بررسی اجتماعات وب و الگوریتم‌های استخراج آن‌ها

توسط

مهیار سالک

رساله ارائه شده به عنوان بخشی از ملزومات برای دریافت درجه

کارشناسی نرم‌افزار کامپیوتر

زیرنظر

دکتر محمد قدسی

مرداد ماه ۱۳۸۴

دانشکده مهندسی کامپیوتر

دانشگاه صنعتی شریف

تهران

تقدیم به

پدر و مادرم

قدردانی

ازاستادم، آقای دکتر محمد قدسی که مرا به علوم نظری کامپیوتر علاقمند و همواره مرا از راهنمایی های خود بهره‌مند ساختند، تشکر فراوان می‌کنم و شایستگی ایشان را در تمامی وجوه زندگانی می‌ستایم. به دوست و همراهم مرتضی منعمی که مرا به این مبحث معرفی و تشویق کردند امتنان می‌ورزم. از آقای دکتر David Kempe که مرا با مسائل شبکه‌های اجتماعی و سایر مسائل مرتبط آشنا کردند و مرا در انجام این پروژه راهنمایی نمودند، بسیار سپاسگزارم. بدون کمک دوستم، محمد درخشانی، نوشتار و ویرایش این پایان‌نامه به زمان و هزینه بیشتری نیاز داشت. کمکش را قدر می‌دانم. و در خاتمه به خانواده و دوستانم به‌خاطر با من بودن‌ها ممنونم.

بررسی اجتماعات وب و الگوریتم‌های استخراج آن‌ها

چکیده

منظور از اجتماعات آن دسته از موجودیت های شبکه وب است که ارتباط قوی با یکدیگر دارند. این موجودیت ها بصورت صفحات و میزبان ها خود را نشان می دهند. برای تشخیص اجتماعات وب باید روی ساختار پیوندی بین موجودیت ها در گراف القایی وب مطالعه کنیم. شناخت اجتماعات وب به ما کمک می کند که از طریق ساختار پیوندی وب اطلاعات قابل اعتمادی در مورد محتویات صفحات وب بدست آوریم.

با داشتن گراف $G(V,E)$ که V نمایشگر صفحات وب و E نمایانگرپیوندهای بین آنهاست مساله،یافتن زیرگراف‌هایی از G است که یا بصورت هاب و مرجعی باشند (دیدگاه HITS) یا تراکم پیوندی بالا داشته باشند.

در این پایان نامه خصوصیات وب و مدل کردن آن بصورت یک گراف را شرح می دهیم. به بیان روش های سنتی و قدیمی تر یافتن اجتماعات می پردازیم. روش درخت برش کمینه را بررسی می کنیم. در ادامه روش Charikar را شرح داده و آن را روی یک مجموعه داده از صفحات وب می‌آزماییم. در انتها به جمع بندی می پردازیم.

واژه‌های کلیدی: اجتماع، هاب، مرجع، شاره، برش

فهرست

۱ مقدمه ۷

۱-۱ اجتماعات داخل وب چه معنا و مفهومی دارند؟ ۷

۱-۱-۱ مسائل مشابه ۷

۱-۱-۲ کاربردها ۸

۱-۲ اجتماعات وب به چه شکلی ظاهر می شوند؟ ۹

۱-۲-۱ نگاه هاب و مرجع ۹

۱-۲-۲ دیدن اجتماعات بصورت یک بخش متراکم یالی در گراف وب . . ۱۱

۲ ساختار گرافی وب در مقیاس صفحات و میزبان ها ۱۲

۲-۱ مقدمه ۱۲

۲-۲ جستجوی عنوان ۱۳

۲-۲-۱ پارادایم حذف و تولید ۱۴

۲-۳ چند اندازه گیری ۱۵
۲-۳-۱ توزیع درجه ۱۵

۲-۴ اندازه قطر وب ۱۸

۲-۵ تحلیل صفحات میزبان ۱۹

۳ الگوریتم HITS و شاره در استخراج اجتماعات وب ۲۱

۳-۱ الگوریتم HITS و هسته های دوبخشی ۲۱
۳-۱-۱ شباهت HITS و Pagerank ۲۴
۳-۱-۲ سازگارشدن الگوریتم PageRank برای تشخیص اجتماعات ۲۴

