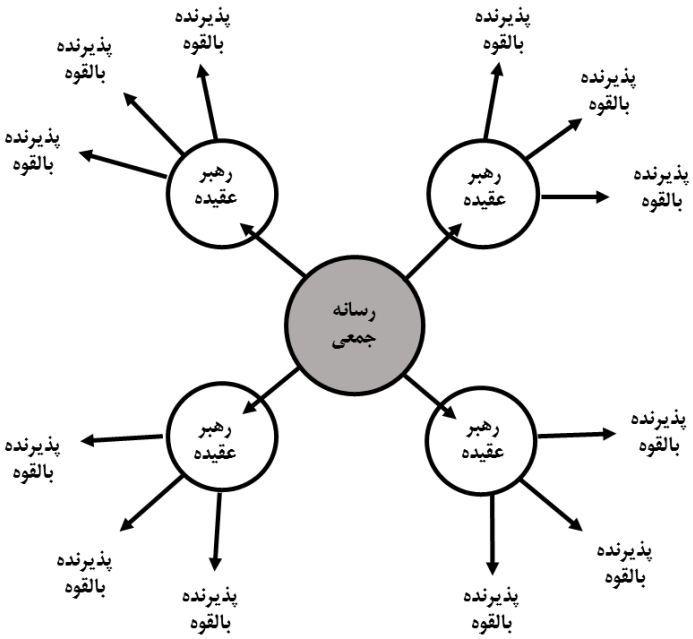
در این مدل، که با رویکرد شبکه‌ای، سازگار است، فرآیندهای ارتباط جمعی به دو مرحله تقسیم می‌شوند.

۱) در مرحله اول، رسانه‌های جمعی بر "رهبران عقیده" تاثیر می‌گذارند. رهبران عقیده[8]، افرادی هستند که به طور معمول بدلیل سن، تجربه، شایستگی، تعصب و تصور همریختی یا همگونی آنها، تأثیر زیادی بر رفتار بقیه اعضا در شبکه خود دارند. آنها از هر دو نوع رابطه، یعنی روابط اجتماعی صامت[9]، (بی پند و موعظه) و نیز روابط دوستانه (و با پند و مشورت) استفاده می‌کنند تا به مخاطبان خویش در مورد اهمیت یک نوآوری و علت پذیرش آن بگویند.

---

[1] Massa Communication
[2] Step Flow Model
[3] Multistep Flow Model
[4] Elihu Katz
[5] Paul Lazarsfeld
[6] Robert Merton
[7] Voting Patterns
[8] Opinion Leaders
[9] Silent Social Relations

۲) در مرحله دوم، رهبران عقیده، عملاً "پذیرندگان بالقوه"[1] را تحت تاثیر قرار می‌دهند. این گروه دوم همان افرادی هستند که در معرض پیامهای رسانه‌های جمعی قرار گرفته بودند ولی به طور خودکار آنها را نمی‌پذیرفتند. از آنجایی که "رهبران عقیده" بیشتر در معرض رسانه‌ها قرار داشتند و بیشتر از تمایلات و گرایشهای جاری آگاه هستند، می‌توانند به کمک ارتباطات رسانه‌ای به پشتیبانی از استدلال خود پرداخته و دیگران را متقاعد ساخته تا دیدگاههایشان را دنبال کنند. این مدل به دلیل اینکه "رسانه" بر "رهبران عقیده" اثر گذاشته و رهبران عقیده نیز بر "دیگر پذیرندگان" تاثیر می‌گذارند، یک "مدل جریان چند مرحله‌ای" نامیده می‌شود (شکل ۱-۸).



شکل ۱-۸ مدل جریان دو مرحله‌ای

در ادامه مثالی ساده از نحوه عملکرد مدل "دو مرحله‌ای جریان" بررسی می‌شود. فرض کنید یک شرکت سعی دارد یک برنامه نرم‌افزاری جدید را در شبکه‌ای از کاربران به بازار عرضه کند. این شرکت بودجه کافی برای مدیریت و راه‌اندازی پویش[2] یا کمپین بازاریابی ندارد. بنابراین، تعدادی از کاربران (صاحبان

[1] Potential Adopters
[2] Campaign

فروشگاه) را برای معرفی و ترویج نرم‌افزار خود انتخاب می‌کند، این کاربران، صاحبان فروشگاههایی هستند که در شبکه‌های محلی خود با بالاترین درجه ورودی بوده، و بنوعی مرکزی هستند. آنها می‌توانند با دوستان، مشتریان و مخاطبان اجتماعی خود در مورد نرم‌افزار و چرایی این روند جدید در بازار صحبت کنند. همسایگان آنها، نیز به نوبه خود، در مورد محصول با همسایگان خویش صحبت می‌کنند و به همین ترتیب، تا زمانی که نهایتاً، اخبار مربوط به محصول در بین جمعیت زیادی از کاربران در شبکه گسترش می‌یابد. در این مثال ساده، شرکت مورد مطالعه در جایگاه "رسانهٔ جمعی" است، صاحبان فروشگاهها در واقع "رهبران عقیده" بوده، و بالاخره همسایگان، همان "پذیرندگان بالقوه" هستند.

۸-۷ شیوع اجتماعی

فرایندی که در آن، نوآوری از یک پیوند یا یال عبور کرده و به دیگری منتقل می‌شود، اشاعه اجتماعی[۱] نامیده می‌شود. این فرایند از جهات بسیاری مشابه گسترش بیماریهای عفونی یا واگیری است. این مدل، که زنجیره‌ای شکل[۲] است، با تعداد افراد کمی که نوآوری را می‌پذیرند، شروع می‌شود. این تعداد در مرحله دوم، با پذیرش بسیاری از افراد دیگر، افزایش می‌یابد. اگرچه تعداد پذیرندگان در مرحله دوم به طور عمده افزایش می‌یابد، اما نرخ رشد آنان کاهش می‌یابد. وضعیت کاهش رشد، به ویژه پس از اینکه ۱۰ تا ۲۰ درصد از کنشگران، آن نوآوری را پذیرفتند، رخنمایی می‌کند. در مرحله سوم، تعداد افرادی که این نوآوری را می‌پذیرند، کاهش می‌یابد تا زمانی که سرانجام روند انتشار به آرامی پایان یابد.

افرادی که تصمیم می‌گیرند یک نوآوری را بپذیرند، این کار را یا بطور مستقل انجام می‌دهند یا بطور وابسته (بطور وابسته یعنی این فرد بعد از اینکه متوجه شد افراد دیگر این نوآوری را پذیرفته‌اند، پذیرای آن شده است). کسانیکه در پذیرش نوآوری وابسته‌اند، چنین فرض می‌کنند که وقتی دیگران آنرا پذیرفته‌اند پس حتماً این نوآوری ارزشمند است و یا حداقل یک دلیل قوی بر درستی آن وجود دارد. اطلاعاتی که آنها دریافت می‌کنند می‌توانند محلی بوده (اطلاعات مربوط به رفتار همسایگان نزدیک

[1] Social Contagion

توجه داشته باشید که ما بعنوان معادل واژه Contagion در حوزه مسائل پزشکی و سلامت، کلمات "واگیری" و "سرایت"را انتخاب کرده‌ایم اما برای همین واژه در حوزه اجتماعی، مثلاً گسترش ایده و نوآوری از کلمات "شیوع" و"اشاعه" استفاده کرده‌ایم.

[2] Chain-Like Form

خود، مثلاً همکاران در یک شرکت) یا فرامحلی باشد (اطلاعات مربوط به رفتار همه افراد در شبکه گسترده، مثلاً یک اتحادیه).

فرایند پذیرش یک نوآوری را می‌توان به صورت یک نمودار به نام منحنی انتشار[1] نمایش داد. در این منحنی، که به شکل "تابع سیگموئید[2]" (S) است، محور x، طول عمر انتشار[3] و محور y درصد پذیرندگان نوآوری است.

لازم به ذکر است که "اندازه پذیرش" و "سرعت آن" در شبکه‌های متراکم و چگال بمراتب از اندازه و سرعت آن در شبکه‌های تُنُک، شبکه‌های غیرمتصل و شبکه‌هایی با نقاط برشی یا پلها، بیشتر است. همچنین احتمال پذیرش فردی با تعداد همسایگان بیشتر، از احتمال پذیرش فردی با همسایگان کمتر، بالاتر است و نیز سرعت فرایند انتشار، به هنگامی که با فردی که دارای موقعیت مرکزی در شبکه است، آغاز شود، بیشتر از سرعت فرایند انتشار است آنگاه که با شخصی در حاشیه شبکه، شروع شود.

۸-۸ نرخ پذیرش

نرخ پذیرش[4] معیاری برای اندازه‌گیری "سرعت فرآیند انتشار" در یک زمان خاص است. این معیار نشاندهنده تعداد افرادی است که نوآوری را در یک زمان معین پذیرفته‌اند. واضح است که ساختار شبکه تأثیر زیادی بر روند انتشار دارد، به گونه‌ای که فرآیند انتشار وقتی با گرهی که در یک موقعیت مرکزی شبکه است آغاز شود، می‌تواند در زمان کوتاهی به تعداد زیادی از افرادِ جامعهٔ هدفش برسد. با این حال، ساختار شبکه، تنها پارامتر مطرح نیست. نوع نوآوری و خصوصیات افراد نیز بر جریان پذیرش تاثیر می‌گذارد.

[1] Diffusion Curve
[2] Sigmoid Function
[3] Lifetime of a Diffusion
[4] Adoption Rate

۸-۹ طبقه‌بندی و آستانه پذیرش

واضح است که کسی با ابتلاء همسایه‌اش به یک بیماری، به آن بیماری مبتلا نمی‌شود. همچنین در حوزه انتشار ایده‌ها، داشتن رابطه دوستی به تنهایی برای متقاعد کردن کسی به منظور پذیرش یک اندیشه و یا یک شیوه کافی نیست. این وضعیت بدان دلیل است که برخی افراد نسبت به بقیه، کمتر تحت تاثیر نوآوری قرار می‌گیرند. یک روش مناسب برای توضیح این واقعیت، استفاده از طبقه‌بندی پذیرش[۱] است.

براساس "طبقه‌بندی پذیرش"، می‌توان افراد را در مقایسه با دیگر پذیرندگان، به لحاظ زمان پذیرشِ یک نوآوری دسته‌بندی کرد. برای مثال، در بازاریابی، پذیرندگان به چهار طبقه یا دسته تقسیم می‌شوند: پذیرندگان آغازین[۲] که ۱۶٪ را تشکیل می‌دهند؛ اکثریت اولیه[۳] که ۳۴٪ را تشکیل می‌دهند؛ اکثریت ثانویه[۴]، که آنان نیز ۳۴٪ را تشکیل می‌دهند و پذیرندگان نهایی[۵] یا دیرپذیران (یا واماندگان)[۶]، که ۱۶٪ از همه پذیرندگان را تشکیل می‌دهند. این نوع طبقه‌بندی برای بازاریابان مهم است؛ زیرا به آنها کمک می‌کند تا مشخصه‌های اجتماعی و جمعیت‌شناختی پذیرندگان آغازین را شناسایی کنند.

۸-۱۰ میزان مواجهه

میزان مواجهه یا در معرض بودن[۷]، برای یک فرد خاص در یک زمان معین، نسبتی از همسایگان آن فرد در شبکه است که تا قبل از آن زمان، یک نوآوری را پذیرفته‌اند. هنگامی که میزان مواجهه به اندازه مقدار مورد نیاز آن فرد می‌رسد، آنگاه آن شخص، نوآوری را می‌پذیرد (اصطلاحاً مبتلا می‌شود) و شروع به مبتلا کردن بقیه می‌کند.

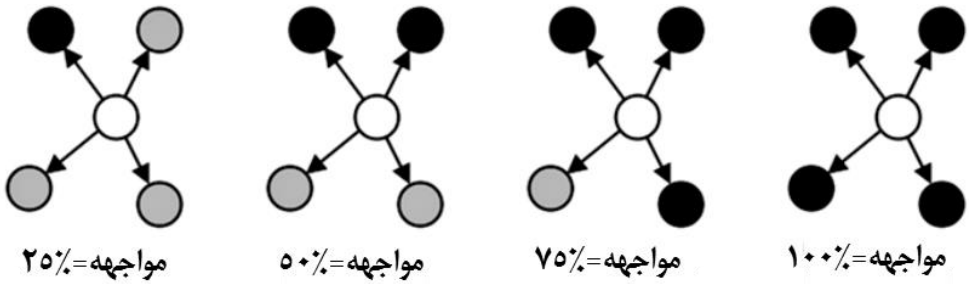
---

[1] Adoption Categories
[2] Early Adopters
[3] Early Majority
[4] Late Majority
[5] Late Adopters
[6] Laggards
[7] Amount of Exposure

این مدل با زمان، برخورد واضح‌تری دارد، بدین معنا که تلاش می‌کند اتفاقات مربوط به هر لحظه از زمانِ انتشار را مدل کند. برخی از محققان از میزان مواجهه بعنوان "مواجهه شبکه"[1] یاد می‌کنند که براساس تاثیر شبکه اجتماعی فرد بوده و با معادله ۱–۸ اندازه‌گیری می‌شود:

$$E_i = \frac{\sum W_{ij} y_i}{\sum W_i} \qquad ۱–۸$$

که در آن: W ماتریس وزن شبکه است که نشاندهنده تعداد ارتباطات مستقیم یک نفر است و y یک بردار رفتار پذیرش[2] است. در مورد شبکه‌های فردانه (یا فردمحور)، مدل مواجهه شبکه می‌تواند با توجه به عوامل بسیاری مانند فراوانی تعاملات و شباهت بین گره فردانه و همسایگانش، وزندار شود. با استفاده از داده‌های جامعه‌سنجی[3]، می‌توان بسیاری از انواع تأثیرات اجتماعی را برای وزندهی مدل مواجهه شبکه بکار گرفت. فرض کنید که یک شخص در شبکه اجتماعی خود دارای چهار دوست است. مواجهه شبکه (E_i) در اینجا معادل نسبت کسانی است که نوآوری را پذیرفته‌اند (شکل ۲–۸).



شکل ۲–۸ میزان مواجهه در چهار مرحله

شکل ۲–۸ "مواجهه شبکه" برای یک فرد با چهار دوست در چهار نقطه از زمان را نشان می‌دهد. در اولین نقطه زمانی، این شبکه، دارای یک پذیرنده است، بنابراین مواجهه شبکه ۲۵٪ است. در دومین نقطه زمانی، دو پذیرنده وجود دارد، بنابراین مواجهه شبکه ۵۰٪ است. در نقطه زمانی بعدی، سه پذیرنده وجود دارد، بنابراین مواجهه ۷۵٪ است. پس از آنکه تمام چهار نفر از دوستان، ایده یا نوآوری را بپذیرند، مواجهه ۱۰۰٪ است.

[1] Network Exposure
[2] Vector of Adoption Behavior
[3] Sociometric Data

پذیرش نوآوری یک فرآیند ساده نیست، زیرا به خصوصیات فرد و ویژگیهای نوآوری بستگی دارد. چه بسا برخی افراد به منبع انتشار به اندازه کافی نزدیک نباشند اما از جمله "پذیرندگان آغازین" نوآوری باشند. زیرا میزان مواجهه در طول زمان و در میان مردم تغییر می‌کند. آگاهی در مورد اینکه دوستان یک فرد، نوآوری را پذیرفته‌اند و یا افرادی که دارای موقعیت مشابه او در شبکه هستند، آنرا پذیرفته‌اند، ممکن است شخص موردنظر را متقاعد به پذیرش نوآوری کند.

برای اینکه فردی پذیرنده بالقوه یک نوآوری باشد، باید آن نوآوری را مرتبط با نیازهایش ببیند و بعلاوه از نظر محتوا، سودمندی، ضرورت و منبع نوآوری با آن موافقت داشته باشد. یعنی، نوآوریها باید قابل مشاهده یا چشمگیر بوده، مزیت نسبی نسبت به الگوها و شیوه‌های فعلی داشته، با پارادایم اجتماعی فرهنگی سازگار بوده و بالاخره بسیار پیچیده نباشند.

این امکان وجود دارد که بر اساس یک فرایند محاسباتی خاص، "زمان پذیرش نوآوری" برای هر فرد در هر نقطه زمانی را تعیین نمود. چنین فرایندی را "تحلیل تاریخچه رویداد"[1] می‌نامند. این فرایند از آن جهت مهم است که وقتی ما محصول جدیدی را عرضه می‌کنیم، می‌توانیم برای هر فرد تحلیل کنیم که آیا او این محصول را پذیرفته یا خیر و اینکه چند نفر از همسایگانش نیز آن را پذیرفته یا نخواهند پذیرفت.

۱۱-۸ پذیرندگان و پذیرش

در مورد موضوع مواجهه، دو مفهوم متفاوت وجود دارد که نباید آنها را یکسان انگاشت. اولی "آستانه پذیرش" است و دومی "امکان یا میزان مواجهه". در مورد مفهوم اول، لازم است بدانیم که افراد با توجه به خصوصیات خود، باید در سطوح مختلفی از مواجهه قرار بگیرند، تا یک نوآوری را بپذیرند. بعنوان مثال، بعضی از افراد به راحتی، با آگاهی از پذیرش عده‌ای دیگر، متقاعد می‌شوند؛ در حالی که برخی دیگر در زمان طولانی‌تری همان ایده را می‌پذیرند. مفهوم دوم که میزان یا امکان مواجهه است بدین معناست که برخی افراد، معمولاً در مواجهه بیشتری نسبت به دیگران قرار می‌گیرند. این مواجهه ممکن است از طریق رسانه‌ها یا روابط اجتماعی باشد. در همین راستا، اثرگذاری نسبی به معنای اثری است که پذیرش یک فرد بر دیگر افراد دارد.

[1] Event History Analysis

همانطور که ذکر شد مفهوم اول، "آستانه مواجهه"[1] است که نشان دهنده سطح مواجهه مورد نیاز یک فرد برای پذیرش نوآوری است. شناسایی آستانه (که نقاط اوج انفرادی[2] نیز نامیده می‌شود) برای محققان در جهت شناسایی انواع مختلف پذیرندگان و آستانه بالا و پایین آنان مهم است.

پذیرندگان آغازین، به نوآوری حساس‌ترند، به طوریکه زمانی که تنها چند پذیرنده در شبکه وجود دارد، نوآوری را خواهند پذیرفت. به عبارت دیگر، آستانه آنها پایین است. چنین افرادی منتظر پذیرش اکثریت کنشگران شبکهٔ خود برای پذیرش نوآوری نیستند. آنها ریسک می‌کنند و قبل از پذیرش دوستانشان، رفتارهای جدیدی را پیش می‌گیرند.

از سوی دیگر، پذیرندگان نهایی (دیرپذیران) به سختی متقاعد می‌شوند و تنها زمانی که اکثریت افراد در شبکه، نوآوری را پذیرفته باشند، آنرا خواهند پذیرفت. آنها به طور معمول در شبکه‌های اجتماعی تُنُک قرار داشته، دارای موقعیت اجتماعی پایینتری بوده، کمتر در مواجهه با رسانه‌های جمعی قرار داشته و بالاخره تمایل دارند در مورد ایده‌ها یا محصولات جدید از کانالهای میان–فردی، به ویژه همتایان مورد اعتماد، بشنوند و یا یاد بگیرند.

افراد با آستانه پایینتر، خلاقیت بیشتری دارند و انتظار می‌رود که یک نوآوری را زودتر از افرادی که خلاق نیستند، بپذیرند. انتظار داریم که این افراد به رسانه‌ها رجوع کنند تا ایده‌ها و روندهای جدید را بیاموزند. معمولاً چنین مراجعه‌ای عملاً رخ می‌دهد، زیرا آنها همتایان کمتری دارند تا بتوانند با آنان در مورد ایده‌ها یا روندهای جدید مشورت کنند. هنگامی که پذیرش توسط آنان اتفاق می‌افتد، آنها ایده یا روند جدید را به جامعه محلی خود انتقال داده و بعنوان پل برای دیگران عمل می‌کنند. رابطهٔ مثبت بین آستانهٔ مواجهه پایین و خلاقیت بالا (یعنی پذیرش نوآوری زودتر از کنشگران دیگر در دایره یا حلقه اجتماعی[3] خود) دارای شرایطی است که از آن جمله: استفاده گسترده از رسانه، داشتن مخاطبین بسیار در خارج از جامعه محلی، سطح بالای تحصیلات و وضعیت بالای اجتماعی و اقتصادی می‌باشد.

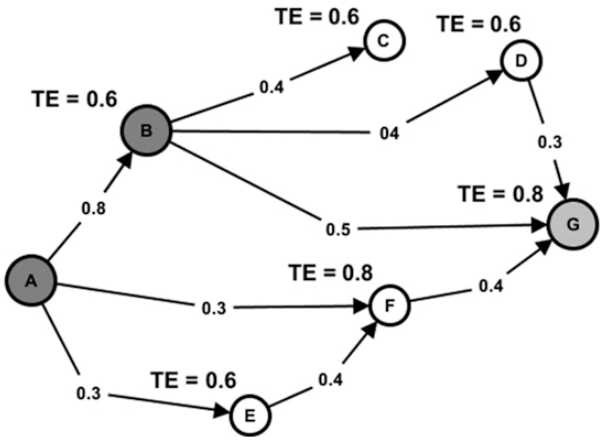
به مدل پذیرشمان برگردیم، پذیرندگانِ نخستین (زودپذیران یا اولین کسانیکه در شبکه ما نوآوری را می‌پذیرند) این فرصت را ندارند که در مواجهه با کسی قرار گیرند که از قبل نوآوری را پذیرفته باشد. بنابراین، آستانه آنها صفر است. از سوی دیگر، پذیرندگانِ آخرین (دیرپذیران) احتمالاً به پذیرندگان نخستین متصل هستند. بنابراین مواجهه و آستانه آنها در زمان پذیرش بالا است.

---

[1] Threshold of Exposure
[2] Individual Tipping Points
[3] Social Circle

مثال

اجازه دهید نگاهی به یک فرآیند انتشار ساده بیندازیم که مربوط به پذیرش نوآوری در یک شبکه کوچک است (شکل ۳-۸):



شکل ۳-۸ فرایند انتشار ساده

در این مورد، معیارهای زیر را داریم:

- این شبکه جهتدار است. گره‌ها (کنشگران) و یالها، کانالهای ارتباطی بین کنشگران را نشان می‌دهند. یک گره فقط می‌تواند بر گره (یا گره‌هایی) که به آن متصل است اثر داشته باشد. هنگامی که یک گره فعال[1] می‌شود، می‌تواند گره‌های همسایه خود را نیز فعال کند.

- مقدار TE در گراف، "آستانهٔ مقادیر مواجهه" است که نشاندهنده سطح مواجهه مورد نیاز فرد جهت پذیرش نوآوری است. بعنوان مثال، گره‌های G و F هر دو دارای آستانه مواجهه ۰٫۸ هستند و هر یک از گره‌های باقیمانده، دارای آستانه‌ای برابر با ۰٫۶ می‌باشند.

- اعداد روی یالها نشان‌دهنده درجه اثرگذاری نسبی (یعنی مواجهه شبکه مانند اعتماد، ترغیب، پیروی یا تقلید کورکورانه) یک گره بر گره دیگر است. بعنوان مثال، گره A که منبع انتشار[1] است، به ترتیب دارای اثرگذاری نسبی ۰٫۸، ۰٫۳ و ۰٫۳ بر گره‌های B، F و E است.

[1] Activated

- گره B در حال حاضر این نوآوری را پذیرفته است، زیرا اثرگذاری نسبی گره A بر گره B بیش از آستانه مواجهه گره B (یعنی ۰٫۶) می‌باشد.

- گره G در حال فکر برای پذیرش این نوآوری است. این گره یک حمایت تجمیعی کافی از سوی همسایگانش دریافت می‌کند که فراتر از آستانه مواجهه‌اش، یعنی ۰٫۸ می‌باشد. در عین حال توجه داشته باشید که تنها پذیرنده بودن گره B برای پذیرش گره G کافی نیست. برای ایجاد اعتماد کافی این گره جهت پذیرش (ارضا آستانه مواجهه)، نیاز به پشتیبانی بیشترِ یکی از دو گره باقی مانده است.

- هیچ کدام از گره‌های C، D، E و F مقدار اثرگذاری یا نفوذ کافی از همسایگان خود جهت ارضا آستانه مواجهه‌شان را دریافت نکرده‌اند؛ یعنی امکان یا میزان مواجهه از آستانه مواجهه کمتر است.

۸-۱۲ جِرم بحرانی

یک فرایند انتشار وقتی به موفقیت نایل می‌گردد که تقریباً همه افراد در جامعهٔ هدف، آن نوآوری را بپذیرند. به بیان دیگر، اگر فقط تعداد کمی از مردم، نوآوری را پذیرفته و گسترش دهند، ممکن است این فرایند شکست بخورد. این وضعیت را می‌توان با مفهومی تحت عنوان "جرم بحرانی"[۲] توجیه کرد. این جرم نشاندهنده حداقل تعداد افراد مورد نیاز در جهت مانایی فرایند انتشار است. هنگامی که جرم بحرانی بدست می‌آید، یک ممان حرکتی ایجاد می‌کند که حرکت انتشار را حفظ کرده و شکست انتشار را سخت می‌کند.

مفهوم جرم بحرانی (که همچنین سطح بحرانی[۳] یا نقطه اوج[۴] نامیده می‌شود) تقریباً یک موضوع فراگیر است و در همه پدیده‌ها وجود دارد. این امر چگونگی تأثیر زیاد رهبران عقیده بر رفتار دیگران را توضیح

[1] Source of the Diffusion
[2] Critical Mass
[3] Critical Level
[4] Tipping Point

می‌دهد. نتایج چنین تاثیری می‌تواند بعضاً از سطح فردی و سازمانی فراتر رفته و در سطح ملی و یا حتی بین‌المللی پدیدار گردد.

با این حال، چرا این اتفاق می‌افتد؟ یعنی چرا جرم بحرانی باعث نشو و نما (یا شکوفا شدن) ناگهانی پذیرنده‌ها می‌شود؟ پاسخ به این سوال از دو منظر یا رخنمون امکانپذیر است:

در رخنمون اول، به فرآیند از یک زاویه صرفاً کمّی نگاه می‌کنیم. هنگامی که به یک تعداد کافی از افرادِ خوش–ارتباط، ایده را قبول کردند، آنگاه گروههای دیگری از افراد در مواجهه با نوآوری قرار می‌گیرند، که پس از این گروهها، قاعدتاً مردم بیشتری در مواجهه با ایده، قرار می‌گیرند.

در رخنمون دوم، به این فرآیند بعنوان یک تغییر کیفی در سیستم نگاه می‌کنیم، یعنی (بعد از مدتی) با کاهش ناگهانی آستانه‌های برخی افراد موجه می‌شویم که آنهم به نوبه خود باعث می‌شود برخی دیگر از آنان احساس اعتماد به نفس کرده یا حتی مجبور شده و نهایتاً نوآوری را پذیرا باشند.

معیار مرکزیت بینابینی نوعاً با جرم بحرانی پیوند دارد. برگزیدن یا هدف قراردادن افرادی با مرکزیت بینابینی بالا در شبکه، یک استراتژی خوب برای معرفی یک نوآوری و موفقیت آن است.

فرآیند انتشار در مراحل اولیه خود، نیاز به کمک بیرونی به شیوه‌های گوناگون مثلاً در قالب یک پویش تبلیغاتی دارد. با این وجود، در مراحل بعدی، به ویژه زمانی که یک تعداد کافی از رهبران عقیده، نوآوری را می‌پذیرند، فرآیند انتشار می‌تواند بدون کمک خارجی، حرکت خود را حفظ کرده و حتی تسریع بخشد. اشاعه اجتماعی پس از آن زمان، انتشار گسترده و سریع را ضمانت می‌کند.

اگر انتشار نتواند به جرم بحرانی در زمان مشخصی، دست یابد، سطح نرخ پذیرش آن کاهش یافته و در نهایت، فرایند انتشار از بین می‌رود. در مقابل، انتشار ممکن است به جرم بحرانی خود برسد و پذیرش به صورت نمایی افزایش یابد تا به "نقطه اشباع"[1] برسد، یعنی نقطه‌ای که تقریباً همه کسانی که مواجهه را دریافت کرده‌اند، نوآوری را پذیرفته باشند. از آن لحظه، انتشار کاهش یافته و در نهایت از بین می‌رود.

تعیین دقیق لحظه‌ای که فرایند انتشار به جرم بحرانی می‌رسد، آسان نیست. ممکن است نیاز به اطلاعات دقیق در مورد اثرات برخی از رویدادها (مانند پویشهای رسانه‌ای و اشاعه اجتماعی) بر فرایند انتشار داشته باشیم. دو رویکرد جهت آگاهی از دستیابی به جرم بحرانی، توسط محققین ارائه شده است: اولین رویکرد با استفاده از یک قاعده سرانگشتی است. یعنی فرض می‌کند که یک پدیده خاص، زمانی اتفاق می‌افتد که نوآوری توسط ۱۶ درصد (یا بین ۲۰–۱۰ درصد) از همه افرادی که نهایتاً آنرا می‌پذیرند،

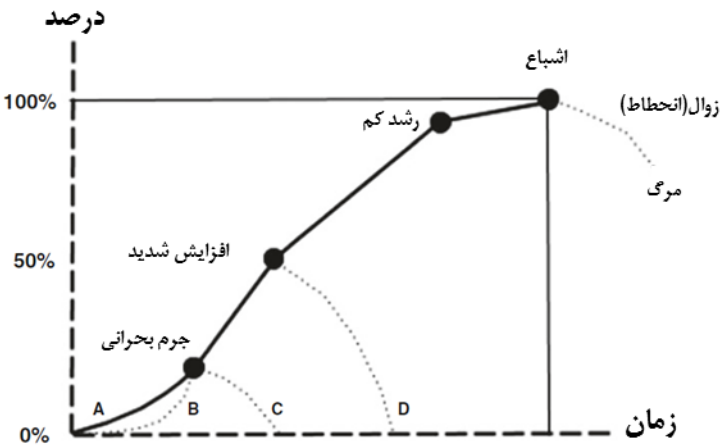
[1] Saturation Point

پذیرفته شود. رویکرد دوم فرض می‌کند که یک فرآیند انتشار زمانی به جرم بحرانی خود می‌رسد که اکثر افراد مرکزی آنرا پذیرفته باشند. در آن زمان، بسیاری از کنشگران شبکه در مواجهه با پذیرندگان قرار گرفته و بسیاری از آنها به آستانه‌های مواجهه خود دست یافته‌اند.

مثال

فرض کنید می‌خواهیم منحنی انتشار را برای معرفی یک پلت‌فرم رسانه اجتماعی جدید رسم کنیم. در این مثال، مدیران اجرایی یک وبسایت تلاش کردند ابتدا آنرا در یک بازار محلی معرفی کنند که بتدریج نام محصول مورد نظر بتواند در سراسر کشور گسترش یابد. به این دلیل آنها در آغاز کار، توجه خود را به یک اجتماع متراکم محلی و کوچک، معطوف کردند، اما بعداً آنرا برای برای پوشش همه اجتماعات کشور گسترش دادند (شکل ۴–۸).

مدل پذیرش نوآوری، که در شکل ۴–۸ نشان داده شده است، از جمعیتی با تعداد محدود شروع می‌کند. محور x، نشاندهنده زمان پذیرش نوآوری و محور y، نمایشگر آستانه‌ها است؛ یعنی درصد مخاطبین شبکه که نوآوری را پذیرفته‌اند. به طور مثال، زمانیکه تقریباً نیمی از زمان گذشته است، ۵۰ درصد اجتماع مورد نظر، نوآوری را پذیرفته‌اند. هرچند می‌توان مشاهده نمود که بسیاری از این افراد، آستانه کمتر و یا بیشتری از این مقدار دارند.



شکل ٤–٨ منحنی انتشار برای پلت‌فرم رسانه اجتماعی جدید

مسیر A، یک پذیرش سریع و کاملِ نوآوری توسط جمعیت را نشان می‌دهد. بعد از دستیابی به جرم بحرانی مورد نیاز در مرحله دوم، تعداد پذیرندگان به سرعت افزایش می‌یابد، اما در بخش آخر، کاهشی در

نرخ رشد (پذیرندگان) ایجاد می‌شود. پس از رسیدن به اشباع در مرحله نهایی، تعداد پذیرندگان کاهش یافته و مدل انتشار از بین می‌رود. منحنی (رشد جمعیت[1]) شاهد نقاط نوسانی[2] است یعنی زمانهایی که منحنی بشدت شتاب گرفته یا دفعتاً کند شده است. آنچه در شکل می‌بینیم مبین این واقعیت است که پلت‌فرم رسانه اجتماعی جدید، به‌ویژه، تحت‌تاثیر جرم بحرانی است، زیرا وقتی این نوآوری، توسط تعداد زیاد (و به حد کفایتی) از افراد پذیرفته شد، برای آنها (یعنی پذیرندگان) رفتن به سمت رسانه دیگر، سخت می‌شود. اگرچه منحنی پذیرش A، یک رشد طبیعی ساده پذیرش را نشان می‌دهد، اما به ما نمی‌گوید چرا بعضی از افراد، محصول را پذیرفتند؟ یا چرا برخی خیلی دیرتر از بقیه محصول را پذیرفتند؟ یا چرا برخی هرگز محصول را نپذیرفتند؟

سناریوهای دیگری نیز برای پذیرش این محصول محتمل‌اند. به طور مثال، مسیر B، الگویی مشابه مسیر A را نشان می‌دهد ولیکن دارای یک مرحله دیرکرد یا تاخیر نسبتاً طولانی‌تری است. در مسیر C، فرایند پذیرش در رسیدن به جرم بحرانی (که برای متقاعد کردن اکثریت مشتریان در مورد مطلوبیت محصول، ضروری است) ناکام مانده است. این در حالی است که مسیر D، علیرغم دستیابی به جرم بحرانی، به دلیل نامعلومی، دچار یک سقوط ناگهانی شده است.

انگیزه‌ها و محرکهای پذیرش نوآوری، قبل و بعد از رسیدن سیستم به جرم بحرانی، کاملاً متفاوت است، و به همین دلیل، فرایند پذیرش قبل از رسیدن به جرم بحرانی، ریسک بیشتری دارد.

انواع مختلفی از مدلهای (انتشار) برای مطالعه فرایند انتشار اطلاعات در شبکه‌های اجتماعی توسعه یافته‌اند؛ که از آن جمله می‌توان به نظریه بازیها[3]، همه‌گیری یا اپیدمی[4]، آستانه‌ای[5] و آبشاری[6] اشاره کرد. در ادامه، انتشار اطلاعات براساس مدلهای همه‌گیری را بررسی خواهیم کرد.

[1] Growth of Population
[2] Fluctuation Points
[3] Game Theoretical
[4] Epidemic
[5] Threshold
[6] Cascade

۸-۱۳ همه‌گیری

فرایند همه‌گیری (یا اپیدمی)، نحوه شیوع بیماریهایی همانند مالاریا، طاعون و یا ایدز است که به طور گسترده از طریق یک مکانیزم گسترش مانند تنفس، انتقال خون، نوشیدن و خوردن و یا غیر آن، درون جمعیتی از میزبانها[1] نظیر انسانها، حیوانات و گیاهان گسترش می‌یابد. اگرچه این عبارت عموماً در زمینه بیماریها و گسترش آنها در یک جمعیت مورد استفاده قرار می‌گیرد، اما می‌تواند برای توصیف اشاعه موضوعات اجتماعی در جامعه و یا پذیرش یک محصول مورد استفاده قرار گیرد.
برای فهم توان بالقوه یک فرایند اشاعه‌ای و یا اپیدمی لازم است که ما به طور کامل، فرآیند بیولوژیکی[2] در هر میزبان، فرآیند سیستم ایمنی، تعاملات میان افراد و ویژگیهای اجتماعی و فرهنگی را درک کنیم. بعلاوه از دیدگاه شبکه‌ای، نیاز به درک ساختار شبکه و اهمیت گره‌ها (افراد) در شبکه داریم. با مبنا قرار دادن ساختار شبکه، آنگاه دستیابی به بینش قابل‌توجهی در فهم نحوه گسترش بالقوه انتشار (بیماری) و اینکه آیا آن بیماری مسری، انتشار بیشتری خواهد داشت یا نه، امکانپذیر می‌شود.

- گره‌های بسیار تاثیرگذار در شبکه (یعنی هابها)، گسترش سریع یک بیماری در شبکه را ممکن می‌سازند. در مقابل، گره‌هایی با اتصال ضعیف یا گره‌های حاشیه‌ای، گسترش اطلاعات (یا بیماری) را کاهش داده و تنها به بخش کوچکی از شبکه، اجازه مواجهه با انتشار را می‌دهند.

- شبکه‌هایی که تعداد زیادی خوشه‌های محلی دارند (یعنی بسیار تکه تکه بوده و دارای چگالی پایینی هستند) بطور طبیعی انتشار را محدود کرده و در ایجاد ممان حرکتی، مشکل دارند. اما شبکه‌های چگال (متراکم) با شکاف اندک (یعنی با تعداد مهره‌ها مرزی یا پلهای محدود) سرایت بیماری را تسریع می‌بخشند.

انتقال یک بیماری در یک شبکه می‌تواند از طریق مداخله خارجی مانند اجتناب (احتراز و پرهیز)، واکسیناسیون یا انزوا (قرنطینه)[3] متوقف شود. چنین رویه‌هایی قویاً، قادر به کاهش و حتی بعضاً توقف گسترش بیماری هستند.

[1] Hosts
[2] Biological Process
[3] Isolation

۸-۱۴ مدلهای همه‌گیری

اخیراً مدلسازی همه‌گیری، یک حوزه پژوهشی فعال برای محققینی شده است که بر روی مدلهای "فرآیندهای پویای شبکه-پایه"[1] کار می‌کنند. همه‌گیر شدن بیماری مرگ سیاه[2] در قرن سیزدهم، طاعون بزرگ لندن[3]، بیماری آبله در قرن هفدهم و بیماریهای همه‌گیر اخیر چون اچ‌آی‌وی، ایدز، سارس[4] و آنفولانزای حاد پرندگان[5] انگیزه‌ای برای مطالعه همه‌گیری و معرفی و توسعه مدلهای آن شده است.
یک مدل همه‌گیری، روشی ساده برای تشریح نحوه انتقال بیماری مسری از طریق افراد است. مدلهای مختلفی برای مطالعه مکانیسمهای گسترش بیماریها، پیش‌بینی مسیر آینده شیوع بیماریهای مختلف جهان و ارزیابی راهبردها برای کنترل یک بیماری همه‌گیر، توسعه یافته‌اند. در بخش بعدی نگاهی به گروهی از مدلهای همه‌گیر سنتی (یعنی غیرشبکه‌ای[6]) می‌اندازیم.

۸-۱۵ مدلهای بخش‌بندی

مدلهای بخش‌بندی (یا کوپه‌بندی) در امر انتشار به دو دسته مدل بخش‌بندی قطعی[7] و بخش‌بندی تصادفی تقسیم می‌شوند. در مدل بخش‌بندی قطعی، افراد در هر جمعیت به زیر گروهها یا بخشهای مختلف تقسیم شده، بطوریکه هر کدام از بخشها یک مرحله خاص از این اپیدمی را نشان می‌دهد. این مدل برای مقابله با جمعیتهای بزرگ مانند آنچه در بیماری سل رخ می‌دهد، مفید است. این مدل افراد را براساس وضعیتشان در بیماری طبقه‌بندی می‌کند. هر فردی در یکی از این سه حالت است:

۱. سالم اما مستعد[8] (S): هنوز آلوده نشده، اما مستعد ابتلا به این بیماری است.
۲. آلوده، ابتلایافته یا مبتلا[1] (I): دارای بیماری مسری بود و به آن مبتلا شده است.

---
[1] Network-Based Dynamic Process Models
[2] Black Death Epidemic
[3] Great Plague of London
[4] SARS
[5] H5N1
[6] Nonnetwork
[7] Deterministic Compartmental Model
[8] Susceptible

۳. رهاشده یا بهبودیافته[۲] (R): دیگر بیماری مسری ندارد (انتقال‌دهنده بیماری نیست) و یا مستعد ابتلا به بیماری مسری نیست.

ترکیب حروف اول این سه واژه یا حالت اصلی، چندین مدل برای شناسایی فرایند انتشار بیماری ایجاد می‌کند که موارد متداول آنها عبارتند از: SIR ، SIS، SI، و $SIRS$. همه این مدلها در اصل برای مطالعه نحوه شیوع یا گسترش بیماریهای مسری، در یک جمعیت، طراحی شده‌اند. همچنین آنها چرخه بیماری در یک میزبان را توصیف کرده و از ترکیبی از عبارات (S، I، R) برای مشخص کردن هر مرحله استفاده می‌کنند. در بخش بعدی، بر روی مدل سار (س.ا.ر. یا SIR) تمرکز خواهیم کرد.

۸-۱۶ مدل سار (SIR)

مدل **سار** (SIR) بمعنای مدل **سلامت-ابتلاء-رها**[۳] است که اولین بار در سال ۱۹۳۲ توسط کرمک[۴] و مک‌کندریک[۵] ارائه شد. این مدل، مراحل یک بیماری را توصیف می‌کند که در آن، فرد سالمی که مستعد پذیرش بیماری است، آلوده یا بیمار شده، سپس رهایی یا بهبود یافته، و در نهایت نسبت به بیماری ایمن می‌شود. این مدل یکی از متداولترین مدلهای همه‌گیری زمان پیوسته[۶] است.

بیماریهایی مانند فلج اطفال[۷]، سرخک[۸]، اوریون[۹] و سرخجه[۱۰] چنین چرخه و فرایندی دارند. در مرحله اول، اول، افراد سالم بتدریج مستعد ابتلا به بیمار می‌شوند. در مرحله دوم، ممکن است در تماس با افراد بیمار،

---

[1] Infected
[2] Recovered
[3] Susceptible-Infected-Recovered (SIR) Model
ما در این کتاب واژه "سار" را بعنوان معادل SIR پیشنهاد داده‌ایم. واژه "سار" سرواژه کلمات ترکیب "سالم-ابتلایافته-رهاشده" است. برای این سه کلمه انگلیسی، ما همچنین سه‌گانهٔ "مستعد-آلوده-شفایافته" برای دچار شدن به بیماری (در حوزه سلامت) و سه‌گانه "سازگار-اقناع‌شده- راه‌افتاده" برای پذیرش ایده (برای حوزه اجتماعی) را استفاده کرده‌ایم.
[4] Kermack
[5] McKendrick
[6] Continuous-Time Epidemic Models
[7] Polio
[8] Measles
[9] Mumps
[10] Rubella

مبتلا به بیماری شوند، همچنین در این مرحله می‌توانند دیگران را نیز بیمار کنند. در مرحله سوم، آنها بهبودی یافته و دیگر مستعد پذیرش بیماری نیستند.

مدل سار (SIR) بطور موفقیت‌آمیزی برای شناسایی اثر ساختار شبکه بر فرایند انتشار، در شبکه‌های جهان واقعی استفاده شده است. از این مدل برای مطالعهٔ گسترش اطلاعات در چارچوبهای بهم پیوسته و یا هم‌پیوند[1] نیز بهره گرفته‌اند. بعلاوه برای کشف موضوعات و عناوینی که برای گسترش خشونت[2]، تندروی، افراط‌گرایی، و نژادپرستی از طریق رسانه اجتماعی پخش می‌شود، از این مدل استفاده شده است.

این مدل شامل سه مولفه یا سازند است: گره‌هایی که به یکدیگر متصل‌اند، مسیری که بیماری در آن گسترش می‌یابد و طریقی که این گره‌ها مبتلا گشته و سپس بهبود پیدا می‌کنند.

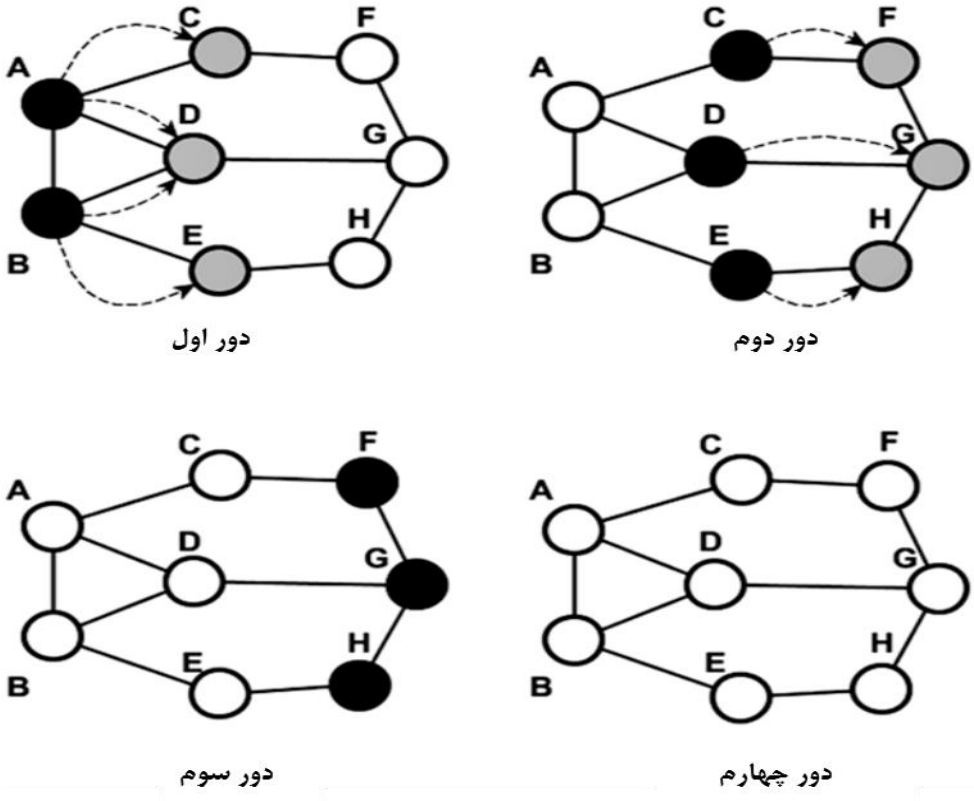
در هر مرحله زمانی از این فرایند، فقط گره‌هایی که مبتلا (به بیماری) هستند، می‌توانند با احتمال β گره‌های مجاورشان، را که در وضعیت سلامت (یا مستعد بیماری) هستند، بیمار کنند. سپس در مرحله بعد، افرادی که در مرحله پیش، بیمار (مبتلا) بودند، با احتمال α بهبود یا رهایی یافته و دیگر به آن بیماری دچار نشده و یا دیگران را آلوده نمی‌کنند. شکل ۵-۸، مدل سار (SIR) را نشان می‌دهد.

شکل ۵-۸ نمایش مدل سار (SIR) و سه مرحله آن

مثال

شکل ۶-۸، انتشار همه‌گیری را بطور فرضی، در یک شبکه کوچک با هشت گره نشان می‌دهد. این گره‌ها با پیوندهای متقارن به یکدیگر متصلند، که اجازه گسترش بیماری مسری از گره‌ای به گرهٔ دیگر را می‌دهد. گره‌هایی که مبتلا گشته‌اند می‌توانند دیگران را نیز مبتلا کنند، اما نهایتاً بهبود می‌یابند. فرض می‌کنیم که وجود یک همسایه مبتلا برای ابتلا یافتن فرد سالم (مستعد بیماری) کافی است. این شکل، چهار دورهٔ انتشار را نشان می‌دهد که با دو گره مبتلا، یعنی گره A و B، شروع شده است. هر دور یا مرحله، وضعیت شبکه در یک مدت زمانی مشخص را نشان می‌دهد.

[1] Conjoint Frameworks
[2] Spread of Violence



شکل ٦-٨ انتشار فرضی برای یک فرایند همه‌گیری

۱) گره‌های A و B در دور اول، از اطلاعات مورد نظر، آگاهی دارند و می‌توانند اطلاعات را به همسایگانشان انتقال دهند (به بیان دیگر، در مبحث همه‌گیری، آنها ابتلاء یافته و می‌توانند مبتلا کنند). ولیکن، آنها در دور بعد بهبود می‌یابند. هنگامی که آنها بهبودی می‌یابند، اطلاعات (یا بیماری) را فراموش یا رها می‌کنند یا دیگر علاقه‌ای به آن ندارند. همچنین آنها دیگر نمی‌توانند مجدداً مبتلا به این بیماری شوند.

۲) گره‌های C، D و E هنوز آگاهی از اطلاعات (یا این بیماری را) ندارند. آنها اگرچه هنوز مبتلا نگشته‌اند ولی بدلیل تماس با همسایگانِ مبتلا در دور اول، مستعد پذیرش هستند. این مطلب بدان معنا نیست، که همه‌گیری، اجتناب‌ناپذیر است زیرا عوامل بسیاری در مبتلا شدن یا نشدن یک فردِ

در مواجهه با بیماری وجود دارد. این گره‌ها در دور دوم مبتلا گشته و در دور سوم بهبود یافته و هرگز مجدداً مبتلا یا ناقل این بیماری نمی‌شوند.

۳) گره‌های G، F و H در دور اول، نه مبتلا و نه مستعد هستند. با این وجود، در دور دوم مستعد و در دور سوم مبتلا گشته و نهایتاً در دور چهارم بهبود می‌یابند.

۱۷-۸ خواص مدل سار (SIR)

برخی از خواص مدل بر اساس کارکرد سار (SIR)، بشرح زیر است:

- در مدل سار، میزان و سطح مستعد بودن یک فرد برای پذیرش بیماری و اینکه چه چیزی باعث افزایش یا کاهش احتمال ابتلای وی به آن بیماری می‌شود، مورد توجه قرار نمی‌گیرد. این بدان معناست که یک فرد یا کاملاً مستعد پذیرش بیماری هست یا اصلا نیست.

- برخلاف برخی مدلهای تصادفی، فرایندهای این مدل، قطعی هستند. به این معنا که مدل فرض می‌کند هر شخصِ مستعد وقتی در مواجهه با فرد مبتلا و بیمار قرار بگیرد، حتما مبتلا می‌شود. این فرض در دنیای خارج از ذهن، خیلی واقعی نیست، چرا که برخی افراد در مقایسه با دیگران که سریع بیمار می‌شوند، مقاومت بیشتری نسبت به بیماریها دارند، همچنین، بیماریها با نرخ متفاوتی گسترش می‌یابند و افراد سطوح مواجهه متفاوتی با بیماریها دارند.