۳-۲ روش شاره بیشینه در بدست آوردن اجتماعات وب ۲۵
۳-۲-۱ مساله شاره بیشینه ۲۵
۳-۲-۲ تعریف اجتماعات از دیدگاه شاره ای و الگوریتم تشخیص آن . . . ۲۷
۳-۲-۳ تحلیل ۲۹

۳-۳ بحث و مقایسه ۳۰

۴ الگوریتم‌های دسته‌بندی با درخت های برش کمینه ۳۲

۴-۱ درخت های برش کمینه ۳۲
۴-۱-۱ الگوریتم ساختن درخت ۳۳

۴-۲ الگوریتم‌های دسته‌بندی و استخراج اجتماعات ۳۷

۴-۲-۱ الگوریتم ۳۷
۴-۲-۲ دسته‌بندی سلسله‌مراتبی ۳۹
۴-۲-۳ نرمالسازی یالی ۳۹
۴-۲-۴ کاربرد در WWW ۴۰
۴-۲-۵ الگوریتم دسته بندی اجتماعات ۴۱
۴-۲-۶ اجتماع ازوتریک ۴۱
۴-۲-۷ تعمیم تعریف اجتماع Esoteric ۴۲

۴-۳ جستجو برای اجتماعات ۴۳
۴-۳-۱ تسهیل شرایط ۴۵

۵ الگوریتم های تقریبی برای پیداکردن اجتماعات ۴۷

۵-۱ گراف بدون جهت ۴۷
۵-۱-۱ محاسبه $f(G)$ ۴۸

۵-۲ الگوریتم تقریبی با ضریب ۲ برای محاسبه $f(G)$ ۵۰

۵-۳ پیاده سازی الگوریتم ۵۲

۵-۴ انتخاب $d^{\max}$ بعنوان حد بالایی ۵۴

۵-۵ مساله‌ی معادل برای گراف جهت دار ۵۴

۵-۶ الگوریتم حریصانه برای محاسبه $d(G)$ ۵۶

۶ نتایج تجربی روی الگوریتم تقریبی Charikar ۵۷

۶-۱ الگوریتم تقریبی استخراج اجتماعات وب ۵۸
۶-۱-۱ اگر کل گراف درحافظه اصلی برنامه گنجانیده نشود ۵۸
۶-۱-۲ اگر بخواهیم همه اجتماعات را شناسایی کنیم ۵۸

۶-۲ داده های واقعی ۵۹

۶-۳ تحلیل نتایج ۶۱

۷ نتیجه گیری ۶۳

۸ واژه‌نامه ۶۵

فهرست شکل‌ها

۱-۱ هاب و مرجع ۱۰

۱-۲ توزیع درجات ورودی ۱۵

۲-۲ توزیع درجات خروجی ۱۶

۲-۳ نمایی از وب ۱۷

۱-۳ $K_{۲۴}$گراف دوبخشی ۲۲

۲-۳ یک شبکه شاره ۲۶

۳-۳ چند گراف مشابه ۳۱

۱-۴ الگوریتم ساختن درخت برش کمینه ۳۴

۴-۲ تشخیص کانون در درخت ۳۸

۴-۳ دسته بندی سلسله مراتبی ۳۹

۴-۴ دو دسته‌بندی غیر قابل کاهش متفاوت ۴۳

۵-۱ گراف ناهنجار ۵۳

۶-۱ خروجی الگوریتم تقریبی ۶۰

۶-۲ خروجی الگوریتم تقریبی در دور دوم ۶۱

فصل ۱

مقدمه

این پایان نامه‌به بررسی روش های تشخیص اجتماعات موجود در شبکه جهانی وب می پردازد. منظور از اجتماعات آن دسته از موجودیت های شبکه وب است که ارتباط قوی با یکدیگر دارند. این موجودیت ها بصورت صفحات و میزبان ها خود را نشان می دهند.