- مدل سار (SIR) فرض می‌کند که هر فرد پس از رهایی از بیماری، دیگر مستعد پذیرش بیماری نمی‌شود. در یک مدل دیگر تحت عنوان مدل سارس ($SIRS$) یا سازگار-آلوده-رهایافته-سازگار[1]، فرض می‌کنیم فردی که سالم (سازگار پذیرش بیماری) است، ابتلاء یافته، رها گشته، مدتی از ایمنی لذت می‌برد و مجدد مستعد (یا سازگار پذیرش بیماری) می‌شود.

[1] Susceptible-Infected-Recover-Susceptible Model

- در مدل سار (*SIR*)، هنگامی که یک فرد از بیماری بهبود می‌یابد، نسبت به آن بیماری ایمن می‌شود. در مقابل در مدل ساس (*SIS*) یعنی سازگار–ابتلاء یافته–سازگار[1]، فرد مبتلا، بعد از ابتلای به بیماری، ایمن نمی‌گردد. از طرف دیگر، در مدل سا (*SI*) سازگار–آلوده[2]، که مربوط به بیماریهای مهلک و کشنده است، فرد مبتلا، هرگز به حالت بهبود یافته یا مستعد برنمی‌گردد.

مثال

در این مثال، نحوه مدلسازی گسترش همه‌گیری در یک شبکه را بررسی خواهیم کرد. مثال پیش رو یکی از مثالهای خوب این کتاب است.

در اینجا، ساختار شبکه بر سرایت بیماری اثرگذار است؛ به طوریکه گره‌هایی با یک درجه بالا، با پتانسیل بیشتری، همسایگانشان را مبتلا کرده و نیز مبتلا می‌شوند.

```
In:  # NetworkX for network representation
     import networkx as nx
     #Some handy mathematical functions
     import math
     #Random numbers from numpy
     import numpy as np
     import numpy.random as rnd
     #Important elements of matplotlib
     import matplotlib.pyplot as plt
     import matplotlib.colors as clr
     import matplotlib.cm as cmx
     #Plot all graphics in-line in the notebook
     %matplotlib inline

In:  SPREADING_SUSCEPTIBLE = 'S'
     SPREADING_INFECTED = 'I'
     SPREADING_RECOVERED = 'R'

     def spreading_init( g ):
         """Initialise all node in the graph to be susceptible""".
      for i in g.node.keys:()
          er.node[i]['state'] = SPREADING_SUSCEPTIBLE

In:  def spreading_seed( g، pSick ):
         """Inject a random proportion of nodes in the graph""".
        for i in g.node.keys:()
```

[1] Susceptible-Infected-Susceptible Model
[2] Susceptible-Infected Model

```
            if(rnd.random() <= pSick):
                er.node[i]['state'] = SPREADING_INFECTED
```

اکنون تابعی تعریف می‌کنیم، که با یک α و β معین، یک تابع مدل سار (SIR) را ایجاد می‌کند. نتیجه این تابع را در بخشهای بعدی فراخوانی می‌کنیم.

```
In:  def spreading_make_sir_model( pInfect، pRecover ):
         """Return an SIR model function for given infection and
     recovery probabilities."""

        # model (local rule) function
        def model( g، i ):
            if g.node[i]['state'] == SPREADING_INFECTED:
                # infect susceptible neighbours with probability
     pInfect
                for m in g.neighbors(i):
                    if g.node[m]['state'] == SPREADING_SUSCEPTIBLE:
                        if rnd.random() <= pInfect:
                            g.node[m]['state'] = SPREADING_INFECTED

                # recover with probability pRecover
                if rnd.random() <= pRecover:
                    g.node[i]['state'] = SPREADING_RECOVERED
     return model
```

اکنون، فرایند انتشار را بعنوان یک تابع تعریف می‌کنیم. این تابع، یک گراف و یک تابع مدل را دریافت کرده و مدل را بر تمامی گره‌های گراف اعمال می‌نماید.

```
In:  def spreading_step( g، model ):
         """Run a single step of the model over the graph."""
         for i in g.node.keys():
             model(g، i)
```

همچنین، تابعی تعریف می‌کنیم که همین مدل (پویای تکراری) را با یک تعداد تکرار مشخص، اعمال می‌کند.

```
In:  def spreading_run( g، model، iter ):
         """Run a number of iterations of the model over the
graph."""
         for i in xrange(iter):
```

```
            spreading_step(g، model)
```

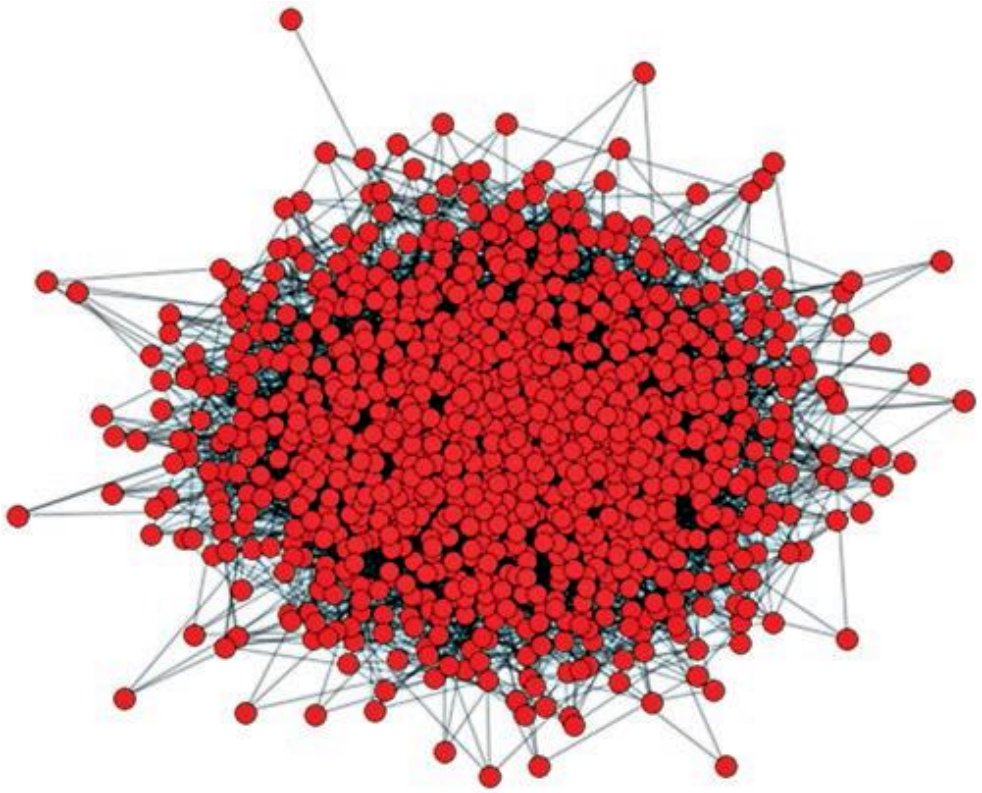
اکنون، مدل را ایجاد می‌کنیم. برای اینکار از مدل اِردوش رینی ER[1]، استفاده می‌نماییم.

```
In:  n = 1000
     er = nx.erdos_renyi_graph(n، 0.01)
```

با استفاده از طرح چیدمان بهار، مدل را رسم می‌کنیم (شکل ۷–۸).

```
In:  fig = plt.figure(figsize=(10، 8))

     # some visual styling for the figure
     ax = fig.gca()
     ax.grid(False) # no grid
     ax.get_xaxis().set_ticks([]) # no ticks on the axes
     ax.get_yaxis().set_ticks([])
     # run the spring layout algorithm over the network
     pos = nx.spring_layout(er، iterations = 50، k = 2/math.sqrt(n))

     # draw the network using the computed positions for the nodes
     nx.draw_networkx_edges(er، pos، width = 1، alpha = 0.4)
     nx.draw_networkx_nodes(er، pos، node_size=100، alpha = 1،
     linewidths = 0.5)
     plt.show()
Out
```

[1] Erdos-Renyi ER Model



شکل ۷-۸ گسترش اپیدمی (همه‌گیری) در یک شبکه

سپس، مدل را با یک نسبت مشخص از افراد مبتلا و بقیه مستعدِ پذیرش، شبکه می‌کنیم:

```
In:  # initialise with 5% sick people
     spreading_init(er)
     spreading_seed(er، 0.05)
```

در نهایت، پویایی شبکه را ایجاد می‌کنیم:

```
In:  # SIR model with 30% infection rate and 10% recovery rate
     model = spreading_make_sir_model(0.3، 0.05)
```

حال مدل را اجرا می‌کنیم:

```
In:  # run SIR model dynamics over the network
     spreading_run(er، model، 100)
```

همچنین، می‌توانیم، درصد افراد مبتلا شده در مدل را نیز گزارش نماییم:

```
In:  infected = [ v for (v، attr) in er.nodes(data = True) if
     attr['state'] == SPREADING_RECOVERED ]
     print float(len(infected)) / n
Out: 0.994
```

فصل نهم

کاربرد تجزیه و تحلیل شبکه در نگاشت گفتگوهای سیاسی برخط

درک روابط میان "کاربران آنلاین" به هنگام انجام مباحثه‌ها و مناظره‌های سیاسی، کمک خوبی به ما در فهم شباهتها و تفاوتهای "تعاملات آفلاین" آنها می‌کند. اجتماعات برخط، عموماً، برای گویایی بیشتر، کاربران خود را به شدت همگون و یکسان نشان می‌دهند، اما گفتگوهای سیاسی برخط، لزوماً مانع بروز تنوع آراء و عقاید نمی‌شوند.

تالار گفتگوی "جنبش پنج ستاره"[1] یک تالار گفتگو یا انجمن مباحثه[2] معروف در ایتالیا است. بررسی گفتگوها در این تالار، بدلیل تنوع جهت‌گیریهای سیاسی شرکت‌کنندگان و گستره وسیع موضوعات، یک

[1] Five Star Movement

[2] تالار گفتگو (Discussion Forum) به برنامه‌های مبتنی بر وب گفته می‌شود که برای نگهداری بحثها و نوشته‌های کاربرهای یک وبگاه به کار می‌روند. در تالار گفتگو معمولاً افراد سؤالات خود را مطرح می‌کنند و سایر کاربران به این سؤالات پاسخ می‌دهند. علاوه بر این تالارهای گفتگو به جز سؤال پرسیدن می‌توانند واسطه بحثی در هر زمینه باشند و افراد می‌توانند نظریات و عقیده‌های خود را برای دیگران بیان کنند.

مطالعه موردی جذاب محسوب می‌شود. لذا در این فصل، به مدد روشهای تجزیه و تحلیل شبکه که در فصول پیش بحث شد، قصد داریم جزئیات رفتار کاربران این تالار را در دو سطح که عبارتند از سطح گره، یعنی تحلیل فردانه (شامل کاربران و پیامها) و سطح شبکه، یعنی تحلیل جمعانه (شامل سازندها و اجتماعات) بررسی کرده تا ویژگیهای شبکه مذکور و رفتار کنشگران آن را توضیح دهیم.

۹-۱ تاریخچه جنبش سیاسی پنج ستاره

جنبش پنج ستاره[1]، یک طیف یا جناح گسترده سیاسی در ایتالیا است که در ۴ اکتبر سال ۲۰۰۹، توسط یک بازیگر، وبلاگ‌نویس و فعال سیاسی به نام جوزپه پیرو بِپِه گریلو[2] تاسیس شد. شکل‌گیری این جنبش، که از نظر اعضای خودش یک حزب نیست، ریشه در موفقیت چشمگیر و تصاعدی وبلاگ[3] او دارد که در سال ۲۰۰۵ آنرا راه‌اندازی نمود. در همین سال، با سازماندهی کاربران وبلاگ، عملاً بحثها و گفتگوها (مباحثه‌ها) از فضای مجازی به جلسات عمومی و میتینگهای سیاسی کشیده شد. در واقع، این اولین گام در راستای موفقیت وبلاگ و پیدایش سیاسی آن بود.

در سال ۲۰۰۷، گریلو، اولین *V-Day*[4] را سازماندهی کرد. *V-Day* در حقیقت، روزی است که بیش از سیصد هزار ایتالیایی در بیشتر از ۲۰۰ شهر مختلف با پیشنهاد گریلو، به خیابانها آمدند و بیانیه‌ای را جهت پاکسازی پارلمان ایتالیا و توقیف نمایندگان مجرم امضا نمودند. آنان همچنین خواهان محدود کردن نمایندگان به حداکثر دو دوره و انتخاب مستقیم قانون‌گذاران توسط مردم شدند. در چنین روزی، بپه گریلو، خواسته یا ناخواسته رهبر توده بزرگی از شهروندان ناراضی گردید که از عملکرد بد سیاستمداران کشور به تنگ آمده بودند و بدینسان جنبش وی ایجاد شد. حرف بزرگ *V* در نام این جنبش نیز به دلیل و نشانهٔ اهمیت این روز، یعنی (*V-Day*) است.

پس از برگزاری یک نشست (میتینگ) ملی که در اوایل سال ۲۰۰۹ برگزار شد، این جنبش در اواسط سال ۲۰۰۹ برای اولین حضور در انتخابات محلی دست بکار شد و لذا رسماً در پایان آن سال تشکیل شد.

---

[1] Movimento 5 Stelle (M5S) عبارت "جنبش پنج ستاره" به زبان ایتالیایی می‌شود:

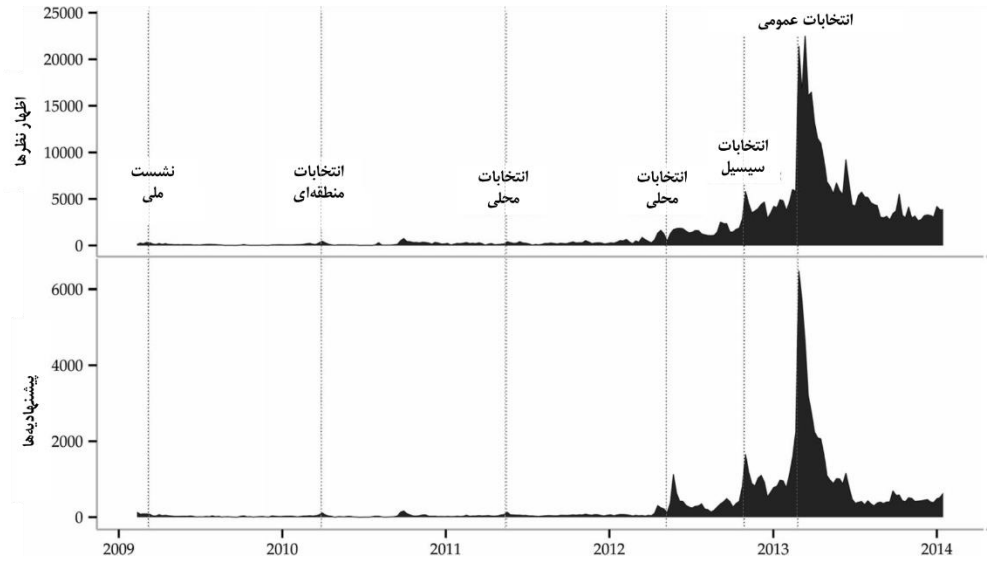
[2] Giuseppe Piero Beppe Grillo

[3] www.beppegrillo.it

[4] Vaffanculo Day, V-Day (این عبارت به زبان ایتالیایی است که اصطلاحاً یعنی: دمار از روزگارمان درآورد!)

بعد از آن، یعنی از سال ۲۰۱۰، جنبش پنج ستاره، در انتخابات برگزار شده در سطوح مختلف مانند: منطقه‌ای (۲۰۱۰)، محلی و منطقه‌ای (۲۰۱۱)، محلی و منطقه‌ای (۲۰۱۲) و عمومی (۲۰۱۳)، رقابت کرده است.

مشارکت در انتخابات، همزمان با افزایش توجه به جنبش و تالار گفتگو آنلاینِ آن بوده است (شکل۱-۹).



شکل ۱-۹ فعالیت هفتگی تالار گفتگو

جدول ۱-۹، تغیرات الگوی رای‌دهنگان ایتالیا را در طی سالهای ۲۰۰۸ تا ۲۰۱۳ نشان می‌دهد. جنبش پنج ستاره موفق شده است رای‌دهندگان بسیاری را از همه طیفهای سیاسی مختلف به سمت خود جذب کند. رای‌دهندگانی از احزاب چپِ چپ تا راستِ راست، یعنی از **جناحهای چپ تندرو** شامل: حزب چپگرا، *رنگین‌کمان*، چپ *منتقد*، حزب *کمونیست کارگران*، و **چپ میانه** شامل: حزب *دموکرات* (*Pd*)، حزب *ارزشهای ایتالیا* (*Idv*)، تا احزاب **راست میانه** شامل: *اتحاد دموکرات مسیحی* (*Udc*)، حزب *خلق آزاد* (*Pdi*)، جنبش *استقلال* (*Mpa*)، و **احزاب راست** شامل: *پیوند شمال* (*Lega*)، و نهایتاً تا **جناحهای راست تندرو** شامل: *راستگرا و نیروی جدید*.

جدول ۱-۹ درصد احزاب رای دهنده به حزب پنج ستاره در انتخابات عمومی سال ۲۰۱۳ در شهرهای بزرگ

	چپ تندرو	*Idv*	*Pd*	*Udc*	*Pdi*	*Mpa*	*Lega*	راست تندرو	رای ندادند	دیگران
تورین	۶	۲۰	۳۷				۲	۱۰	۲۵	
میلان	۱۰	۱۳	۱۳	۴			۲۹	۹	۲۰	۲
بولونیا	۵	۱۲	۴۸		۴		۸	۱۲	۱۰	۱
فلورانس	۵	۲	۵۸		۵		۲	۵	۲۲	۱
رم	۷	۹	۱۱		۲۶			۱۰	۳۵	۲
ناپل	۸	۹	۴۴	۳	۲۶			۲	۷	۱
کاتانیا		۶	۱۸	۱۱		۳۴		۲	۲۷	۲

۹-۲ آمار تالار گفتگو

در این مطالعه، از یک مجموعه داده شامل ۸۴۲۰۳ نفر کاربر، حدود ۸۶۹۴۳ مضمون[۱] مربوط به پستهای آغازین[۲] (منتشره توسط اعضای ثبت‌نامی) و ۴۶۱۲۹۷ اظهار نظر[۳] استفاده شده است. در این تالار، هر

[۱] ما از واژه "مضمون" بعنوان معادل کلمه Thread استفاده کرده‌ایم که اشاره به محتوا، مفاد و یا فحوای پیامهای ارسالی در تالار دارد. همچنین از واژه "مقوله" بعنوان معادل کلمه Theme بهره برده‌ایم که اشاره به تجمیع و ادغام چند مضمون برای تشکیل یک مفهوم ترکیبی دارد. ضمناً در این فصل از دو واژه "مباحثه" (Discussion) و "مناظره" (Debate) استفاده شده تا دو نوع فرایند گفتگو در تالار را نشان دهد. در کنار اینها، دو واژه "پیشنهادیه" (Proposal) و "اظهارنظر" (Commemt) نشاندهنده دو نقش اصلی اعضا و کاربران است، یعنی یا آنها مضمونی را پیشنهاد داده و یا نسبت به مضامین موجود، اظهارنظری داشته‌اند. همچنین کابران تالار یا ثبت‌نام کرده‌اند (اعضا) و یا ثبت‌نام نکرده‌اند که کاربر آزاد نامیده می‌شوند. بعلاوه منظور از مضمون در این تالار، پیشنهادیه‌ای است که توسط یک عضو تالار پست شده و آمادهٔ دریافت نظرات کاربران است. با یک مثال همه این واژه‌ها را در کنار هم قرار می‌دهیم: در مناظره‌ای که پیرامون مقوله مهاجرت صورت گرفته است، ۲۰ مضمون مختلف توسط برخی اعضا پیشنهاد شده است که بیش از صدها نظر از سوی کاربران در مورد آن اظهار شده است.

[2] Opening Post

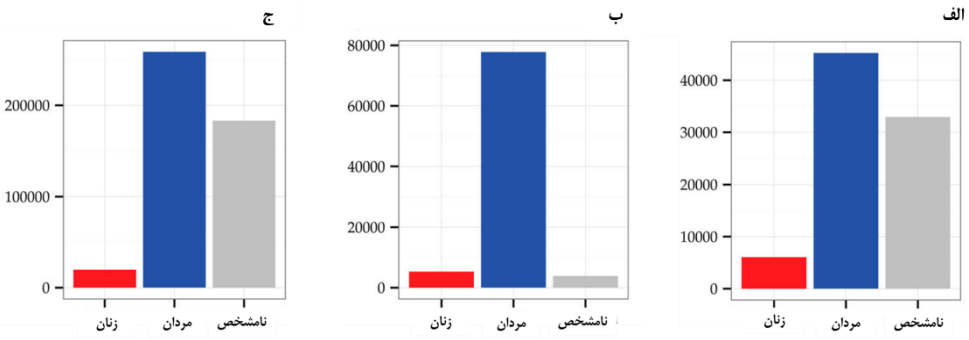
[3] Comments

"پست آغازین" را "پیشنهادیه"[1] می‌نامند. منظور از پست آغازین، پیامیست که از طرف اعضای تالار (کابران ثبت‌نامی) منتشر شده و باعث آغاز بحث و گفتگو در مورد یک مضمونِ پیشنهادی می‌شود. اظهارنظرها و پیشنهادیه‌های منتشر شده از تاریخ ۳۱ ژانویه ۲۰۱۴ به بعد، در این مطالعه در نظر گرفته نشده‌اند.

شکل ۱–۹ پیشنهادیه‌ها و اظهارنظرهای منتشر شده هفتگی، از تاریخ ۱۱ فوریه سال ۲۰۰۹ تا ۳۱ ژانویه سال ۲۰۱۴ را نشان می‌دهد. در این شکل، همبستگی زیاد میان "میزان شرکت در تالار گفتگو" و "رویدادهای انتخاباتی" توجه را جلب می‌کند. همه انتخابات با یک اوج محلی در سریهای زمانی منطبق هستند و استثنا قابل‌توجه، انتخابات سال ۲۰۱۲ است که در آن، *اوج سری زمانی*، بلافاصله *انتخابات* را "دنبال" می‌کند. انتخابات عمومی سال ۲۰۱۳، بعنوان بیشینه این سری زمانی، با بیش از ۲۰،۰۰۰ اظهارنظر هفتگی همزمان شده است. مناظره[2] در مورد "امکان مشارکت در یک دولت ائتلافی" پس از نتایج مثبت غیرمنتظره انتخابات ۲۰۱۳ قوت گرفته است. همانگونه که در شکل ۱–۹ می‌توان دید حجم تبادل نظر در روز ۲۵ فوریه ۲۰۱۳ (روز انتخابات) اوج گرفته، سپس برای چند روز متعادل گشته و بعد از آن، مجدداً، در ۱ مارس با انتشار پستی مبنی بر "مخالفت با مشارکتِ جنبش پنج ستاره در دولت ائتلافی" تعداد نظرات، به سرعت، به ۱۲۶۰۶ نظر افزایش یافته است.

با دقت در نام کاربران و مراجعه به فرهنگ نامها در زبان ایتالیایی، امکان تشخیص جنسیت اکثر افرادی که نام کوچک خود را اعلام کرده‌اند، وجود دارد. نامهای مربوط به ۳۰۹۲۱ عضو جنبش پنج ستاره را می‌توان هویت واقعی آن کاربران دانست، زیرا ثبت نام اعضا با ارائه کپی کارت شناسایی انجام می‌شود. البته سایر کاربران با هر نامی می‌توانند نظر خود را منتشر کنند. با این حساب، جنسیت ۵۱۲۶۷ نفر از میان ۸۴۲۰۳ کاربر، حدس زده شده است. در میان این تعداد، تقریبا ۱۲ درصد نامها، زنانه و ۸۸ درصد نامها، مردانه‌اند. در شکل ۲–۹ نسبت کاربرانی که جنسیت آنان مشخص شده، را مشاهده می‌کنید. شکل ۲–۹ الف، توزیع جنسیت کاربران تالار گفتگو را نشان می‌دهد. شکل ۲–۹ ب، توزیع جنسیت ارائه‌دهندگان پیشنهادیه (یا مضمون جدید) را نمایش داده و شکل ۲–۹ ج، نیز مربوط به توزیع جنسیت نظردهندگان است.

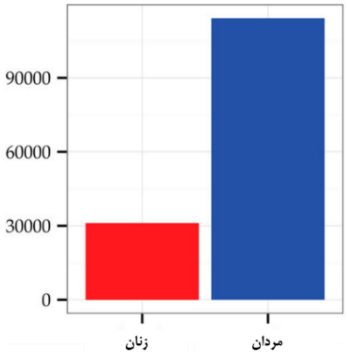
[1] Proposals
[2] Debate



شکل ۲-۹ توزیع جنسیت

اگرچه عدم تعادل جنسیت شرکت‌کنندگان در گفتگوهای سیاسی ممکن است ریشه‌های متفاوتی داشته باشد، اما به نظر می‌رسد که تعداد بسیار کم شرکت‌کنندگان زن در تضاد با ایده‌ای است که می‌گوید: "بحثهای اینترنتی عموماً بیش از مواجهه‌های رو در رو، طرفدار مساوات و برابری هستند"[۱]. به زبان دیگر، آمار تالار گفتگو، ایده فوق را تایید نمی‌کند.

همچنین، برخلاف ایده مساوات مطرح شده در فوق، وضعیت توزیع جنسیتی تالار تقریباً همانند وضعیت جنسیت مدیران دولتی ایتالیاست که در نمودار شکل ۳-۹ نشان داده شده است. براساس این نمودار، از میان ۱۴۵۲۵۶ مدیر انتخاب شده برای ادارات در سطوح محلی، استانی و منطقه‌ای در سال ۲۰۱۳، تنها ۲۱ ٪ زن بوده‌اند.



شکل ۳-۹ مدیران انتخاب شده در ایتالیا

[۱] این ایده در کتاب " دموکراسی ارتباطی: مشورت آنلاین و جریان ارتباطات سیاسی"، مطرح شده است.

اثر تقریباً خنثای فناوریهای ارتباطی بر جنسیت و عدم تعادل وضعیت آن در مشارکت سیاسی، توسط دو مطالعه تجربی دیگر (دالبرگ[1] در سال ۲۰۰۱ و مانویر اسمیت[2] در سال ۲۰۱۲) تایید شده است. هر دو مطالعه نشان‌دهنده اثر ناچیز مناظره‌ها و مباحثه‌های اینترنتی در بهبود توازن جنسیتی هستند.

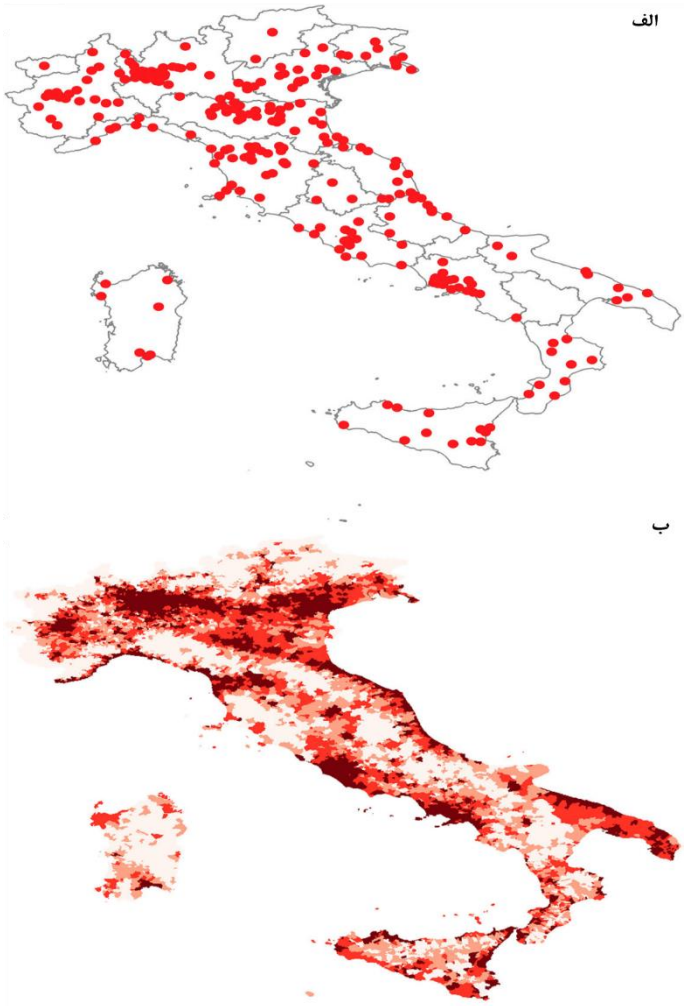
مقایسه "توزیع جغرافیایی کاربران برخط" با "تراکم جمعیت شهر آنان" و در نتیجه شناسایی مناطقی که به نسبت جمعیتشان، کاربران بسیاری دارند، نیز جالب است.

کاربرانی که در این تالار گفتگو ثبت‌نام کرده‌اند امکان نشان دادن موقعیت جغرافیایی خود را دارند. از میان ۲۸۲۹۸ کاربر که موقعیتشان پرسیده شده است، موقعیت مکانی ۳۶۴ نفر به دست آمده است (یعنی ۲۷۸۷۱ نفر عملاً هیچ موقعیتی را مشخص نکرده‌اند و ۶۳ کاربر یک مکان مبهم یا مکانی خارج از ایتالیا را نشان داده‌اند).

این واقعیت که تعداد بسیار کمی از کاربران، موقعیت مکانیشان را اعلام کردند تعجب‌آور نیست؛ چراکه، کاربران اینترنت به طور کلی از پر کردن بخشهای کم اهمیت فرم ثبت‌نام خودداری می‌کنند. با این حال، این امر دلیلی برای عدم تطابق تراکم جمعیت و تراکم موقعیت کاربران نمی‌باشد (به بیان دیگر، دلیلی برای بی‌اعتباری داده‌های مربوط به موقعیت جغرافیایی کاربران نیست).

با مقایسه توزیع موقعیتهای کاربران (شکل ۴-۹ الف) با توزیع تراکم جمعیت ساکنان (شکل۴-۹ ب)، به نظر می‌رسد که کاربران متناسب با جمعیت ساکن، توزیع شده‌اند و هیچ منطقه‌ای بیش از جمعیت خود، کاربر ندارد.

[1] Dahlberg
[2] Monnoyer-Smith



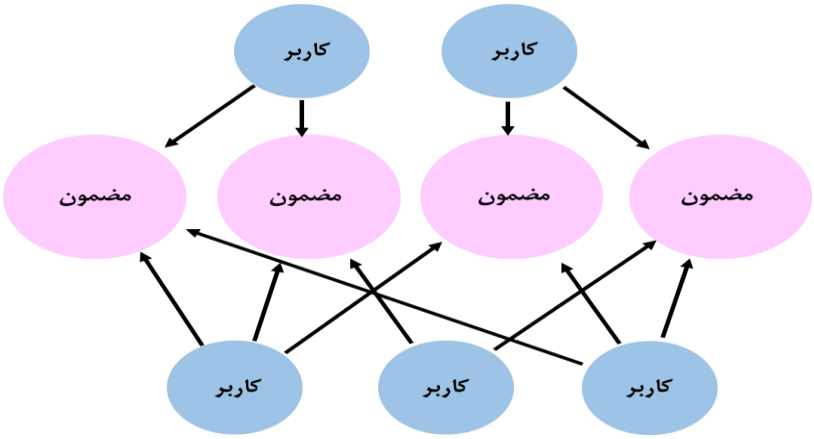
شکل ٤-٩ موقعیت کاربران، الف- کاربران جنبش پنج ستاره ب- توزیع جمعیت

وضعیت این انجمن به لحاظ نوع موضوعات مناظره‌شده (براساس مقوله‌هایی که احتمالاً توسط کاربران انتخاب شده‌اند) کاملاً ناهمگن به نظر می‌رسد. جدا از موضوع "تالار گفتگو" (که به نظر می‌رسد یک برچسب کلی برای همه عناوین مباحث است) مقولات مختلف دیگری مورد بحث قرار می‌گیرند که عبارتند از: اشتغال‌زایی (۱۴,۷ ٪ از کل تعداد مضامین)، اتصال اینترنت (۱۳,۳ ٪)، خدمات جوانان (۶,۹ ٪)، توسعه محلی (۶,۳ ٪)، تجارت (۴,۹ ٪)، مدیریت آب (۳. ۸ ٪) و تولید و ذخیره انرژی (۳,۸ ٪).

۹-۳ تجزیه و تحلیل شبکه

۹-۳-۱ شبکه دوحالته: مضامین و کاربران

تالار گفتگو را در اینجا، یک شبکه دو–حالته درنظر می‌گیریم، که در آن، گره‌ها (یا راسها) دو نوع هستند: مضمون و کاربر. این شبکه جهتدار است؛ چراکه، یالها در یک جهت بوده و اصطلاحاً یک طرفه هستند. در واقع، یالها از کاربر به مضمون کشیده می‌شوند. یک کاربر می‌تواند با یک مضمون پیوند داشته باشد؛ زیرا وی یا آن مضمون را شروع نموده است و یا بعداً در مورد آن اظهارنظر می‌کند. در شکل ۵–۹ این شبکه دو–حالته ترسیم شده است. در این مرحله، تمایزی میان خَلق مضمون و یا نظردهی در مورد آن وجود ندارد.



شکل ۵–۹ شبکه جهتدار دو–حالته

"توزیع درجه‌ای شبکه" به شدت به "توزیع قانون–قدرت" از منظر مضمون (درجه ورودی) و کاربران (درجه خروجی) نزدیک است. مضامین اندکی، اظهارنظرات زیادی را به خود جذب می‌کنند؛ ولی اکثر مضامین پیشنهادی یا پیشنهادیه‌ها، تعداد اندکی (یا یک) اظهارنظر دارند. به‌طور متقارن، تعداد بسیار اندکی از کاربران، عملاً پیشنهادیه‌ها و اظهارنظرهای فراوانی پست می‌کنند؛ ولی اکثر کاربران، تعداد کمی اظهارنظر و پیشنهادیه، پست می‌کنند.

شبکه این تالار گفتگو تقریباً، نزدیک به یک شبکه با مقیاس آزاد ایجاد شده توسط الگوریتم مدل باراباشی آلبرت است. البته تعجبی ندارد، چون توزیعهای قانون قدرت، رفتار کاربران را در اینترنت توضیح می‌دهند. این وضعیت بدان دلیل است که غالباً جستجوی مستقل اطلاعات، غیرعملی (یا غیر ممکن) است اما در عین حال، تکثیر و ارسال پستهای دیگران، معمولاً یک استراتژی پرطرفدار است. علاوه بر این، رابط کاربر این تالار گفتگو، آشکارا نظردادن در مورد پیشنهادیه‌های پرطرفدار را با قرار دادن آنها در بالای صفحه ترویج می‌کند.

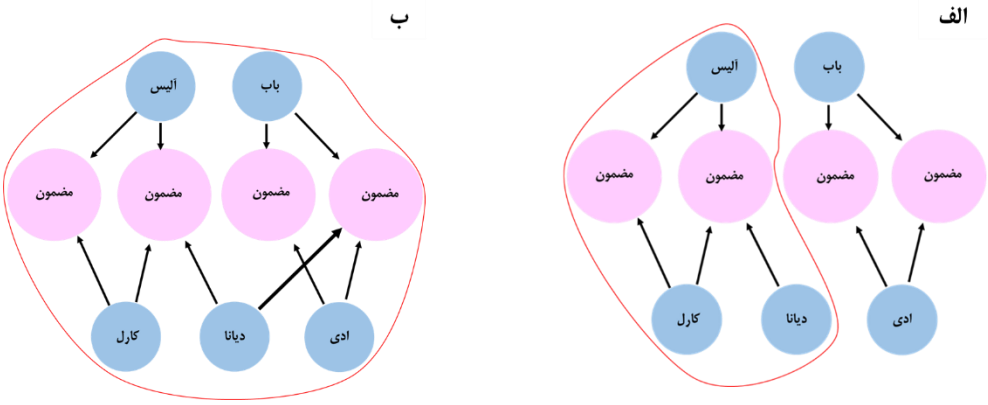
با نگاهی به درجه خروجی، می‌بینیم که ۷۲ درصد از مضامین مورد بحث، دست‌کم یک نظر را دریافت کرده‌اند. تمرکز کردن روی نظرات و نادیده گرفتن اکثر مطالب ارسال شده از رفتارهای آنلاین معمول نیست، ولی می‌توان این رفتار را با این حقیقت که اجتماعات سیاسی، تمایل به جذب اعضا دارند، توجیه نمود. دلیل دیگر می‌تواند (براساس پژوهش چنگ[۱]، دیل[۲]، و لیو[۳] در سال ۲۰۰۸) بعلت خاصیت حیاتی[۴] تالارهای گفتگو باشد. نتایج کار این محققان نشان داد که در مورد ۷۳ درصد از ویدئوهای یوتیوب، حداقل یک نظر داده شده است. این درصد تا حد زیادی با آنچه در این تالار گفتگو اتفاق افتاده، مطابقت دارد.

از نظر درجه خروجی نیز، ۶۱ درصد نظرات، توسط یک درصد کاربران نوشته شده‌اند. درصد بالای کاربران با یک نظر (۵۴ درصد) را می‌توان با سیاست سیستم تایید تالار گفتگو تبیین نمود. این سیستم، کاربران مهمان را به حفظ نام کاربری خود، در طول زمان، تشویق نمی‌کند. بنابراین کاربری که با نامهای متفاوت نظرات خود را می‌نویسد، بیش از یک نفر در نظر گرفته می‌شود.

همانطور که در فصول قبل نیز مطرح کردیم، در یک شبکه، همانند شبکه‌ای که در این مطالعه برای توضیح جنبش پنج ستاره، مورد استفاده قرار گرفته است (شکل ۵-۹)، یک سازند (یا مولفه) زیرمجموعه‌ای از گره‌ها است، به‌طوریکه یک مسیر ارتباطی میان تمامی گره‌های آن زیرمجموعه وجود داشته باشد. بطور مشابه، در شبکه دو قسمتی (دو-حالته) تحلیل شده در اینجا نیز، یک سازند زیرمجموعه‌ای از کاربران است که در زیرمجموعه مشابهی از بحثها (یا مضامین) شرکت می‌کنند. البته لازم نیست که کاربران یک سازند، در تمامی مباحثات موجود در آن سازند، مشارکت داشته باشند. در

[1] Cheng
[2] Dale
[3] Liu
[4] Vitality

حقیقت، هر دو کاربر موجود در یک سازند، در این حالت، عمدتاً بطور غیرمستقیم، به هم مرتبط هستند. در شکل ۶-۹ الف، کارول و ادی عضو دو سازند متفاوت هستند. در شکل ۶-۹ ب، یعنی بعد از اظهارنظر دیانا در بحثی که ادی نیز در آن فعال بوده است، آنها (یعنی کارول و ادی) عضو یک سازند یکسان می‌شوند.



شکل ٦-۹ ادغام دو سازند با افزودن یک نظر الف-قبل ب-بعد

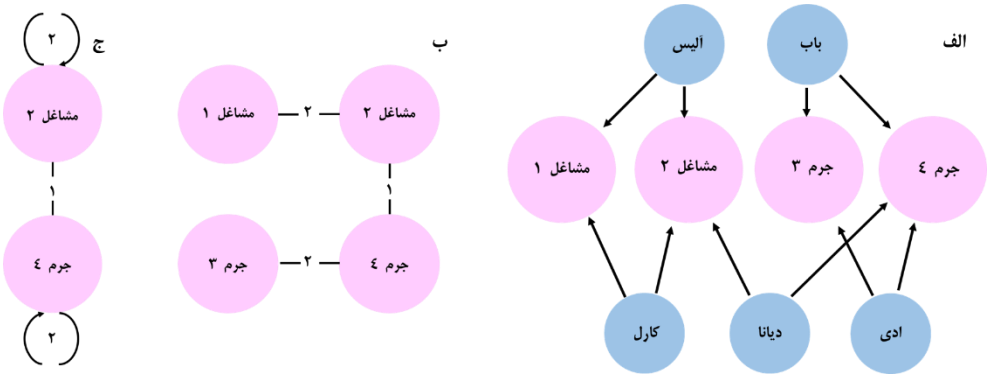
از میان ۷۰۱۱ سازند شناسایی شده در شبکه، یک ابرسازند[1] تکی، با اندازهٔ ۱۵۵۰۰۵ گره، حدود ۹۱ درصد از کل شبکه را در برمی‌گیرد. همچنین ۵۴۶۰ سازند –که حدوداً سه درصد اندازه شبکه هستند– و تنها از دو گره تشکیل شده‌اند (مثلاً یک کاربر و پیشنهادیه او) کاملاً از بقیه شبکه جدا هستند. کاربران این سازندها، کسانی هستند که با عضویت در جنبش، تنها یک پیام منتشرکرده و مشارکت بیشتری در بحث مربوط به سایر مضامین نداشته‌اند. این پیکره‌بندی خاص در شبکه‌های آنلاین بسیار رایج است. ابرسازندها یا ابرمولفه‌ها به طور طبیعی تمایل به ترکیب شدن دارند، زیرا احتمال ادغام هر دو مولفه، با افزایش اندازه آنها، افزایش می‌یابد. در نتیجه با افزایش کاربران یک سازند، نهایتاً شانس هر یک از اعضا جهت اظهارنظر در مورد مضامین موجود در یک مولفه متفاوت، آشکارا افزایش می‌یابد. از طرف دیگر، به این دلیل که احتمال دریافت اولین اظهارنظر برای هر مضمونی نسبتاً پایین است، اکثر سازندها هرگز اندازه‌ای بزرگتر از دو ندارند.

[1] Super-Component

۹-۳-۲ تصویر یک‌حالته: طبقه‌بندی مضامین

شبکه دو–حالته مربوط به تالار گفتگوی مذکور در این مطالعه، روابط میان "کاربران" و "مضامین" را ترسیم می‌نماید. بررسی ارتباط مستقیم میان مضامین مختلف، با ایجاد یک شبکه یک–حالته مستخرج از شبکه دو–حالته (یا دو–بخشی) امکان‌پذیر است. به این صورت که، هر دو مضمونی که به یک کاربر پیوند دارند، بهم متصل می‌شوند. به بیان دیگر، این شبکه، روابط میان مباحثی که شرکت‌کنندگان مشترکی دارند را ترسیم می‌کند. شکل ۷–۹ نشان می‌دهد که چگونه یک شبکه یک–حالته از یک شبکه دو–حالته، ایجاد می‌شود.

شبکه دو–حالته در شکل ۷–۹ الف، روابط میان پنج کاربر و چهار مضمون را ترسیم نموده است. در شکل ۹–۷ ب، هر یال غیرمستقیمی که دو مضمون را توسط یک کاربر پیوند می‌دهد، به صورت یک یال مستقیم میان مضامین، ترسیم شده است. شکل ۷–۹ ج، نیز یک گام دیگر ساده‌سازی را نشان می‌دهد: مضامین متعلق به مقوله‌های مشابه (بعنوان مثال مشاغل و جُرم) در یک گره ادغام شدند. در این حالت، روابط بین مضامین مشابه با حلقه‌های انعکاسی نمایش داده می‌شوند. این تغییر و تبدیل، به ما این امکان را می‌دهد که روابط میان "مقوله‌های بحث" را به تصویر بکشیم که در غیر این صورت، آنها در شبکه پیچیده‌ای از روابط پنهان بوده‌اند.

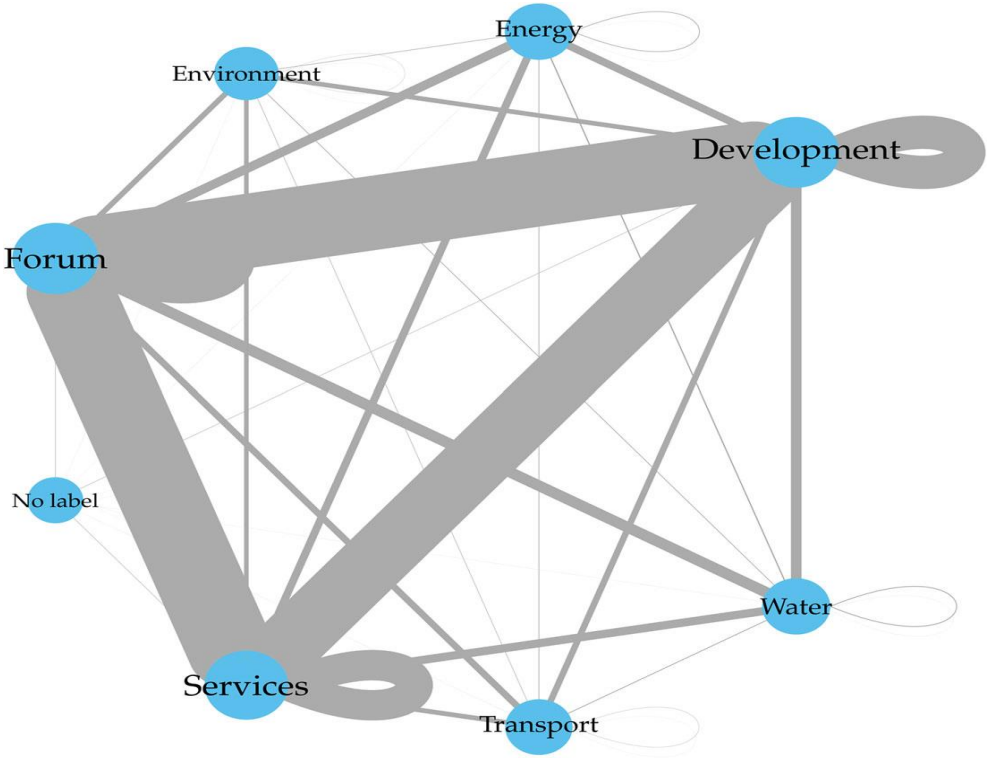


شکل ۷–۹ شبکه یک–حالتهٔ مضامین الف– شبکه دو–حالته ب– ارتباطات میان مقوله‌های مضامین ج– شبکه یک–حالته

به طور خاص شاید بخواهیم بدانیم که آیا کاربرانی که در یک "مقوله خاص" تخصص دارند، نهایتاً توجه‌شان را روی عناوین همان مقوله (به عنوان مثال مدیریت آب) متمرکز کرده و یا عناوین مربوط به

مقوله‌های متفاوت را مورد بحث قرار داده و در آن مشارکت می‌کنند. با توجه به شبکه نشان داده شده در شکل ۸-۹، مشهود است که کاربران در بحثهای مربوط به مقوله‌های مختلف درگیر می‌شوند، یعنی تخصص‌گرایی و درگیر شدن در یک مقوله خاص امری عادی نیست، بلکه یک استثنا می‌باشد. با مقایسه تعداد "یالهای خروجی از هر گره" به تعداد "یالهای انعکاسی[1] برای همان گره"، به نظر می‌رسد تنها بخشی از فعالیت کاربران در یک مقوله واحد رخ می‌دهد.

علاوه بر این، تجسم‌بخشی و مصورسازی قویترین یالها و ضعیفترین یالها نیز امکانپذیر است. منظور از قویترین یالها، کمانهایی هستند که مقوله‌های پرمخاطب مثل "توسعه"، "خدمات" و "تالار گفتگو" را بهم وصل می‌کنند. ضعیفترین یالها نیز، کمانهایی هستند که گره‌های کم مخاطب را بهم ربط می دهند. با کمال تعجب، "موضوعات زیست‌محیطی" از نوع مقولات کم طرفدار می‌باشد.

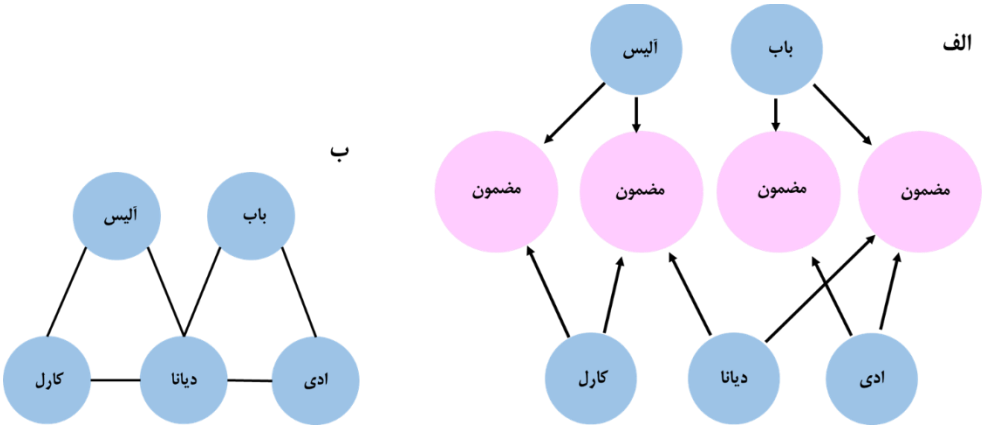


شکل۸-۹ روابط میان مقوله‌های مباحثه

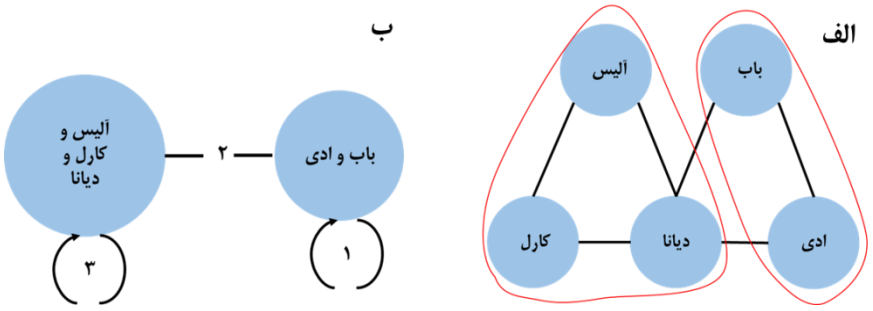
[1] Reflexive Ties

۹-۳-۳ تصویر یک‌حالته: کاربران

به طور مشابه با آنچه در مورد مصورسازی و ادغام شبکه یک–حالته برای " مضامین" مورد گفتگو قرار گرفت، شکلهای ۹–۹ و ۱۰–۹، جزئیات مربوط به شبکه "کاربران" و ادغام منتجه بر روی "اجتماعات کاربر" را نشان می‌دهند. تصویرسازی و تجسم‌بخشی نشان داده شده در شکلهای ۹–۹ الف و ۹–۹ ب، همان قواعد مربوط به تجسم‌بخشی مضامین (در بخش ۲–۳–۹) را دنبال می‌کند: یالهایی که دو کاربر را بطور غیرمستقیم از طریق یک مضمون بهم متصل می‌کنند، به یک یال (یا رابطه) مستقیم بین دو کاربر تبدیل می‌شوند. مرحله‌ای که در شکل ۱۰–۹ الف، نشان داده شده، یک الگوریتم پایین–بالا را برای انجمن‌یابی در شبکه‌های بزرگ اعمال می‌کند. به طوریکه، تا زمانی که پیمانگی شبکه بهینه شود، گره‌ها را در انجمنها گروهبندی می‌کند. به بیان ساده، بهینه‌سازی بدنبال آن است که تعداد یالهای موجود در اجتماعات حتی‌الامکان زیاد شده و یالهای میان اجتماعات تا حد امکان کم باشد. در نهایت چگونگی ادغام اجتماعات پس از شناسایی نیز در شکل ۱۰–۹ ب، نشان داده شده است.

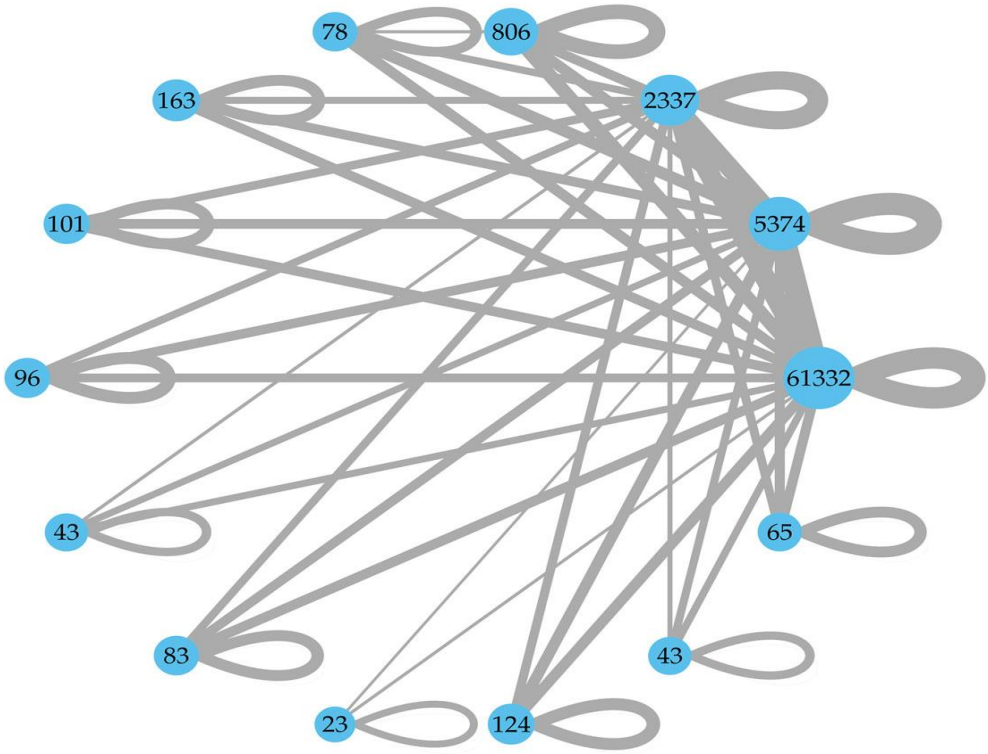


شکل ۹–۹ ادغام شبکه کاربران الف– شبکه دو–حالته ب– تجسم یک–حالته



شکل ۱۰-۹ شبکه یک-حالته کاربران الف- انجمن‌یابی ب- ارتباط اجتماعات

با اعمال الگوریتم بیشینه‌سازی پیمانگی، نهایتاً ۷۰۲۱ انجمن موجود در شبکه شناسایی شده که ۸۶ درصد از آنها، انجمنهای تک عضوی هستند. روابط میان اجتماعاتی که بیش از ۲۰ عضو دارند در شکل ۱۱-۹ نشان داده شده است. جالب است بدانیم دو انجمنی که بیشتر از بقیه عضو دارند، دارای ویژگیهای متفاوتی هستند: بزرگترین انجمن ۷۳ درصد از کاربران را گروهبندی نموده است؛ درحالیکه دومین انجمن، تنها شش درصد کاربران را گروهبندی نموده است. انجمن دوم، با کمتر از یک دهم جمعیت، به دلیل داشتن یالهای بازگشتی (حلقه)، به اندازه دو برابر بزرگترین انجمن، متراکم است. به نظر می‌رسد، انجمن دوم، اجتماعی بسیار فعال از کاربرانی است که قصد دارند روی مضامین نسبتاً کمتری تمرکز کنند.



شکل ۱۱-۹ روابط میان اجتماعات کاربران

۹-٤ تجزیه و تحلیل مناظره‌ها

در این قسمت، نگاهی به دو مناظره مهم تالار، یعنی "مهاجرت" و "حمایت برای یک دولت ائتلافی" می‌اندازیم. مناظره اول این امکان را می‌دهد که طبقه‌بندیهای سیاسی سنتی (یعنی نگرشهای سیاسی چپ و راست) را بر استدلالها، مشاجرات و محاجه‌های موجود اعمال کنیم. درحالیکه، مناظره دوم، در مقطع خاصی مطرح شده و به گونهٔ دیگری (یعنی بر اساس دیدگاههای موافق و مخالف دولت ائتلافی) قابل تفسیر است.

این تجزیه و تحلیل، کاربران را با توجه به جهتگیری سیاسی بحث، کدگذاری کرده و سپس آنها را در طول مسیر مباحثات در هر مضمون، دنبال می‌کند. به منظور شناسایی اینکه آیا هر جفت مضمون، به طور غیرمنتظره، تعداد کم یا تعداد زیادی "شرکت‌کننده مشترک" دارند، یک معیار شناسایی تعریف شده است. منظور از شرکت‌کننده یا کاربر مشترک آن عضوی است که در مورد دو مضمون مختلف، اظهارنظر کرده باشد. در واقع این فرد از منظر آن دو مضمون، "یک کاربر مشترک" است. معیارِ شناسایی فوق، برای هر یالی که دو مضمون را مرتبط می‌کند، از طریق یک مدل رگرسیون پواسون[۱] محاسبه می‌شود. مدل موردنظر را می‌توان بصورت معادله ۱–۹ نوشت:

$$Log_e(shared_users) = \beta_0 + \beta_1 \times multi_size + \beta_2 \times time_diff \qquad ۱–۹$$

در رابطه ۱–۹، *shared_users*، تعداد کاربران مشترک هستند (یعنی تعداد اعضایی که در مباحثه مربوط به یک زوج مضمون مشارکت دارند)، *mult_size*، نمایشگر حاصلضرب تعداد کاربران مضمون اول در تعداد کاربران مضمون دوم است و *time_diff*، قدرمطلق تفاوت زمانی، اولین پست مضمون اول و اولین پست مضمون دوم، بر حسب ثانیه می‌باشد.

نتیجه تابع برازش رابطه ۱–۹ برای هر جفت مضمون، تخمین تعداد کاربران مشترکی است که انتظار می‌رود مشاهده کنیم. بعلاوه برای مضامینی که اندازه و زمان نقاط پایانی آنها مشخص است، می‌توان فاصله مقادیر مشاهده‌ای و انتظاری را محاسبه کرد. یعنی با مقایسه "تعداد واقعی کاربران مشترک" در مورد یک جفت مضمون، با "تعداد تخمین زده شده توسط مدل" با فاصله اطمینان ۹۵ ٪، امکان "کمّی کردن" فاصلهٔ بین مقدار مشاهده‌ای و مقدار انتظاری وجود دارد[۲]. در این صورت، یک مقدار مشاهده‌ای بالاتر از فاصله اطمینان ۹۵ ٪ عملاً، نمایشگر "سازگاری یا الفت"[۳] بین دو گروه از کاربران است و یک مقدار پایین‌تر از این فاصله، نشاندهنده "ناسازگاری یا انزجار"[۴] بین دو گروه از کاربران است. یک مقدار در میانه فاصله فوق هم، نشان می‌دهد که کاربران این دو مضمون، در مقایسه با کل شبکه، مطابق با آنچه انتظار می‌رود، رفتار می‌کنند.

---

[۱] این مدل در حقیقت، یک *مدل گرانش یا جاذبه ساده* است (که در آن زمان بجای فضا قرار گرفته است). این مدل فرض می‌کند که تعاملات میان دو جمعیت، رابطه مستقیمی با اندازه آنان داشته و همزمان، رابطه غیرمستقیمی با فاصله آنها بر حسب زمان دارد.

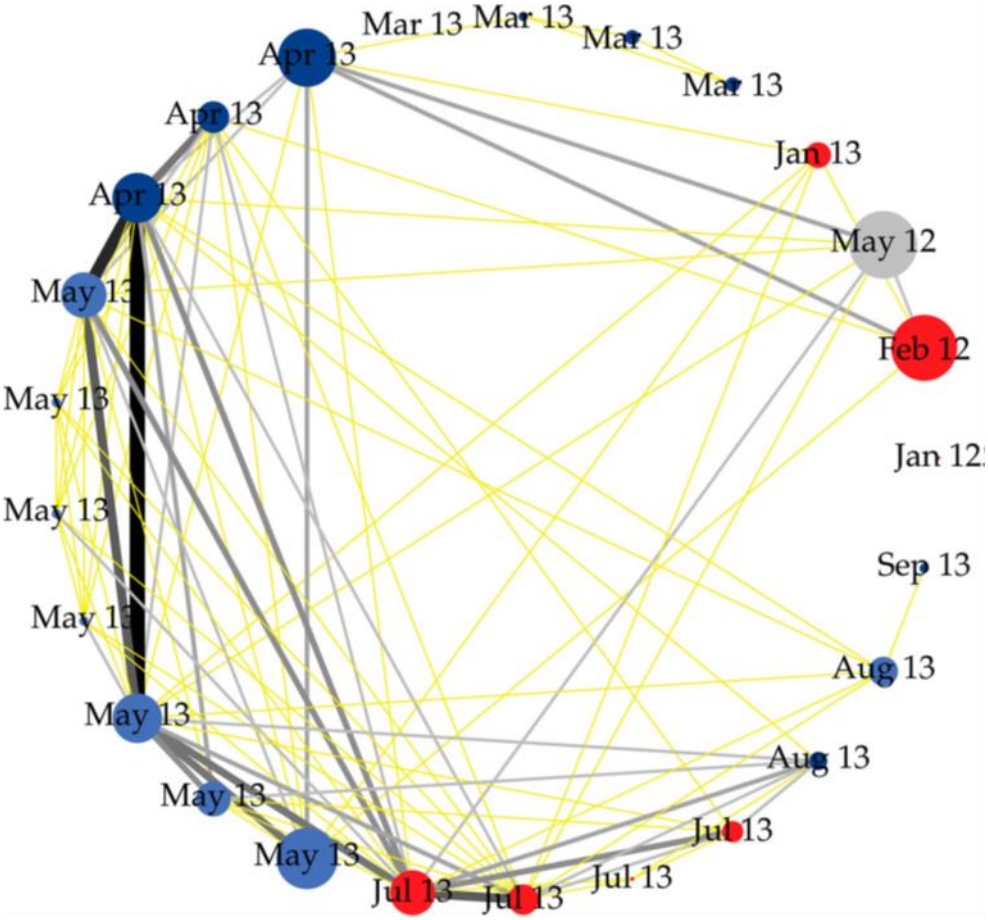
[۲] فاصله اطمینان ۹۵ ٪ در واقع دامنه آن مقادیری است که شامل "مقدار ناشناخته درست" (True Unknown Value) پارامترها (در اینجا یعنی کاربران مشترک) به احتمال ۹۵. است.

[3] Affinity

[4] Antipathy

۹-٤-۱ مهاجرت

روابط میان مضامین مربوط به مباحثات پیرامون مهاجرت، در شکل ۱۲-۹ رسم شده‌اند. شبکه نشان‌داده شده در اینجا یک منطقه کوچک (یا زیرگرافی) از یک شبکه یک-حالته است، که در قسمتهای پیش تشریح و در شکل ۷-۹ ب، نمایش داده شد. جهتگیری سیاسی هر مضمون تا حدی بر اساس پستی که مضمون مورد بحث را خلق کرده و به آن عنوانی را نسبت داده، قابل تشخیص است. حال ما با رنگ‌آمیزی گره‌ها، براساس همین جهتگیری سیاسی، آنها را رمزگذاری کرده‌ایم: قرمز تیره برای چپ تندرو، قرمز کم رنگ برای چپ متعادل، آبی روشن برای راست متعادل و آبی تیره برای راست تندرو. "اندازه گره‌ها" متناسب با تعداد نظرات ارائه بوده و "تاریخ اولین پست" هم بر روی هر گره نشان داده است. "پهنای یالها" متناسب با تعداد کاربرانی است که در مباحثهٔ هر دو مضمون شرکت کرده‌اند. با بالاتر رفتن ارزش واقعی یالها، از حد بالایی اطمینان ۹۵ درصدِ مدل رگرسیون پواسون، رنگ یالها تیره‌تر می‌شوند. یالهای زرد نشان می‌دهند که تعداد کاربران مشترک، کمتر از حد مورد انتظار است. حدودا ۶۳,۷ درصد یالها، از بین ۹۱ یال اتصال‌دهنده مضامین بحثها پیرامون مهاجرت، کمتر از فاصله اطمینان تخمینی مدل هستند (هرچند برخی تنها اندکی کمتر بوده‌اند). اما از آن مهمتر، یالهای زرد، هیچ الگوی تبعیض‌آمیزی ندارند و آشکارا گره‌های همرنگ را به همان اندازه گره‌های رنگارنگ به هم پیوند می‌دهند. بعلاوه، همه "مضامین اصلی مباحثه" به شدت به گره‌های با رنگ مخالف متصل هستند که نشان می‌دهد به طور متوسط، کاربران علاقه‌مند به این عناوین، تنها با همفکران سیاسی خود مباحثه نکرده‌اند.



شکل ۱۲-۹ روابط میان مضامین مباحثات پیرامون مهاجرت

۹-٤-۲ دولت ائتلافی

مناظره ترسیم شده در شکل ۱۳-۹، در مورد حمایت از دولت ائتلافی، بشدت دورسته[1] است. این شکل، دورستگی میان گره‌های قرمز (به نمایندگی از مضامینی که به نفع حمایت جنبش پنج ستاره از دولت جدید ائتلافی هستند) و گره‌های آبی (به نمایندگی از مضامینی که در مقابل یا مخالف با دیدگاههای گره‌های قرمز هستند) و یک گره خاکستری (با موقعیت خنثی) را نشان می‌دهد.

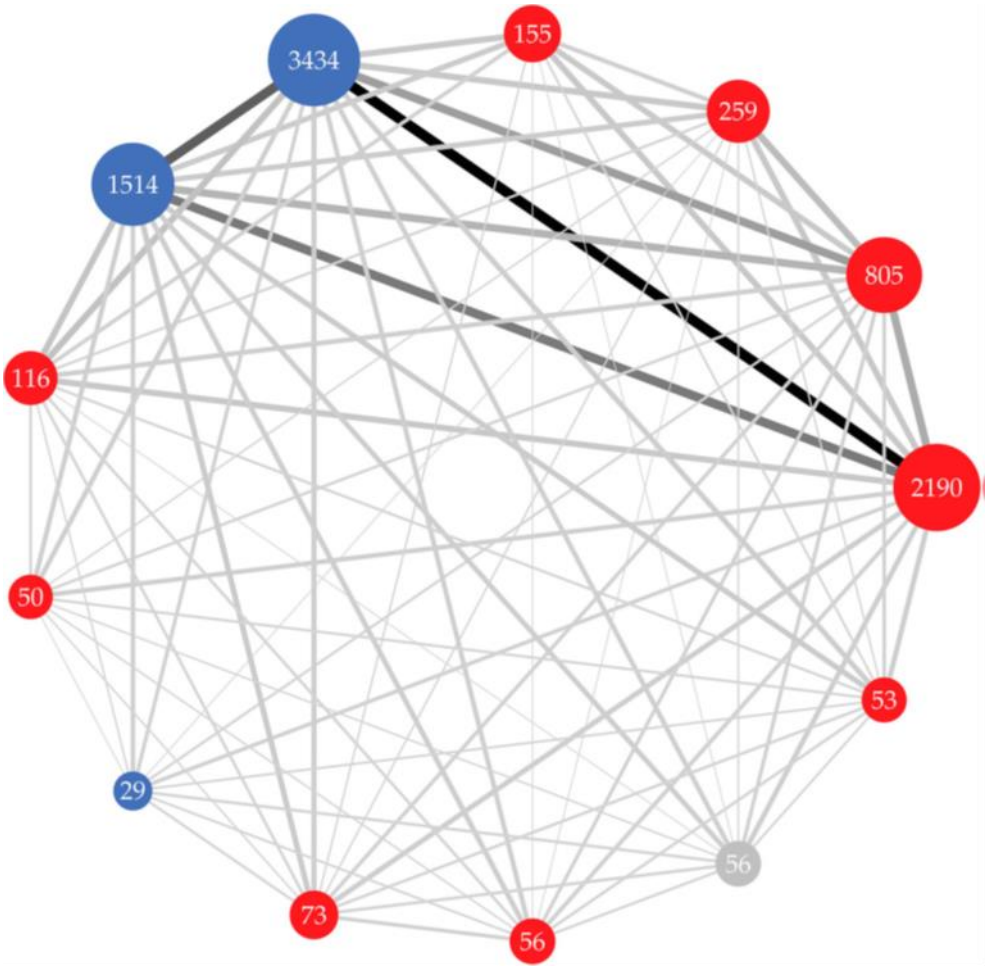
ارسال پیامها با مضامین سه‌گانه فوق در یک دوره زمانی خاص، آغاز شده است. در واقع آغاز ارسال این مضامین از فوریه ۲۰۱۳ (بلافاصله پس از انتخابات عمومی) تا آوریل ۲۰۱۳ بوده است.

این مناظره در مقایسه با آنچه از ابتدا تا ۱۷ فوریه ۲۰۱۴ در تالار رخ داده، بسیار شلوغ بوده و بیشترین تعداد اظهارنظر را دارد. تعداد شرکت‌کنندگان در هر مضمون مباحثه، بر روی گره نشان داده شده است.

همانند مناظره مهاجرت، به نظر نمی‌رسد کاربرانی که در موضع مدافع یا مخالف دولت ائتلافی هستند، از اظهارنظر در مورد مضمونی که (حداقل در آغاز پست) دیدگاههای مخالفین را ترویج می‌دهد، خودداری کنند.

کاربران مشترک هیچ زوج گرهی، کمتر از حد مورد انتظار نیستند – یعنی همه مقادیر مشاهده شده بالاتر از سطح اطمینان پیش‌بینی‌شده توسط مدل است – که این امر گواهی می‌دهد، بحث تا چه حد شدید بوده است!

[1] Dichcotomic



شکل ۱۳-۹ روابط میان مضامین مباحثات پیرامون حمایت از دولت ائتلافی

۹-۵ نتیجه‌گیری

تجزیه و تحلیل شبکه به ما اجازه می‌دهد تا از حجم عظیم داده‌های متنی که از طریق تعاملات اجتماعی آنلاین تولید می‌شوند، بخوبی استفاده کنیم. این داده‌ها، فرصت بی‌نظیری را برای تحقیق و مشاهده

مکالمات غیررسمی توسط پژوهشگران (به عنوان ناظران بی‌طرف) فراهم می‌کند. چنین امکان و فرصتی تا قبل از تولید این‌گونه داده‌های انبوه از طریق فناوریهای ارتباطی، هیچگاه تا این حد گسترده وجود نداشت. از نتایج ارایه شده در این فصل، می‌توان این‌گونه استنتاج کرد که کنشگران برخط نسبت به کنشگران برون‌خط، رفتار متفاوتی ندارند. این مطلب نشان می‌دهد که دو مجموعه از بازیگران (آنلاین/آفلاین) احتمالاً بسیار مشابه یا همپوشان هستند. یعنی هر دو فضا توسط افرادی اشغال شده‌اند که متناوباً و به طور منظم در هر دو قلمرو عمل می‌کنند و به تدریج مرز میان این دو قلمرو را محو می‌کنند. تقارن و تشابه در رفتارهای آنلاین یا آفلاین، نه تنها برای کسانی که فضای آنلاین را فرصتی برای اصلاح عدم توازن در فضای سنتی می‌دانستند، خبر خوبی نیست؛ بلکه درسهای آموخته‌شده از تعاملات آنلاین ممکن است به صورت آفلاین نیز معتبر باشند.

یک نتیجه جالب دیگر که از دور اول "تحلیل جنبش پنج ستاره" پدیدار شد این است که ترس از تکه‌تکه شدن اجتماعی، به دلیل مشابهت سخت‌افزاری افراد و امکان کاهش تنوع در شبکه آنان، احتمالاً اغراق‌آمیز است. هر چند به نظر می‌رسد تنوع در برخی موارد، موجب تاثیر منفی بر تبادلات می‌شود، اما موجب جدایی و محرومیت افراد نمی‌شود. پس نباید فراموش کرد که مردم –بخصوص در مسایل سیاسی– ممکن است توسط نوعی حس اجتماعی و هویت مشترک، هدایت شوند. از این رو، مردم حتی در زمانی که ارزشها و عقاید متفاوتی داشته باشند، با یکدیگر همکاری می‌کنند.

این حقیقت که کاربران تالار گفتگو تمایل دارند در بحثهای مربوط به موضوعات مختلف درگیر شوند، با آن ایده‌ای که می‌گوید افراد صرفاً براساس علایق و ارزشهایشان برانگیخته می‌شوند، سازگاری نداشته و آنرا تضعیف می‌کند. در واقع، همانطور که کانور[1] و همکاران در سال ۲۰۰۵ مشاهده کردند، افراد بیشتر به درگیر شدن در بحثها علاقمند هستند تا متقاعد کردن یکدیگر. گفتگوهای سیاسی روزانه، ممکن است با قواعدی انجام شود که بیشتر به یک بازی اجتماعی[2] مربوط می‌شود تا یک بازی خردگرایانه سیاسی[3]. کاربران عمدتاً، به تالار گفتگو می‌آیند تا عقاید خود را اجتماعی[4] کرده (یعنی در سطح جامعه معرفی کرده) و با افکار دیگران پیرامون موضوعات مورد علاقه خود، مواجه شوند.

[1] Conover
[2] Social Game
[3] Political Rationalistic Game
[4] Socialise

نهایتاً، عجیب نیست که این داده‌ها، نوعی همبستگی میان رویدادهای مهم سیاسی و حجم مباحث سیاسی در تالار را نشان می‌دهند. این موضوع البته نشان می‌دهد که بحثهای سیاسی در خلاء انجام نمی‌شود بلکه باید در بافتار فرایندهای سیاسی تحلیل شوند.

منبع:

Bailo, F. (2015). Mapping online political talks through network analysis: a case study of the website of Italy's Five Star Movement. *Policy Studies*, 36(6), 550-572.

پیوستها

پیوست الف: پایتون ۳.X، یک راهنمای نحوی (یا ساختاری) سریع

پایتون یک زبان برنامه‌نویسی همه‌منظوره و شی‌گرای[۱] پویای چند-پارادایمی[۲] است. نام این زبان برنامه‌نویسی، براساس اسم یک گروه کمدی بریتانیایی، به نام **مانتی پایتون**[۳]، انتخاب شده است. پایتون یک زبان برنامه‌نویسی ساده، قابل حمل (پرتابل)[۴]، منبع باز (متن‌باز) و قدرتمند است که به سرعت کار می‌کند و به طور موثر ادغام می‌شود.

اولین نسخه کد پایتون (۰,۹,۰) توسط *خودو فان روسوم*[۵]، در فوریه ۱۹۹۱ در *CWI*[۶] (مؤسسه ملی تحقیقات ریاضی و رایانه) در آمستردام هلند، منتشر شد. زبان برنامه‌نویسی پایتون، از زبان برنامه نویسی *ABC* گرفته شده است که یک زبان برنامه‌نویسی عمومی بوده و در *CWI* توسعه داده شده بود. اگرچه امروز پایتون توسط یک تیم توسعه‌دهنده اصلی در این مؤسسه نگهداری می‌شود، همچنان فان راسوم نقش مهمی در هدایت و پیشرفت آن دارد.

پایتون را می‌توان اینطور شرح داد:

۱) زبانی تفسیری[۷] است، یعنی، توسط مفسر در زمان اجرا پردازش می شود؛

۲) زبانی تعاملی است، یعنی، هنگام استفاده از پایتون امکان ارتباط مستقیم سریع با مفسر وجود دارد؛

۳) زبانی شی گرا است؛

۴) زبانی مناسب برای مبتدیها است، یعنی، گزینه‌ای عالی برای برنامه‌نویسان مبتدی است که می‌خواهند برنامه‌های کاربردی را توسعه دهند؛

---

[۱] Object-Oriented

[۲] Multi-Paradigm

[۳] Monty Python

[۴] Portable

[۵] Guido Van Rossum

[۶] Centrum Wiskunde & Informatica

[۷] Interpreted Language

پایتون دارای کاربردهای فراوانی است؛ عموماً از آن برای موارد ذیل استفاده می‌شود:

۱) توسعه وب، مانند: *Django*، *TurboGears* و *Plone*؛
۲) ارتباط با پایگاههای داده، مانند: *MongoDB*، *MySQL*، *PostgreSQL* و *Oracle*؛
۳) ارتباط‌سازی گرافیکی با کاربر (*GUI*)در دسکتاپ، مانند: *GTK*، *QT*، *Tk* و غیره؛
۴) پایتون با محاسبات علمی، مانند: *SciPy*، *Scientific Python* و غیره؛
۵) برنامه نویسی شبکه: *Twisted*؛
۶) توسعه نرم افزار: *SCons*، *Buildbot*، *Roundup* و غیره؛
۷) بازیها و گرافیک سه‌بعدی، مانند: *PyGame*، *PyKyra* و غیره؛

یکی از بزرگترین نقاط قوت پایتون، اندازه و ابعاد کتابخانه استاندارد آن و دیگر کتابخانه‌های منبع باز آن مانند کتابخانه‌های توابع ریاضی، محاسبات علمی، گرافیک، یادگیری ماشین، تجزیه *XML*، دانلود صفحات وب و غیره است. در حقیقت، طراحی "یک زبان اصلی کوچک" با "یک کتابخانه استاندارد بزرگ و قابل گسترش"، در همان ابتدا ایدهٔ فان راسوم بود.

همه چیز در پایتون بعنوان یک شی (مثلاً اعداد صحیح، فهرستها، رشته‌ها، توابع، ماژولها، کلاسها، و غیره) و یا بعنوان روابط بین اشیا به گونه‌ای ارائه می‌شود که هر شی دارای شناسه[۱]، نوع[۲] و مقدار[۳] است. گرچه پایتون شی‌گرا است و همه چیز در زبان یک شی است، ولیکن این برنامه همچنین از سبکهای مختلف برنامه‌نویسی همچون رویه‌ای[۴]، عملکردی[۵]، جنبه‌گرا[۶]، طراحی–با–قرارداد[۷] و برنامه نویسی منطقی[۸] پشتیبانی می‌کند. این امر در هنگام پیاده‌سازی الگوریتمهای یادگیری ماشین مفید است، بنابراین، در عمل، مناسب‌ترین سبک برنامه‌نویسی در هر موردی است.

پایتون با دیگر زبانهای سطح بالا (مثلاً *C*، *C++*، یا *Java*) متفاوت است، زیرا کد نوشته شده در زبانهای نوشتاری (یا تایپی) پویا همچون پایتون معمولاً کوتاهتر از کد نوشته شده در زبان *C*، *C++*، یا

---

[۱] Identity
[۲] Type
[۳] Value
[۴] Procedural
[۵] Functional
[۶] Aspect-Oriented
[۷] Desigh-by-Contract
[۸] Logic Programming

Java می‌باشد. در پایتون، هیچ اشاره‌گری مورد استفاده قرار نمی‌گیرد و هیچ تبدیل و یا ترکیب‌بندی قبلی[۱] آن به بایت‌کد[۲] مورد نیاز نیست، زیرا می‌توان آن را مستقیماً تفسیر کرد.

قواعد نحوی پایتون

- فایلهای *Python* دارای پسوند *py*. هستند.
- بجای پرانتز در پایتون برای حذف بلوکها از دندانه استفاده می‌شود. تعداد فضاها یا فاصله‌ها، متغیر است، اما تمام عبارات[۳] در یک بلوک خاص باید به یک مقدار مشابه دندانه داشته باشند.
- انواع داده‌های پایه‌ای: اعداد (بعنوان مثال، عدد صحیح، شناور[۴] و مختلط[۵])، بولی[۶] و دنباله‌ها (بعنوان مثال، رشته‌ها، فهرستها، دیکشنریها و اصطلاحات).
- خط هدر عبارات ترکیبی، مانند: *if*، *while*، *def* و *class*، باید با دونقطه(:)، خاتمه یابد.
- نقطه ویرگول(؛)، در انتهای عبارات، اختیاری است.
- *Print*، یک کلمه کلیدی برای دادن خروجی به یک کنسول یا یک فایل است. بعنوان مثال:

```
print(“Hello Python!”)
```

- خواندن از صفحه کلید، مثال زیر، این روش یک خط ورودی کاربر را بعنوان یک رشته، باز می‌کند.

```
name = input(“enter your name”)
```

- نظرات[۷]: نظرات یک خطی (#) و نظرات چند خطی (*‘‘‘ ____ ’’’*).
- *help(<obj>)*، برای شی مورد نظر بنوعی کمک/سند را با استفاده از *pydoc.help* فراهم می‌کند.
- تابع *dir (<obj>)*، فهرست مرتب‌سازی شده‌ای برمی‌گرداند که حاوی ویژگیها/روشهای موجود در آن شی است. با توجه به اینکه ویژگیها/روشهایی که با *’’/’’* شروع می‌شوند، ویژگیها/روشهای داخلی هستند و نباید استفاده شوند مگر اینکه شما بدانید که چه (ویژگی یا روشی) استفاده می‌کنید، لذا *dir()* نامها را به دامنه فعلی باز می‌گرداند.

---

۱ Prior Compilation
۲ Bytecode
۳ Statements
۴ Float
۵ Complex
۶ Boolean
۷ Comments

متغیرها

- هیچ اعلام یا اظهار قبلی برای نوع متغیرها لازم نیست.
- یک متغیر می‌تواند به هر نوع داده (مانند *Tuple*، *List*، *Dictionary*، *Int*، *String*، *Complex* یا هر شی دیگری) اشاره کند. آنها مرجعی برای حافظه تخصیص یافته هستند.
- پایتون به صورت پویا تایپ می‌شود. هنگامی که یک مقدار به یک متغیر اختصاص داده می‌شود، تصریح و تخصیص آن بصورت خودکار صورت می‌گیرد.
- متغیرها می‌توانند نوع را تغییر دهند. این کار به سادگی با اختصاص دادن یک مقدار جدید از یک نوع متفاوت صورت می‌گیرد.
- پایتون اجازه می‌دهد یک مقدار واحد را به چندین متغیر بطور همزمان اختصاص داد.
- همچنین ممکن است چندین شیء را به چندین متغیر اختصاص داد.

اعداد

- اعداد در پایتون، اشیاء تغییرناپذیر[۱] هستند. تغییرناپذیری به این معناست که اشیا نمی‌توانند مقادیر خود را تغییر دهند، بعنوان مثال، ۱۲۳۴، ۳,۱۴۱۵، *۴+۳j*.
- سه نوع داده ساخته شده برای اعداد در پایتون ۳ عبارتند از: ۱- اعداد صحیح، ۲- اعداد اعشاری و ۳- اعداد مختلط که شامل دو قسمت است: واقعی و موهمی.
- توابع عددی متداول شامل: *int(x)*، *float(x)* ، *abs(x)*، *exp(x)*، *log(x)*، *pow(x,y)* و *sqrt(x)* هستند.

رشته‌ها

- رشته‌ها، مجموعه‌ای متناوب از کاراکترها در بین علامت نقل قول هستند، بعنوان مثال:
mystr = ''This is a quick Python 3.x syntax guide.''
- رشته‌های پایتون اشیاء تغییرناپذیر هستند (یعنی، نمی‌توانند مقادیر خود را تغییر دهند).
- نوع داده *''character''* در پایتون وجود ندارد.
- برای به روزرسانی یک رشته موجود، می‌توانیم (مجدداً) یک متغیر را به یک رشته دیگر اختصاص دهیم.

---

[۱] Immutable Objects

- رشته‌هایی به طول یک کاراکتر بعنوان رشته‌های عادی مورد استفاده قرار می‌گیرند (یعنی هیچ نوع کاراکتری (*type character*) در قواعد نحوی پایتون استفاده نمی‌شود).
- نقل قولهای تکی (‘)، دوتایی (“) و سه‌تایی (‘‘‘ یا “””) برای نشان دادن رشته‌ها استفاده می‌شود.
- شاخصهای (یا اندیسهای) رشته‌ای از ۰ شروع می‌شوند و در انتها از مقدار ۱- عمل می‌کنند.
- اپراتورهای رشته‌ای متداول، شامل (+) برای پیوند، (*) برای تکرار، ([]) برای برش محدود، ([:]) برای برش محدوده و (*in*) برای بررسی عضویت، هستند.
- کاراکترهای ویژه (کنترلی) با استفاده از کاراکتر بک‌اسلش (’’\’’)، می‌توانند وارد شوند.
- روشهای رشته‌ای متداول شامل:

str.count(sub, beg=0, end=len(str)),
str.isalpha*()*,
str.isdigit(),
str.lower(),
str.replace(old, new),
str.split(str = ‘ ‘),
str.strip(),
str.title(),

- توابع رشته‌ای متداول شامل *str (x)*، برای تبدیل *x* به یک رشته و *len(string)*، برای پیدا کردن طول کل رشته است.

فهرستها

- فهرست یک گروه مرتب از اقلام یا عناصر است. عناصر فهرست نباید از یک نوع باشند. می‌توان فهرستی را در درون فهرست دیگر قرار داد.
- فهرست شامل مواردی است که توسط کاما جدا شده و با براکت محصور شده‌اند، مثلاً

myList = [1, [2, 'three'], 4]

- فهرستهای پایتون اشیاء ناپایدار یا تغییرپذیر[۱] هستند؛ یعنی، می‌توانند ارزشهای خود را تغییر دهند.
- شاخصهای فهرست، مانند رشته‌ها، از ۰ شروع می‌شوند و در انتها از مقدار ۱- عمل می‌کنند. آنها را می‌توان درست در انتها گسترش داد.

[۱] Mutable Objects

- فهرستها می‌توانند فهرستهای فرعی را بعنوان عناصر داشته باشند و این فهرستهای فرعی ممکن است شامل فهرستهای فرعی دیگری باشند.
- عملیات فهرست عبارتند از برش ([] و [:])، تکرار (+)، تکرار (*) و عضویت (در).
- توابع فهرستی متداول شامل موارد زیر است:

slicing ([ ] and [:]),
concatenation (+),
repetition (*),
membership (in),

- روشهای فهرستی متداول شامل موارد زیر است:

list.append(obj),
list.insert(index, obj),
list.count(obj),
list.index(obj),
list.remove(obj),
list.sort(),
list.reverse(),
list.pop().

چندتاییها یا تاپلها

- یک تاپل، دنباله ترتیبی پایدار یا تغییرناپذیر[۱] از اقلام است. تغییرناپذیری به این معنی است که پس از این که یک بار اعلام شد نمی‌توان آن را تغییر داد. تاپلها را می‌توان بعنوان یک آرایه ثابت در نظر گرفت.
- یک تاپل حاوی آیتمهایی است که با کاما (،)، جدا می‌شوند و در داخل پرانتز قرار می‌گیرند، ممکن است تاپلهای آشیانه‌ای، درون یکدیگر، وجود داشته باشند؛ مثلاً:

myTuple = (1, 'spam', 4, 'U')

- یک تاپل را می‌توان با اختصاص (مجدد) یک متغیر به یک تاپل دیگر به روز کرد.
- جهت پردازش سریعتر و حفاظت از داده‌ها در برابر تغییرات تصادفی، تاپلها نسبت به فهرستها ترجیح داده می‌شوند.
- عملیات تاپل، شامل برش ([] و [:])، تکرار (+)، تکرار (*) و عضویت (*in*) می شود.
- یک تاپل با یک مقدار تکی باید کاما (،)، داشته باشد؛ مثلاً: *t = (17,).*

[۱] Immutable Ordered Sequence

دیکشنریها

- دیکشنریها، ظروفی هستند که آیتمها را بصورت زوج کلید/مقدار ذخیره می‌کنند.
- دیکشنریهای پایتون اشیا قابل‌تغییری هستند که می‌توانند مقادیر خود را تغییر دهند.
- دیکشنریها، نوعی جدول هش[۱] (جدول درهم‌سازی) هستند، که شامل زوج کلید/مقدار مربوط به عناصر نامرتب هستند. کلیدها باید نوع داده‌های تغییرناپذیر باشند که معمولاً اعداد یا رشته هستند و مقادیر می‌توانند هر نوع شیء دلخواه پایتون باشند.
- یک دیکشنری، به وسیله یک جفت آکلاد یا براکت منحنی‌شکل[۲] ({}) محصور می‌شود، آیتمها با کاما (،) و هر کلید از مقدار آن با یک علامت دونقطه (:) جدا می‌شوند؛ مثال:

```
{'food':'spam', 'taste': 'yum'}
```

- ارزش مقادیر دیکشنریها را می‌توان با استفاده از براکت ([]) و یک کلید، تعیین کرد.
- روشهای دیکشنری رایج شامل موارد زیر است:

```
dict.keys(),
dict.values(),
dict.items(),
dict.get(key,
default = None),
dict.has_key(key),
dict.update(dict2),
dict.clear()
```

- برای تکرار در دیکشنری:

```
for key, value in a_dictionary.items(): print key, value
```

شرطها

- عبارات *if*، عبارات قدرتمند تصمیم‌گیری هستند که برای کنترل جریان اجرای برنامه استفاده می‌شوند. قواعد نحوی دستور *if*:

```
[if expression: statement(s)],
[if expression: statement(s); else: statement(s)],
```

---

[۱] Hash Table Type

[۲] Curly Braces

[if expression1: statement(s); elif expression2: statement(s); else: statement(s)],

- درست و غلط[۱] اشیاء بولی از کلاس "*bool*" هستند و تغییرناپذیر هستند.
- پایتون هر مقدار غیر صفر و غیر تهی را درست فرض می‌کند. در غیر این صورت مقداری اشتباه است.
- پایتون عبارات سوئیچ یا موردی را ارائه نمی‌دهد.

حلقه‌ها

- حلقه، فرآیند اجرای مکرر یک بلوک از عبارات است.
- حلقه *for*: حلقه *for* با یک فهرست از مقادیر تکرار می‌شود. حلقه *for* در پایتون، ممکن است عبارت اختیاری، *else*، را نیز داشته باشد؛ مثال:

[for x in X: print(“current letter is :”, x)]

- حلقه *while*: حلقه *while* برای اجرای مکرر یک بلوک از کدها و تا زمانی است که یک شرط توقف، برقرار شود. قاعده نحوی:

[while condition: statement(s)]

- عبارت *while-else*: بخش اختیاری *else*، تنها در صورتی اجرا می‌شود که ما به صورت معمول (و نه با انقطاع[۲] کار) از حلقه خارج شویم؛ قاعده نحوی آن بصورت زیر است:

while condition: statement(s); else: statement(s)

- دستورات و یا عبارات کنترل حلقه: این دستورات عبارتند از:

Break: باعث می‌شود که اجرای حلقه قطع شده و اجرا به عبارتی که بلافاصله بعد از حلقه می آید انتقال یابد؛

Continue: باعث می شود که حلقه، ادامه فرآیند خود را از دست بدهد و مجدد تنظیم شود؛

Pass: زمانی استفاده می‌شود که یک عبارت از لحاظ نحوی مورد نیاز است، اما نمی‌خواهیم دستور یا کد اجرا شود.

- *range(N)*: فهرستی از اعداد، *[0, 1, …., N-1]*، تولید می‌کند، بعنوان مثال: *range(i, j, k)*، که "*i*" (نقطه) شروع، "*j*" شرط توقف و "*k*" گام است.

---

[۱] True and False

[۲] Break

- درک فهرست. استفاده عادی از بک‌اسلش (\)، برای "حلقه" جهت تکرار و ایجاد یک فهرست جدید است. درک فهرست، کار فوق را ساده کرده است. قاعده نحوی:

[<expression> for <target>in <iterable> <condiction>],

توابع پایتون

- یک تابع عملاً گروهی از عبارات است که در صورت درخواست اجرا می‌شود. قاعده نحوی:

def function_name(parameters): "function_docstring"; function_statements; return [expression],

- توابع پایتون نیز شی هستند؛ نوع خروجی تابع مورد نیاز نیست. اگر تابع هیچ مقداری را بر نگرداند، مقدار پیشفرض هیچ[۱] را بازمی‌گردد. یک تابع می‌تواند نام تابع دیگری را بعنوان یک آرگومان بگیرد و نام تابع را بازگرداند. (همانند کاری که در زبانهای برنامه نویسی کاربردی انجام می‌شود)
- یک تابع با استفاده از کلمه کلیدی *def* به دنبال نام و پارامتر تابع تعریف می‌شود.
- انواع توابع پایه‌ای: ۱- توابع تعریف شده توسط کاربر[۲]، که با کلمه کلیدی *def* ایجاد می‌شود و ۲- توابع داخلی ساخته شده (توابع پیشفرض)[۳]، مانند:

dir(),
len(),
abs(),

- چهار نوع آرگومان (ورودی توابع)[۴] در توابع پایتون استفاده می‌شود:

۱) آرگومانهای موردنیاز[۵]، که آرگومانها به تابع به ترتیب مکانی صحیح منتقل می‌شوند.

۲) آرگومان کلیدی[۶]، که در فراخوانی تابع پایتون می‌توان آرگومانها را با نام پارامتر شناسایی نمود.

۳) آرگومانهای پیشفرض[۷]، که در آن آرگومان دارای مقدار پیشفرض در اعلام تابع است. مقدار پیشفرض زمانی استفاده می‌شود که هیچ مقدار در فراخوانی تابع وجود ندارد.

---

۱ None
۲ User-Defined Function
۳ Built-in Functions
۴ Argument
۵ Required Arguments
۶ Keyword Argument
۷ Default Arguments

۴) آرگومانهای طول متغیر[۱]: زمانی که می‌خواهیم پردازش اضافی نامشخص، انجام دهیم، استفاده می‌شود. در این حالت، از علامت ستاره (*) قبل از نام متغیر استفاده می‌شود.

- توصیه می‌شود که تمامی توابع مورد استفاده، همراه با تعریف تابع، مستنداتی داشته باشند.

مدیریت فایل

- فایل پایتون یک پیشفرض، در پایتون است. این امر اجازه می‌دهد تا دسترسی به فایلها از طریق سیستم عامل مستقل[۲] انجام شود.
- بازکردن فایل[۳]: برای بازکردن فایل، عبارات زیر استفاده می‌شود:

fileObject = open(file_name [, access_mode][, buffering]),

- حالتهای دسترسی مشترک عبارتند از:
  - ✓ "*r*": جهت بازکردن یک فایل برای فقط خواندن،
  - ✓ "*w*": جهت بازکردن یک فایل برای فقط نوشتن،
  - ✓ "*a*": جهت بازکردن یک فایل برای فقط افزودن،
  - ✓ "*r* +": جهت بازکردن یک فایل برای خواندن و نوشتن،
  - ✓ "*w* +": جهت بازکردن یک فایل برای نوشتن و خواندن،
  - ✓ "*a* +": جهت بازکردن یک فایل برای خواندن و نوشتن که در آن داده‌های جدید در پایان اضافه می‌شوند،
  - ✓ "*b*": جهت بازکردن یک فایل در حالت باینری،
- بستن یک فایل[۴]: برای بستن یک فایل در پایتون، از *fileObject.close()* استفاده کنید. روش *close()*، هر اطلاعات نانوشته‌ای را پاک می‌کند و شی فایل را می‌بندد. این روش زمانی استفاده می‌شود که برنامه دیگر نیازی به فایل ندارد.
- تغییر نام فایل و حذف فایل[۱]: ماژول "*os*" پایتون، روشهایی را برای تغییر و حذف فایلها فراهم می‌کند، مثالی برای وارد کردن "*os*":

---

[۱] Variable-Length Arguments

[۲] System-Independent Manner

[۳] File Opening

[۴] Closing a File

```
os.rename(“old_name.txt”, “new_name.txt”),
os.remove(file_name),
```

- خواندن یک فایل[۲]: برای خواندن یک فایل، می‌توانید از *fileObject.read([count])* ، استفاده کنید. فرمتهای مختلف خواندن:

  روش *read()*: تمام فایل را یکباره می‌خواند،

  روش *readline()*: هر بار یک خط از فایل را می‌خواند،

  روش *readlines()*: تمام خطوطِ یک فهرست از فایل را می‌خواند،

- نوشتن در فایل[۳]: روش *write()*، هر رشته را در یک فایل می‌نویسد. (البته) فایل باید باز باشد.

مدیریت استثناها

- استثناهای[۴] رایج در پایتون:

```
NameError,
TypeError,
IndexError,
KeyError,
```

- استفاده از دستور *exception* (استثنا)، که هیچ خطا یا استثنایی در آن تعریف نشده، باعث می‌شود، که هر استثنائی شامل شود.
- در نهایت، قبل از اتمام دستور *try*، عبارت همیشه اجرا می‌شود.

ماژولها

- یک ماژول[۵] یک فایل با کد پایتون است که شامل تعاریفِ توابع، کلاسها و متغیرها است. ماژولها کمک می‌کنند تا کدها را گروهبندی کرده و آنها را بهتر درک کنیم.

---

[۱] File Renaming and Deleting

[۲] Reading a File

[۳] Writing in a File

[۴] Exceptions

[۵] Module

- هر فایل پایتون (*.py*) می‌تواند بعنوان یک ماژول کار کند. اگر فایل برای اجرا نوشته شده باشد، هنگام وارد شدن اجرا می‌شود.
- ماژولها می‌توانند با استفاده از دستور *import* اجرا شوند، مثال:

```
import module1,
module2,
module3,
```

- از دستور پایتون می‌توان برای وارد کردن ویژگیهای خاصِ یک ماژول استفاده کرد ، برای مثال:

```
from module1 import name1, name3, name3,
```

- عبارت * *import* می‌تواند برای واردکردن تمام نامهای یک ماژول استفاده شود؛ مثلاً:

```
from module1 import *,
```

کلاسها

- یک کلاس مجموعه‌ای از ویژگیهایی است که اشیا کلاس را مشخص می‌کند.
- ویژگیها، عملاً اعضای داده (متغیرهای کلاس و متغیرهای نمونه) و روشها هستند.
- روش سازنده[۱]: اولین روش *__init__()*، سازنده کلاس یا روش اولیه‌سازی نامیده می‌شود که هنگامیکه یک نمونه جدید از این کلاس ایجاد می‌شود، پایتون این روش را فرا می‌خواند.
- ایجاد کلاسها:

```
class class_name: class_body
```

- ایجاد اشیا:

```
obj1 = class_name()
```

[۱] Constructor Method

پیوست ب: آموزش *NetworkX*

NetworkX یک بسته نرم‌افزاری زبان پایتون و یک ابزار منبع باز برای ایجاد، دستکاری و مطالعه ساختار، پویایی و عملکرد شبکه‌های پیچیده است. شبکه‌های پیچیده، شبکه‌هایی با ویژگیهای توپولوژیکی غیرمعمول هستند. اینها ویژگیهایی هستند که در شبکه‌های ساده مانند شبکه‌ها یا نمودارهای تصادفی رخ نمی‌دهند، اما اغلب در گرافهای واقعی همچون: ریتمهای روزانه، واکنشهای الکتروشیمیایی، آرایه‌های لیزری، شبکه‌های نورونی، آرایه‌های اتصال جوزفسون و امثال آن رخ می‌دهند.
NetworkX می‌تواند شبکه‌ها را بارگیری، ذخیره و تجزیه و تحلیل کند، شبکه‌های جدیدی تولید نماید، مدلهای شبکه را ایجاد کرده و شبکه‌ها را طراحی کند. *NetworkX*، یک ابزار مدلسازی محاسباتی است و نه ابزار توسعه نرم افزار. اولین انتشار عمومی این کتابخانه، که همه مبتنی بر پایتون بود، در آوریل ۲۰۰۵ بود. این کتابخانه می‌تواند با زبانهای غیر از پایتون مانند *C*، *C ++* و *FORTRAN* کار کند.
این کتابخانه بعنوان یک دیکشنری از دیکشنریها، با نمای گره–محور (یا گره–مرکز) از شبکه‌ای، که مبتنی بر گره‌ها و ارتباطات بین آنهاست، اجرا می‌شود. گره‌ها می‌توانند هر شی انبساط‌پذیری[۱] مانند یک رشته متنی، یک تصویر، یک شیء *XML* ، یک گراف دیگر، یک شی گره سفارشی و غیره باشند. هیچ شی پایتون نباید بعنوان یک گره استفاده شود، زیرا به معنای اختصاص آرگومانهای تابع اختیاری به توابع بسیاری است.
در هر گراف، گره و یال می‌تواند زوجهای ویژگی کلید/ارزش (مثلاً وزن، برچسب، و رنگ) را در یک دیکشنری مرتبط نگه دارد (کلیدها باید انبساط‌پذیر باشند). یالها به صورت تاپلهایی با داده‌های یال اختیاری نمایش داده می‌شوند و می‌توانند داده‌های دلخواه (مانند وزن، سری زمانی و غیره) را نگه دارند.
اگر چه *NetworkX* برای مسائل در مقیاس بزرگ که نیازمند پردازش سریع هستند، ایده‌آل نیست، اما این گزینه برای تجزیه و تحلیل شبکه‌های دنیای واقعی ایده آل است؛ زیرا:

- ✓ بسیاری از الگوریتمهای اصلی به کدهای قدیمی بسیار سریع، وابسته‌اند.
- ✓ از الگوریتمهای گراف استاندارد استفاده می‌کند.

[۱] Hashable Object

- ✓ مجموعه گسترده‌ای از فرمتهای قابل خواندن و نوشتن محلی (بومی)[۱] دارد.
- ✓ برای نصب و استفاده بر روی سیستم عامل با مستندات آنلاین و به روزرسانی قوی، آسان است.
- ✓ برای ارائه انواع شبکه‌های مختلف از قبیل گرافهای کلاسیک[۲]، گرافهای تصادفی[۳] و شبکه‌های مصنوعی[۴]، ایده‌آل است.
- ✓ از توانایی پایتون برای وارد کردن داده‌ها از منابع خارجی استفاده می‌کند.

NetworkX شامل توابع تولیدکننده گراف بسیار و امکاناتی برای خواندن و نوشتن گراف در فرمتهای بسیاری مانند: *edgelist*، *adjlist*، *gml*، *graphml*، *pajek*. و غیره است.

NetworkX بعنوان یک بسته گرافیکی طراحی نشده است. با این‌حال، قابلیتهای گرافیکی پایه را با *matplotlib* فراهم می‌کند. برای تکنیکهای پیچیده تجسم‌سازی، ترجیح داده می‌شود از بسته نرم‌افزاری *Graphviz* استفاده شود (*NetworkX* یک رابط کاربری برای آن بسته فراهم می‌کند). همچنین بسته گرافیکی در *NetworkX* هنوز با نسخه های *Python 3.0* و بالاتر سازگار نیست.

انواع گراف

```
>>> Import networkx as nx # import library
```

ایجاد گراف خالی، بدون گره و یال:

```
<<<G = nx.Graph() # create new simple undirected graphs
<<<EG = nx.empty_graph(100) # create an empty graph
<<<DG = nx.DiGraph # create a simple directed graphs
<<<MG = nx.MultiGraph() # create undirected with parallel edges
<<<MDG = nx.MultiDiGraph() # create directed with parallel edges
<<<CG = nx.complete_graph(10) # create a complete graph
<<<PG = nx.path_graph(5) # create a chain of nodes
<<<CBG = nx.complete_bipartite_graph(n1, n2) # create bipartite
<<<GG = nx.grid_graph([10, 10, 10, 10]) # arbitrary dimensional lattice
```

---

[۱] Native

[۲] Classic Graphs

[۳] Random Graphs

[۴] Synthetic Networks

دستیابی به ویژگیهای گراف:

```
>>> G.graph
```

تبدیل به گراف بدون جهت:

```
>>> G.to_undirected()
```

تبدیل به گراف جهتدار:

```
>>> G.to_directed()
```

پاک کردن گره‌ها و یالهای گراف:

```
>>> G.to_directed()
```

گره‌ها

اضافه کردن یک گره در یک زمان:

```
>>> G.add_node(1)
```

اضافه کردن فهرستی از گره‌ها:

```
>>> G.add_nodes_from([3, 4, 5]) # takes any iterable collection
```

اضافه کردن n دسته گره:

```
>>> H = nx.path_graph(10)
>>> G.add_nodes_from(H)
```

اضافه کردن یک گراف، بعنوان یک گره:

```
>>> G.add_node(H)
```

نمایش تعداد گره‌ها:

```
>>> G.number_of_nodes()
```

نمایش گره‌های گراف:

```
>>> G.nodes()
```

نمایش انواع گره‌ها:

```
>>> type(G.nodes()) # it will show class list
```

برچسب زدن مجدد گره‌ها:

```
>>> nx.relabel_nodes(G, mapping, copy = True) # mapping new labels
```

بررسی عضویت گره‌ها:

```
>>> G.has_node(1)
```

یالها

اضافه کردن یک یال:

```
>>> G.add_edge(1, 2)
```

اضافه کردن فهرستی از یالها:

```
>>> G.add_edges_from([6, 7]) # automatically add those nodes
```

اضافه کردن یالها از یک فهرست:

```
 <<<edge = (“a”, “b”)
>>> G.add_edge(*edge)
```

بررسی عضویت یال:

```
>>> G.has_edge(“a”, “b”)
```

نمایش تعداد یالها:

```
>>> G.number_of_edges()
```

حذف یک یال:

```
>>> G.remove_edge(1, 2)
```

حذف فهرستی از یالها:

```
>>> G.remove_edges_from([(1, 2), (3, 4)])
```

نمایش یالهای گراف:

```
>>> G.edges()
```

خواندن عمومی فرمت:

```
>>> nx.read_format(“path/to/file.txt”,...options...)
```

خواندن فهرست یال از فایل:

```
>>> el = nx.read_edgelist(“test.edges”, comments = “#”)
```

خواندن فهرست مجاورت از فایل:

```
>>> al = nx.read_adjlist(“test2.adj”)
```

نوشتن عمومی فرمت:

```
>>> nx.write_format(g,“path/to/file.txt”,...options...)
```

نوشتن فهرست یال:

```
>>> nx.write_edgelist(G, “newFile.edges”, comments = “#”, data = True)
```

نمایش انواع یالها:

```
>>> type(G.edges()) # it will show class list
```

گراف جهتدار

ایجاد گراف جهتدار خالی

```
>>> DG = nx.DiGraph() # creates simple directed graphs
```

اضافه کردن یالهای وزندار به *DG*:

```
>>> DG.add_weighted_edges_from([(1, 2, 2.7), (3, 1, 0.5)])
```

محاسبه درجه خروجی:

```
>>> DG.out_degree()
```

محاسبه درجه خروجی به هراه مشخصه‌ها:

```
>>> DG.out_degree(with_labels = True) # Boolean should be capitalized
```

محاسبه بعدی یا پسین *successors*:

```
>>> DG.successors(1)
```

محاسبه قبلی یا پیشین *predecessors*:

```
>>> DG.predecessors(1)
```

محاسبه همسایه‌ها:

```
>>> DG.neighbors(1)
```

تبدیل گراف جهتدار به گراف بدون جهت:

```
>>> DG.to_undirected()
```

گراف مشخصه‌دار

اضافه کردن مشخصه (ویژگی):

```
>>> G = nx.Graph(day = “Wednesday”)
```

به روز کردن مشخصه:

```
>>> G.graph[“day” ] = “Thursday”
```

اضافه کردن مشخصه به گره‌ها:

```
<<<G.add_node(1, time = “5am”) # attributes are optional
>>> G.add_nodes_from([3], time = “2am”)
```

دریافت مشخصه‌های گره‌ها:

```
>>> G.node[1]
```

اضافه کردن مشخصه به یالها:

```
>>> G.add_edges_from([(1, 2), (3, 4)], color = “blue”)
```

دریافت مشخصه‌های یالها:

```
>>> G[1][2]
```

دریافت مقدار یک مشخصه خاص:

```
>>> G[1][2][“color”]
```

گراف وزندار

اضافه کردن یال وزندار به گراف:

```
 <<<G.add_edge(1, 2, weight = 3.3)
>>> G.add_edge(3, 4, weight = 4.3)
```

محاسبه درجه گره بدون وزن:

```
>>> G.degree(1)
```

محاسبه درجه گره وزندار:

```
>>> G.degree(1, weight = “weight”)
```

محاسبه درجه همه‌ی گره‌ها:

```
>>> G.degree(weight = “weight”)
```

گراف چندگانه

ساختن گراف چندگانه:

```
>>> MG = nx.MultiGraph()
```

اضافه کردن یال به گراف *MG*:

```
>>> MG.add_weighted_edges_from([1, 2, 0.75), (1, 2, 1,25), (2, 3, 0.75)])
```

محاسبه درجات:

```
>>> MG.degree(weight = “weight”)
```

عملیات گراف کلاسیک

```
 <<<nx.subgraph(G, nbunch) # induce subgraph of G on nodes in
nbunch
```

```
 <<<nx.union(G1, G2) # graph union
 <<<nx.disjoint_union(G1, G2) # graph union/all node are different
 <<<nx.cartesian_product(G1, G2) # return Cartesian product graph
 <<<nx.compose(G1, G2) # combine graphs identifying common nodes
 <<<nx.complement(G) # graph complement
 <<<nx.create_empty_graph(G) # return an empty copy of the same
graph class
 <<<nx.convert_to_undirected(G) # return an undirected copy of G
>>> nx.convert_to_directed(G) # return a directed copy of G
```

تولیدکننده‌های گراف

```
# Using a call to one of the classic small graphs
>>> petersen = nx.petersen_graph()
>>> tutte = nx.tutte_graph()
>>> maze = nx.sedgewick_maze_graph()
>>> tet = nx.tetrahedral_graph()
# Using a (constructive) generator for a classic graph
>>> k_5 = nx.complete_graphs(10)
>>> k_3_5 = nx.complete_bipartite_graph(3, 5)
>>> barbell = nx.barbell_graph(15, 15)
>>> lollipop = nx.lollipop_graph(5, 10)
# Using a stochastic graph generator
>>> er = nx.erdos_renyi_graph(50, 0.5)
>>> ws = nx.watts_strogatz_graph(20, 2, 0.5)
>>> ba = nx.barabasi_albert_graph(50, 5)
>>> red = nx.random_lobster(100, 0.9, 0.9
```

تحلیل شبکه پایه‌ای

یافتن سازندها یا مولفه‌های مرتبط:

```
>>> nx.connected_components(G)
```

مرتب کردن گره‌ها براساس درجه گره:

```
>>> sorted(nx.degree(G).values())
```

محاسبه درجه گره مشخص:

```
>>> G.degree(1)
```

محاسبه همه درجات:

```
>>> G.degree()
```

بررسی متصل یا همبند بودن گراف:

```
>>> nx.is_connected(G)
```

محاسبه ضریب خوشه‌گی سراسری شبکه:

```
>>> nx.clustering(G)
```

محاسبه ضریب خوشه‌گی هر گره:

```
>>> nx.clustering(G, with_labels = True)
```

محاسبه ضریب خوشه‌گی گرهٔ خاص و مشخص:

```
>>> nx.clustering(G, 1)
```

یافتن کوتاهترین مسیر بین دو گره:

```
>>> nx.shortest_path(G, 1, 3)
```

یافتن طول کوتاهترین مسیر بین دو گره:

```
>>> nx.shortest_path_length(G, 3, 1)
```

یافتن توزیع درجه ورودی گراف G:

```
>>> G.in_degree()
```

یافتن توزیع درجه خروجی گراف G:

```
>>> G.out_degree()
```

محاسبه تعداد گره‌ها:

```
<<<G.order()
<<<nx.number_of_nodes(G)
>>> len(G)
```

محاسبه تعداد یالها:

```
<<<G.size()
>>> nx.number_of_edges(G)
```

یافتن قطر شبکه:

```
>>> nx.diameter(G)
```

یافتن شعاع شبکه:

```
>>> nx.radius(G)
```

یافتن مراکز شبکه:

```
>>> nx.find_cores(G)
```

معیارهای مرکزیت

محاسبه مرکزیت درجه‌ای:

```
>>> nx.degree_centrality(G)
```

محاسبه مرکزیت میانی:

```
>>> nx.betweenness_centrality(G)
```

محاسبه مرکزیت نزدیکی:

```
>>> nx.closeness_centrality(G)
```

محاسبه مرکزیت برداریژه:

```
>>> nx.eigenvector_centrality(G)
```

رسم گراف

```
>>> import matplotlib.pyplot as plt # Can use GraphViz
```

پاک کردن گراف قبلی:

```
>>> plt.clf()
```

رسم یک گراف:

```
>>> nx.draw(G)
>>> nx.draw_random(G)
>>> nx.draw_circular(G)
>>> nx.draw_spectral(G)
```

نمایش یک گراف:

```
>>> plt.show() # to show the file
```

ذخیره گراف:

```
>>> plt.savefig(“myFig.png”) # save as png file
```

بستن فایل:

```
>>> plt.close()
```

استخراج سازندها یا مولف‌های اصلی گراف *G*:

```
>>> nx.connected_component_subgraphs(G) # graph should be undirected
```

بسته الگوریتمها (الگوریتمهای *NetworkX*)

- *Bipartite*
- *Block*
- *Boundary*
- *Centrality (package)*
- *Clique*
- *Cluster*
- *Components (package)*
- *Core*
- *Cycles*
- *Dag*
- *Distance measures*
- *Ow (package)*
- *Isolates*
- *Isomorphism (package)*
- *Link analysis (package)*
- *Matching*
- *Mixing*
- *MST*
- *Operators*
- *Shortest paths (package)*
- *Smetric*

خواندن و نوشتن

- *Adjacency list*
- *Multiline adjacency list*
- *Edge list*
- *GEXF*
- *GML*
- *Pickle*
- *GraphML*
- *LEDA*
- *YAML*

- *SparseGraph6*
- *Pajek*
- *GIS shapefile*

واژه نامه

فارسی–انگلیسی

الف

m–قطاع	m-Slice
m–هسته	m-Core
اُ اس ایکس	OSX
ابرِ کلمه	Word Clouds
اَبَرداده	Metadata
اَبَرسازند، غول‌مولفه	Super-Component
ابرلینکها	Hyperlinks
ابزار روش‌شناسی	Methodological Tool
ابزار مدل‌سازی و محاسباتی شبکه	Computational Network Modeling Tool
اثر دنیای کوچک	Small-World Effect
اثر زنجیره‌ای	Chain Effect
اثربخشی	Effectiveness
اثرگذاری وابسته	Related Influence
اجتماع گرافهای مجزا	Disjoint Union of Graphs
اجتماعات متمایز غیرهمپوشان	Disjoint Non-Overlapping Communities
اجتماعات همپوشان	Overlapping Communities
اجتماعی کردن	Socialise
احتمال گذار	Transition Probability Matrix (TPM)
ادبیات حمایت اجتماعی	Social Support Literature
ارتباط بلاواسطه	Immediate Connections
ارتباط جمعی	Massa Communication
ارتباطات ایمیلی	Email Communications
ارتباطات چهره به چهره	Face-to-Face Communications

Indirect Ties	ارتباطات غیرمستقیم
Boolean Value	ارزش بولین
Cultural Values	ارزشهای فرهنگی
Fitness of Block Models	ارزیابی تناسب مدلهای بلوک
Lazy Evaluation	ارزیابی کاهلانه
Exceptions	استثناها
Solely a Proposed Metaphor	استعاره پیشنهادی
Inference	استنتاج
Network Robustness and Vulnerability	استواری و آسیب‌پذیری شبکه
Archival Records	اسناد موجود در آرشیو
Immutable Objects	اشیاء تغییرناپذیر
Mutable Objects	اشیاء ناپایدار، اشیاء تغییرپذیر
Comments	اظهار نظر، نظرات
Information Validity	اعتبار اطلاعات
Trustor	اعتماد کننده
Mutual Trust	اعتماد متقابل
Horizon of Observability	افق قابل مشاهده، افق دید
Persuasion	اقناع، ترغیب
Early Majority	اکثریت اولیه
Late Majority	اکثریت ثانویه
Administrative Privileges	امتیازات ادمین
Expansion and Contraction of Communities	انبساط و انقباض اجتماعات
Free Vs. Fixed Choice	انتخاب آزاد در مقابل انتخاب معین
Information Diffusion	انتشار اطلاعات
Diffusion of Innovation	انتشار نوآوری
Sign Shifting	انتقال علائم
Transferring of Non-Materials	انتقال غیرمواد
Static Communities	انجمنهای ایستا
Dynamic Communities	انجمنهای پویا
Link Communities	انجمنهای پیوندی
Information Fusion Communities	انجمنهای نفوذ اطلاعاتی، اجتماعات نفوذ اطلاعاتی
Isolation	انزوا، قرنطینه
Anthropology	انسان‌شناسی، مردم‌شناسی
Structural Cohesion	انسجام ساختاری

انفجار داده	Explosion of Data
انقطاع	Break
اوریون	Mumps
اوزان یال	Edge Weights
آبشار اطلاعات	Information Cascades
آرگومان کلیدی	Keyword Argument
آرگومان، ورودی توابع	Argument
آرگومانهای پیشفرض	Default Arguments
آرگومانهای طول متغیر	Variable-Length Arguments
آرگومانهای موردنیاز	Required Arguments
آستانه مواجهه	Threshold of Exposure
آلوده، ابتلایافته، مبتلا	Infected
آنالیز انسجام گروهی	Cohesive Group Analysis
آنفولانزای حاد پرندگان	H5N1
الگوریتم جستجوی سطحی	Breath-First Search (BFS)
الگوریتم جستجوی عمقی	Depth-First Search (DFS)
الگوریتم دوبخشی طیفی	Spectral Bisection Algorithm
الگوریتم کراسکال	Kruskal's Algorithm
الگوریتمهای چندسطحی	Multilevel Algorithms
الگوریتمهای هیوریستیک	Heuristic Algorithms
الگوهای ارتباطات	Tie Patterns
الگوهای امتزاج، الگوهای اختلاط	Mixing Patterns
الگوهای رای‌گیری	Voting Patterns
الگوی اختلاط ناهماهنگ، الگوی اختلاط ناهمسان	Disassortative Mixing Pattern
الگوی اختلاط هماهنگ، الگوی اختلاط همسان	Assortative Mixing Pattern
الگوی اختلاط، الگوی جذابیت	Assortativity

ب

بافتار اجتماعی	Social Tissue
باندها	Gangs
بایت‌کد	Bytecode
بخش، اِفراز	Partition

بخش‌بندی بازار	Market Segmentation
بخش‌بندی سطح–ساختار	Level-Structure Partitioning
بخش‌بندی شبکه	Network Segmentation
بخش‌بندی قطعی	Deterministic Compartmental Model
بخش‌بندی گراف	Graph Partitioning
بخش‌بندی مدارات الکترونیکی	Partitioning Electronic Circuits
بدون مقیاس	Scale-Free
براکت منحنی‌شکل	Curly Braces
بردار رفتار پذیرش	Vector of Adoption Behavior
بردار ویژه تصادفی	Stochastic Eigenvector Scoring
برش کمینه	Min-Cut
برشهای نرمال	Normalized Cuts
برنامه سفارشی	Custom Application
برنامه‌های تخصصی	Specialized Applications
بستار	Closure
بستن یک فایل	Closing a File
بک–لینکها	Backlinks
بلمن فورد	Bellman-Ford
بولی	Boolean
به صورت نمایی	Exponentially
بهینه‌سازی مدولاسیون	Modularity Optimization
بی‌ثبات‌سازی	Destabilizing
بی‌جهت، بی‌طرف	Directionless
بینابینی جریان فعلی یال	Current-Flow Edge Betweenness
بینابینی کروی یال	Geodesic Edge Betweenness
بینابینی گام–تصادفی یال	Random-Walk Edge Betweenness

پ

پالایش ائتلافی، پالایش مشارکتی	Collaborative Filtering (CF)
پالایش مبتنی بر محتوا	Content-Based Filtering (CBF)
پخش شایعات	Rumor Propagation
پذیرش نوآوری	Adopt the Innovation
پذیرندگان آغازین	Early Adopters

پذیرندگان بالقوه	Potential Adopters
پذیرندگان نهایی	Late Adopters
پذیرنده	Adopter
پروس (نام یک منطقه)	Prussia
پستهای آغازین	Opening Post
پشتیبانی درون‌–سازی	Built-in Support
پلتفرمهای بزرگ	Main Platforms
پلها، رابطها	Connectors
پویاییهای شبکه	Network Dynamics
پویش	Campaign
پیاده‌سازی برنامه‌های SNA توصیفی	Descriptive SNA Applications
پیاده‌سازی برنامه‌های SNA توضیحی	Explanative SNA Applications
پیامهای ارتباطی	Communication Messages
پیدا کردن اجتماعات	Finding Communities
پیروی، دنبال کردن	Follow
پیش‌بینی پیوند	Link Prediction
پیشنهادیه	Proposals
پیکره‌بندیهای پایدار	Stable Configurations
پیمایش سطحی، پیمایش اول–سطح	Breadth-First Traversal (BFT)
پیمایش عمقی، پیمایش اول–عمق	Depth-First Traversal (DFT)
پیمایش گراف	Graph Traversals
پیوست ترجیحی	Preferential Attachment
پیوند اجتماعی	Social Tie
پیوند چندگانه	Multiplex
پیوند کامل	Complete Linkage
پیوند متوسط	Average Linkage
پیوند مرکزی	Centroid Linkage
پیوند نمونه‌گیری	Sampled Linkage
پیوند همه‌جانبه	All-Pairs Linkage
پیوند یگانه	Uniplex
پیوندهای دووَند نامتقارن	Asymmetric Dyad Ties
پیوندهای مفقوده یا نامشهود	Missing or Unobserved Links

ت

تابع آماده	Ready-Made Function
تابع سرشماری سه‌وَند	Triadic Census Function
تابع سیگموئید	Sigmoid Function
تابع کاهشی	Reduce Function
تالار گفتگو، انجمن مباحثه	Discussion Forum
تجزیه و تحلیل شبکه کامل	Complete Network Analyses
تجزیه و تحلیل پیوند	Link Analysis
تجزیه و تحلیل جمع–محور، تجزیه و تحلیل جمعانه	Sociocentric Analysis
تجزیه و تحلیل سه‌وَندی، تجزیه و تحلیل سه‌وَندانه	Triadic Analysis
تجزیه و تحلیل شبکه اجتماعی	Social Network Analysis
تجسم اطلاعات	Visualize Data
تجسم بخشی دادها، تصویرسازی داده‌ها	Data Visualization
تحلیل تاریخچه رویداد	Event History Analysis
تحلیل شبکه پویا	Dynamic Network Analysis
تحلیلگران جامعه‌شناسی	Sociometric Analysts
تراگذری، انتقال‌پذیری، تریا، متعددی	Transitivity
ترانهاده ماتریس	Matrix Transpose
ترجیحات پیشنهادی	Suggested Preferences
ترکیب و تجزیه اجتماعات	Merging and Splitting of Communities
تصویر لحظه‌ای	Snapshot
تعامل اکولوژیکی	Ecological Nnteractions
تعامل بین‌گونه‌ای	Between-Species Interaction
تعامل درون‌گونه‌ای	Within-Species Interaction
تعاملات اجتماعی	Social Interaction
تغییر دائمی	Discontinuous Change
تغییر نام فایل و حذف فایل	File Renaming and Deleting
تفریق ماتریسی	Matrix Subtraction
تقاطع سطری–ستونی	Column-Row Intersection
تقسیمات طبیعی	Natural Divisions
تکنیک رتبه‌بندی صفحه	Page Ranking Technique
تکنیکهای تجزیه و تحلیل استاندارد	Standard Analysis Techniques

تلاقی	Incident
تماس اجتماعی	Social Contact
تمامیت	Completeness
تُنُک، پراکنده	Sparse
تنوع	Diversity
توابع تعریف شده توسط کاربر	User-Defined Function
توابع تولیدکننده گراف	Graph Generator Functions
توابع داخلی ساخته شده، توابع پیشفرض	Built-in Functions
توالی روابط	Sequence of Relationships
توری دو بعدی	Two-Dimensional Grid
توزیع اندازه مولفه‌ها	Distribution of the Component Sizes
توزیع قانون قدرت	Power Law Distribution
توزیع نرمال	Normal Distribution
توسعه ابزار نرم‌افزار	Software Tool Development
تولیدکننده شی	Object Generator

ج

جامعه‌نگار، جامعه‌نما، سوسیوگرام	Sociogram
جداکننده رئوس	Vertex-Separator
جدایشی	Divisive
جدول هش، جدول درهم‌سازی	Hash Table
جرم بحرانی	Critical Mass
جریان اثرگذاری یا نفوذی	Flow of Influence
جزایر	Islands
جعبه نمودارها	Box Plots
جمع ماتریسی	Matrix Addition
جمع‌آوری داده‌ها در شبکه‌های اجتماعی	Data Collection in Social Networks
جنبش پنج ستاره	Five Star Movement
جهشها، پرشها	Hops

چ

چارچوبهای هم‌پیوند	Conjoint Frameworks
چرخه‌ای، دوره‌ای	Cyclic

چرخه، دور	Cycle
چگالی گردشده	Rounded Density
چند پارادایمی	Multi-Paradigm
چند عضویتی بودن گره	Node Multi-Membership
چند مشارکتی بودن	Multi-Participation
چندحالته، چندمُد	Multimode
چندگانگی	Multiplexity
چندگانگی خطوط، کثرت خطوط	Line Multiplicity
چندگانگی، کثرت	Multiplicity
چندگانه	Polytree
چوله راست	Right-Skewed
چیدمان صفحه تعاملی	Interactive Plotting Pane

ح

حاشیه‌ای از گره‌ها	Periphery of Nodes
حاشیهٔ شبکه	Network Periphery
حافظه بلند مدت	Long-Term Memory
حالت	State
حداکثر برون‌مرکزی	Maximum Eccentricity
حریم‌خصوصی	Privacy
حلقه	Loop
حیاتی	Vitality

خ

خُرد	Micro
خرده‌فروشی	Retail Marketing
خزنده‌های وب	Web Crawlers
خطای اندازه‌گیری	Measurement Error
خطوط چندگانه	Multiple Lines
خواص ساختاری/آماری	Statistical Properties/Structural
خواص شبکه‌های اجتماعی	Properties of Social Networks
خواندن یک فایل	Reading a File
خوش‌–ارتباط، ارتباطات قویتر	Well-Connected

خوشه‌بندی	Clustering
خوشه‌بندی سلسله‌مراتبی	Hierarchical Clustering
خوشه‌بندی گراف	Graph clustering
خوشه‌های طولانی	Elongated Clusters
خوشه‌های کروی	Spherical Clusters
خوشه‌های نامنظم	Irregular Clusters
خوشه‌های هسته‌ای	Core Clusters

د

داده‌های جامعه‌سنجی	Sociometric Data
داده‌های جمعیت‌شناختی	Demographic Data
داده‌های رابطه‌ای سیستماتیک	Systematic Relational Data
داده‌های وابستگی، داده‌های وابستاری	Affiliation Data
دانه رشته‌ای، بذر رشته‌ای	String Seed
دانه‌بندی	Granularity
دایره اجتماعی، حلقه اجتماعی	Social Circle
دایره واژگان، بیان‌افزار	Vocabulary
دایکسترا	Dijkstra
درجه خروجی	Out-Degree
درجه متوسط	Average Degree
درجه ورودی	In-Degree
درجه هسته	Core Number
درجه‌بندی در مقابل رتبه‌بندی کامل	Ratings Vs. Complete Rankings
درخت پوشا کمینه، درخت فراگیر کمینه	Minimum Spanning Trees
درخت فراگیر، درخت پوشا	Spanning Tree
درخواستها	Queries
دستگاه‌های عصبی مرکزی	Central Nervous Systems
دستور	Command
دسته‌بندی مبتنی بر ویژگی	Feature-Based Classification
دقت اطلاعات	Information Accuracy
دلتای کرونکر	Kronecker Delta
دنباله	Trail
دنباله ترتیبی پایدار، دنباله ترتیبی تغییرناپذیر	Immutable Ordered Sequence

دنیادیده	Cosmopolitans
دو طرفه	Bidirectional
دوبخشی	Dichotomous
دورسته‌ای	Dichotomic
دوستان پیشنهادی	Suggested Friends
دوسویگی	Reciprocity
دو–مُد	Bimodal Network
دووَند، دوگاه	Dyad
ده نفرِ اول، ده‌تای اول	Top Ten
دیرپذیران، واماندگان	Laggards

ذ

ذخیره، پشته	Stack

ر

رابط برنامه‌نویسی نرم افزارها	APIs
رابطه اجتماعی پایدار	Stable Social Relationships
رابطه باینری	Binary Relationship
رابطه متقابل	Reciprocated Relationship
رابطهٔ یکطرفه، رابطهٔ یکسویه	One-Way Relationship
راس	Vertices
راس ارزشدار، راس برچسبدار	Labeled Vertex
راس برشی	Cut-Vertex
راس برگ	Leaf Vertex
راس فروچاله	Sink
راه میانبر دوربرد، راه میانبر دوررس	Long-Range Shortcuts
رتبه‌بندی مبتنی بر پیوند	Link-Based Ranking
رتبه–صفحه	PageRank
رخدادهای زیست پزشکی	Biomedical Events
ردیابی	Crawl
رسانه، واسطه	Medium
رشد جمعیت	Growth of Population
رفتار گروهی، رفتار گله‌ای، رفتار توده‌ای	Herd Behavior

رله‌ها، واسطه‌ها	Relays
رنگ‌آمیزی گراف	Graph Coloring
روابط	Relation
روابط اجتماعی صامت	Silent Social Relations
روابط ضعیف	Weak Relations
روابط طولانی مدت	Longitudinal Relations
روابط مرکب	Compounding Relations
روابط منسجم	Cohesive Relations
روابط میان افراد	Individual Evaluation
روانشناسی اجتماعی	Social Psychology
روش پیوند–تکی	Single-Linkage Method
روش پیوند–کامل	Complete-Linkage Method
روش خوشه‌بندی پیوند–متوسط	Average-Linkage Clustering Method
روش سازنده	Constructor Method
روشهای مبتنی بر راستنمایی	Likelihood-Based Methods
رویکرد حریصانه	Greedy Approach
رویکرد مقیاس‌پذیر	Scalable Approach
رهاشده، بهبودیافته	Recovered
رهبران عقیده	Opinion Leaders
ریشه‌دار	Rooted

ز

زبان تفسیری	Interpreted Language
زبانهای برنامه‌نویسی روبی	Ruby Programming Languages
زمینه، بافتار	Context
زنجیره شکل	Chain-Like
زوج مرتبهای گره–ارزش	Node-Value Pairs
زیرخوشه	Sub-Clusters
زیرشبکه کامل	Complete Subnetwork
زیرگرافِ چگال، زیرگرافِ متراکم	Dense Subgraphs
زیرگراف غول‌پیکر	Giant Subgraph
زیرگرافِ منتجهٔ بیشینِ همبند	Connected Maximal Induced Subgraph
زیرگروهبندی	Subgrouping

زیرگروههای منسجم	Cohesive Subgroups

س

ساختار اجتماع	Community Structure
ساختار بدنه دانش	Construction of Knowledge Body
ساختار درخت–مانند	Tree-Like Structure
ساختار رابطه‌ای	Relational Structure
ساختارهای داده استاندارد	Standard Data Structures
ساختارهای محفل–مانند	Clique-Like Structures
ساختارهای نقش	Role Structures
ساخته انسان، دست‌ساز انسان	Man-Made Networks
سارس	SARS
سازگار–ابتلاء یافته–سازگار	Susceptible-Infected-Susceptible
سازگار–آلوده	Susceptible-Infected
سازگار–آلوده–رهایافته–سازگار	Susceptible-Infected-Recover-Susceptible
سازگاری، الفت	Affinity
سازه k–هسته	k-Core Construct
سازه‌های اجتماعی، آمیزه‌های اجتماعی	Social Compositions
سامانه‌های توصیه دوستی	Friend Recommender Systems
سرخجه	Rubella
سرخک	Measles
سرمایه اجتماعی	Social Capital
سطح بحرانی	Critical Level
سلامت–ابتلاء–رها	Susceptible-Infected-Recovered
سلسله‌مراتبی	Hierarchy
سنجه‌های شبکه، معیارهای شبکه	Network Measures
سوالات ترتیبی در مقابل فراخوان آزاد	Roster vs. Free Call
سه‌تایی	Ternary
سه‌گانه‌های بسته، سه‌تاییهای بسته	Closed Triples
سه‌وَند کامل	Complete Triad
سه‌وَند همپوشان، سه‌وَند متداخل	Overlapped Triad
سه‌وَند، سه‌گاه	Triad
سه‌وَندهای ناهمبند	Unconnected Triads

سیستم اقتصاد جهانی	World Economic System
سیستم خاص مرجع	Special System of Reference
سیستم دوگانه مخمر	Yeast Two-Hybrid System
سیستم عامل مستقل	System-Independent Manner
سیستمهای بهم متصل، سیستمهای همبند	Interconnected Systems
سیستمهای توصیه‌گر	Recommendation Systems
سیستمهای فیلترینگ مشارکتی	Collaborative Filtering Systems
سیگنالینگ	Signaling

ش

شاخص کارت	Kart Index
شاخص لایچ–هولم نیومن	Leicht-Holme-Newman Index
شبکه اجتماعی قهرمان	Hero Social Network
شبکه ارتباطات اجتماعی	Social Communication Network
شبکه ارزش‌دار، شبکه ارزش‌گذاری شده	Valued Network
شبکه اقلام–کاربر	User-Item Network
شبکه باشگاه–به–باشگاه	Club-to-Club Network
شبکه بین‌المللی	International Network s
شبکه توییتر ساختگی	Fabricated Twitter Network
شبکه فرد–به–فرد	Person-to-Person Network
شبکه فرد–محور، شبکه فردانه	Ego-Centered Network
شبکه فضایی	Spatial Network
شبکه فنآوری جهانی	Global Technological Network
شبکه قهرمانی	Heroic Network
شبکه کاراکترهای کتابهای طنز	Network of Comic Book Characters
شبکه مشتق‌شده	Derived Network
شبکه‌های اجتماعی آنلاین	Online Social Networks
شبکه‌های اجتماعی حیوانات	Animal Social Networks
شبکه‌های استنادی	Citation Networks
شبکه‌های اطلاعاتی	Information Networks
شبکه‌های اعتماد	Trust Networks
شبکه‌های بیولوژیکی	Biological Networks
شبکه‌های پیچیده	Complex Networks

شبکه‌های تحقیقاتی	Research Networks
شبکه‌های تعامل پروتئین به پروتئین	Protein-Protein Interaction Networks
شبکه‌های تماس میان–فردی	Interpersonal Contact Networks
شبکه‌های جهان کوچک	Small-World Networks
شبکه‌های جهت‌دار، وزندار و در حال تکامل	Directed, Weighted, and Evolving Networks
شبکه‌های خوشه‌پذیر	Clusterable Networks
شبکه‌های دو–حالته، شبکه‌های دو–بخشی	Two-Mode Networks
شبکه‌های سه–حالته	Tripartite Networks
شبکه‌های شباهت	Networks of Similarities
شبکه‌های غذایی	Food Webs
شبکه‌های فناوری	Technological Networks
شبکه‌های مخرب	Malicious Networks
شبکه‌های مصنوعی	Synthetic Networks
شبکه‌های نیروی برق، شبکه‌های قدرت	Power Grids
شبکه‌های وابستگی، شبکه‌های وابستاری	Affiliation Networks
شبکه‌های هم–استنادی	Co-Citation Networks
شبکه‌های همبستگی	Networks of Correlations
شبکه‌های هم–تألیفی	Coauthorship Networks
شبکه‌های هم–شرکتی	Co-Participation Networks
شبکه‌های همکارانه	Corporate Networks
شبکه‌های یک–حالته	One-Mode Networks
شدت	Intensity
شرکتهای نوپا، استارتاپها	Startups
شعاع	Radius
شمارش تعداد آراء	Upvote
شمارش موتیف	Motif Count
شناسایی اجتماعات، انجمن یابی	Community Detection
شناسایی زمانی انجمن، شناسایی موقت انجمن	Temporal Community Detection
شناسه	Identity
شناور	Float
شنونده منفعل	Passive Listener
شهود ساختاری	Structural Intuition
شی انبساط‌پذیری	Hashable Object
شی‌گرای	Object-Oriented

شیوع اجتماعی، اشاعه اجتماعی	Social Contagion

ص

صعودی	Ascending

ض

ضرب تانسور گرافها	Tensor Product of Graphs
ضرب ترتیبی گرافها، لکسیکوگرافیک گرافها	Lexicographic Product of Graphs
ضرب دکارتی گرافها	Cartesian Product of Graphs
ضرب قوی گرافها	Strong Product of Graphs
ضرب ماتریسی	Matrix Multiplication
ضرب ماتریسی سریع	Fast Matrix Multiplication
ضریب جاکارد	Jaccard Coefficient
ضریب خوشه‌گی	Clustering Coefficient
ضریب خوشه‌گی کل شبکه	Whole-Network Clustering Coefficient

ط

طاعون بزرگ لندن	Great Plague of London
طبقه‌بندی بدون نظارت	Unsupervise Task
طبقه‌بندی پذیرش	Adoption Categories
طبقه‌بندی تحت نظارت	Supervised Classification
طول عمر انتشار	Lifetime of a Diffusion

ع

عامل کانونی	Focal Actor
عبارات	Statements
عدم شباهت	Dissimilarity
عضو ضعیف، عضو کم‌کار	Light Members
عکسهای متوالی	Successive Snapshots
عمق جستجو	Depth of Search
عملیات اولیه	Elementary Operations

عملیات بازنویسی گراف	Graph Rewrite Operations
عملیات پایه‌ای ماتریسی	Basic Matrix Operations
عملیات دودویی	Binary Operations
عملیات ماتریسی	Matrix Operations
عملیات یکتایی، عملیات یگانی	Unary Operations

غ

غیرشبکه‌ای	Nonnetwork

ف

فاصله کروی، ژئودزیک	Geodesic Distance
فاکتور تعدیل	Damping Factor
فراخوانی	Call
فرآیند بیولوژیکی	Biological Process
فرآیندهای پویای شبکه–پایه	Network-Based Dynamic Process Models
فرستنده	Sender
فرو اُفت شبکه، تباهیدگی شبکه	Degeneracy of the Network
فرو ریختن، تکه‌تکه شدن	Fragmentation
فروریخته	Collapsed
فلج اطفال	Polio
فنون تجسم چندبعدی	Multidimensional Visualization Techniques
فورد– فالکرسون	Ford-Fulkerson
فهرست چندتایی از گره‌ها	List of Tuples of Nodes
فهرست مجاوران گره‌ها	Nodes’ Adjacency Lists
فهرست مجاورت	Adjacency List
فهرست یال	Edge Lists

ق

قابل حمل، پرتابل	Portable
قابلیت اطمینان اطلاعات	Information Reliability
قابلیت دسترسی	Reachability
قابلیت دسترسی اشخاص	Person’s Reachability

قدم، سیر، گشت	Walks
قطبها	Hubs
قطر شبکه	Network Diameter
قوم‌نگاری، مردم‌شناسی	Ethnography
قهرمان–به–قهرمان	Hero-to-Hero
قهرمان–به–کمیک	Hero-to-Comic

ک

k–اتصالی	k-Connected
کاردینالیتی	Cardinality
کارکردگرایی	Coadaptation
کارگزاری	Correspondence
کتاب کُمیک	Comic book
کدهای قدیمی	Legacy Code
کلاف درهم	String
کلان	Macro
کلید دیکشنری	Dictionary’s Keys
کم–ارتباط، ارتباطات ضعیفتر	Ill-Connected
کمانها	Arcs
کمیک–به–قهرمان	Comic-to-Hero Network
کنشگر، عامل، فرد، بازیگر، فعال، نهاد، موجودیت	Actor
کوچکترین متوسط طول مسیر	Smallest Average Path Length

گ

گذرگاههای شبکه	Network Traversals
گراف اجتماعی دنیای مارول	Marvel Universe Social Graph
گراف اِردوش رینی	Erdos-Renyi Graph
گراف استاندارد	Standard Graph
گراف با k راس متصل	k-Vertex-Connected Graph
گراف بادبادک مانند	kite-like graph
گراف باراباشی–آلبرت	Barabasi-Albert Graph
گراف تصادفی–لوبستر	Random-Lobster Graph
گراف چندگانه	Multigraph

Dual Graph	گراف دوگان
Mathematical Graph	گراف ریاضی
Ego Graph	گراف فردانه، گراف فرد–محور، گراف خود–محور
Complete Graph	گراف کامل
Connected and Disconnected Graph	گراف متصل و منفصل
Orthogonal Graph	گراف متعامد
Mixed Graph	گراف مختلط
Planar Graph	گراف مسطح
Grid-Based Graph	گراف مشبک
Regular Graph	گراف منظم
Ladder Graph	گراف نردبان
Half-Edges, Loose Edges Graph	گراف نیمه یال یا سست یال
Watts-Strogatz Graph	گراف واتس–استروگاتز
Weakly and Strongly Connected Graph	گراف همبند ضعیف و قوی
Random Graphs	گرافهای تصادفی
Bipartite Graphs	گرافهای دو–بخشی
Biased Graph	گرافهای سوگیرانه
Classic Graphs	گرافهای کلاسیک
Finite and Infinite Graphs	گرافهای متناهی و نامتناهی
Lattices Graphs	گرافهای مشبک
Normalized Graphs	گرافهای نرمال
Distinct Groups	گروه‌های متمایز
Activated	گره فعال
Single Broadcast Node	گره مخابره کنندهٔ تنها
Long-Term Core Nodes	گره‌های هسته بلندمدت
Alters	گره‌های دگر–خود، گره‌های دگرانه
Eccentricity	گریز از مرکز، برون‌مرکزی
Scholar-Scholar Citation Records	گزارشهای استنادی محقق–محقق
Extensiveness	گستردگی
Spread of Violence	گسترش خشونت
Deployed Applications	گسیل برنامه‌های کاربردی
Surfing the Web	گشت و گذار در وب، وبگردی
Bottleneck	گلوگاه، تنگا
Receiver	گیرنده

ل

لایه‌بندی	Stratification
لگاریتم رتبهٔ شبکه	Logarithm of the Order of the Network
لنگر، سرپل	Anchors

م

ماتریس پادمتقارن	Skew-Symmetric Matrix
ماتریس تصویر	Image Matrix
ماتریس جایگشت	Matrix Permutation
ماتریس ستونی	Column Matrix
ماتریس سطری	Row Matrix
ماتریس صفر	Null Matrix
ماتریس فاصله، ماتریس مسافت	Distance Matrix
ماتریس قطری	Diagonal Matrix
ماتریس لاپلاس	Laplacian Matrix
ماتریس متقارن	Symmetric Matrix
ماتریس مثلثی	Triangle Matrix
ماتریس مجاورت	Adjacency Matrix
ماتریس مربعی	Square Matrix
ماتریس همانی	Identity Matrix
ماتریس–اجتماعی	Sociomatrix
ماتریس‌گراف جهتدار، ماتریس دایگراف	Digraph Matrix
ماژول	Module
متابولیسم	Metabolic
متعادل	Balanced
مثلثهای کاملاً بسته	Completely Closed Triangles
مجاورت ساده	Simple Adjacency
مجموعه ابزار گرافیکی مبتنی بر پایتون	Python-Based Graph Toolkit
مجموعه اشیا	Object Set
مجموعه‌های ناهمنام، مجموعه‌های مخالف	Opposite Sets
مجموعه‌های نقش	Role Sets
محافل همپوشان	Overlapping Cliques

محدودیت تجمعی	Aggregate Constraint
محدودیتهای اجتماعی	Social Constraints
محدودیتهای دووندی، محدودیتهای دوگاهی	Dyadic Constraint
محدودیتهای شناختی	Cognitive Limit
محرکها	Motives
محفل بیشین	Maximal Clique
محفل بیشینه	Maximum Clique
محفلی	Cliquishness
محققان هاروارد	Harvard Researchers
مختلط	Complex
مداخله	Intervention
مدل جریان چند مرحله‌ای	Multistep Flow Model
مدل جریان دو مرحله‌ای	Step Flow Model
مدلسازی بلوکه‌ای	Blockmodeling
مدلسازی شبکه‌های آنلاین در مقیاس گسترده	Modeling Large-Scale Online Networks
مدلهای بلوکه‌ای	Blockmodels
مدلهای گرافیکی	Graphical Models
مدلهای واگیری	Contagion Models
مدلهای همه‌گیری زمان پیوسته	Continuous-Time Epidemic Models
مدولار بودن، پیمانه‌ای بودن	Modularity
مدیریت بحران	Emergency Management
مراکز تحقیقاتی شبکه	Centers for Network Research
مردم‌شناسان منچستر	Manchester Anthropologists
مرکزیت بردار ویژه	Eigenvector Centrality
مرکزیت درجه‌ای	Centrality Degree
مرکزیت میانی، مرکزیت بینابینی	Betweenness Centrality
مرکزیت نزدیکی	Centrality Closeness
مسائل در مقیاس بزرگ	Large-Scale Problems
مستعد، سازگار	Susceptible
مسطحی، تختی	Planarity
مسیر	Path
مسیرهای اویلری	Eulerian Paths
مسیرهای عصبی	Neurological Pathways
مسیرهای همیلتون	Hamiltonian Paths

مسئله NP کامل	NP-Complete
مسئله فروشندهٔ دوره‌گرد	Traveling Salesman Problem
مضمون	Thread
معتمد	Trustee
معروفیت، محبوبیت	Popularity
معیارهای هم‌ارز	Equivalence Measure
مقدار ویژه غالب	Dominant Eigenvector
منابع خارجی	Outside Sources
مناظره	Debate
منبع انتشار	Source of the Diffusion
منبع و مقصد یک رابطه	Source and Destination of a Relationship
منحنی انتشار	Diffusion Curve
منحنی زنگوله‌ای	Bell-Shaped Curves
منطقهٔ مرتبهٔ اول	First-Order Zone
مواجهه شبکه	Network Exposure
موقعیت	Position
موقعیت افقی ریشه	Horizontal Location of the Root
موقعیت عمودی ریشه	Vertical Location of Root
موقعیتهای ساختاری گره‌ها	Structural Positions of Nodes
مولفه، سازند	Component
مولفه‌های متصل، سازندهای همبند	Connected Components
مولفهٔ ایزوله‌ای، مولفهٔ مجزا، سازند تنها	Isolated Component
مهره‌های اتصال مرزی	Boundary Spanners
مؤلفه‌های ویژه	Eigencomposition
میانجی‌گری، واسطه‌گری	Intermediary
میانگین کوتاهترین مسیر	Average Shortest Path
میانی	Meso
میزان مواجهه، در معرض بودن	Amount of Exposure
میزبانها	Hosts

ن

نابدیهی، ناصفر، نامعلوم	Nontrivial
ناسازگاری، انزجار	Antipathy

نامتوازن	Skewed
ناوبری	Navigation
نرخ پذیرش	Adoption Rate
نرم‌افزار منبع باز	Open Source Software
نزدیکترین مجاورت	Nearest Neighbor
نزولی	Descending
نصب در سطح کاربر	User-Level Installation
نظاره بر رفتار کاربر	Monitoring User Behavior
نظرسنجیهای آنلاین	Online Surveys
نظریه تعادل ساختاری	Structural Balance Theory
نظریه کنشگر- شبکه	Actor-Network Theory
نظریه گراف علامتدار	Signed Graph Theory
نقاط اوج انفرادی	Individual Tipping Points
نقاط پایانی	Endpoint
نقاط نوسانی	Fluctuation Points
نقشهای موذیانه، نقشهای خرابکارنه	Subversive Roles
نقطه اشباع	Saturation Point
نقطه اوج	Tipping Point
نگاشت پایتون	Python Map
نمادها	Symbols
نمایش داده‌های شبکه	Network Data Representation
نمایه‌های فیزیکی	Physical Representations
نمودارهای پراکنده	Scatter Plots
نمونه‌گیری از پیوند	Link Sampling
نمونه‌گیری از گره	Node Sampling
نمونه‌گیری پاسخگو-محور	Respondent-Driven Sampling
نمونه‌گیری داده‌ها	Data Sampling
نمونه‌گیری زنجیره‌ای	Chain Sampling
نمونه‌گیری گلوله‌برفی	Snowball Sampling
نوشتن در فایل	Writing in a File
نیمه‌حاشیه‌ای	Semi-Periphery

Affiliation	وابستاری، وابستگی سازمانی
Joint Affiliation	وابستاری مشترک
Import Data	واردکردن داده‌ها
Social Networking Terminology	واژگان شبکه‌های اجتماعی
Brokerage	واسطه‌گران، کارگزاران
Mediate	واسطه‌گری
Third Party	واندگرانه، شخص ثالث
World Wide Web	وب گسترده جهانی
Social Media Websites	وب‌سایتهای رسانه‌های اجتماعی
Mediation	وساطت
Ebola virus	ویروس ابولا
Attributes or Identities of Nodes	ویژگی یا هویت گره‌ها
Social Characteristics	ویژگیهای اجتماعی
Nontrivial Topological Features	ویژگیهای توپولوژیکی غیرمعمول
Emergent Properties	ویژگیهای نوپدید، ویژگیهای نوظهوری
Wikipedia	ویکی‌پدیا

ه

Correlated	همبستگی
Node’s Coreness	هسته‌ای بودن یک گره
Multiple Cores	هسته‌های چندگانه
Network Core	هستهٔ شبکه
Stochastic Equivalence	هم ارزی تصادفی
Structural Equivalence	هم ارزی ساختاری
Regular Equivalence	هم ارزی منظم
Network Connection	همبندی شبکه، اتصال شبکه
Overlapping Social Circles	همپوشانی حلقه‌های اجتماعی
Agglomerative	همجوشی
Direct Neighbors	همسایگان مستقیم
Co-Membership	هم–عضوی، عضو مشترک
Homogeneous	همگن
Gene Co-Expression	هم–گونی ژن
Homophily	همگونی، همریختی، یک ریختی

همه‌گیر شدن بیماری مرگ سیاه	Black Death Epidemic
همه‌گیری، اپیدمی	Epidemics
همه‌گیری‌شناسی، اپیدمیولوژی	Epidemiology
هنجارها	Norms
هیچ	None

ی

یالهای انعکاسی	Reflexive Ties
یالهای بین اجتماعات	Edges Between Community
یالهای بین–بخشی	Inter-Partition Edges
یالهای پسروی، یالهای بازگشتی	Back Edges
یالهای پیشروی، یالهای اکتشافی	Discovery Edges
یک طرفه	One Way
یکنواخت	Uniform