اساس وجود ارتباط بین این موجودیت ها بصورت ارجاع دادن یکی دیگری را بیان می شود. اصولا وجود پیوند بین دو صفحه نشان دهنده ارتباط معنایی بین آن دو صفحه می باشد. می توان انتظار داشت که صفحه ارجاع دهنده به یک موضوع معنایی در صفحه ارجاع شونده علاقمند است. پس تا این لحظه مشخص است که برای تشخیص اجتماعات وب باید روی ساختار پیوندی بین موجودیت ها مطالعه کنیم.

۱-۱ اجتماعات داخل وب چه معنا و مفهومی دارند؟

۱-۱-۱ مسائل مشابه

مساله تشخیص اجتماعات یک مساله جدید نیست. اصولا مطالعات مشابهی روی سایر شبکه ها انجام شده است. بعنوان مثال شبکه ارجاعات کتب از جمله موضوع مطالعات گذشتگان بوده است. یک فرق اساسی که بین این شبکه با شبکه وب می باشد ماهیت زنده و متغیر سایت های وب در

مقابل موجودیت ایستا و همیشگی کتاب است. بدین لحاظ است که مساله زمان در مورد سایت های وب محلیت ندارد. چه بسا صفحات پیشکسوت و قدیمی تری در وب یافت می شوند که به یک تازه وارد رجوع می دهند.

شبکه دومی که شاید بیشتر شبیه مساله ما باشد شبکه های اجتماعی [۱] می‌باشند. انواع مختلفی از اینگونه شبکه‌ها که هر کدام نوع خاصی (اجتماعی، اقتصادی و سیاسی ...) از ارتباط بین اشخاص و موجودیت‌های انسانی را بیان می‌کنند وجود دارد. اینگونه شبکه‌ها که اخیرا موضوع بسیار جذابی برای تحقیقات شده‌اند برای تحلیل و ارائه‌ی راه‌حل های انسانی مورد استفاده قرار می گیرند. بعنوان مثال در تحلیل نژادها و فرهنگ‌های قبایل بومی افریقا پیداکردن موجودیت‌هایی که به عنوان پل در بین دو قوم با سیستم‌های بسته ایفای نقش می کنند مساله جالبی است. البته مطالعه‌ی اینگونه شبکه ها لازمه تحلیل های گسترده ای در زمینه های انسانی حاکم بر آنها علاوه بر مدل های ریاضی پیشنهادی برای آنهاست.

۱-۱-۲ کاربردها

به بحث وب باز گردیم. بازیابی اجتماعات وب کاربردهای فراوانی می تواند داشته باشد. بطورکلی تحلیل معنایی اطلاعات موجودیت های وب کاری سخت و زمان‌بر است و احتیاج به یک قاضی انسانی دارد. اجتماعات وب از آن جهت می توانند به با کمک کنند چرا که بهترین منبع برای شناختن یک چیز آن چیزهایی هستند که خود به او ارجاع داده‌اند یا از او ارجاع گرفته‌اند. پس شناخت اجتماعات وب به ما کمک می کند که از طریق ساختار پیوندی وب اطلاعات قابل اعتمادی در مورد محتویات صفحات وب بدست آوریم.

چنین اطلاعاتی می تواند در بهبود کارایی موتورهای جستجو بسیار موثر واقع شود. کاربردهای متفاوت دیگری نیز می توان براساس حیطه برای چنین اطلاعاتی قائل شد. بعنوان مثال بسیاری از کاربردهای سیستم‌های توزیع شده‌ی وب و عملیات I/O و ... با استفاده از شناسایی اجزاء بسیار به هم مرتبط وب یا همان اجتماعات بسیار بهبود می یابد.

---

[۱] Social networks

۱-۲ اجتماعات وب به چه شکلی ظاهر می شوند؟

تا اینجا می دانیم اگر بخواهیم بدنبال یک اجتماع در وب بگردیم کاندیداهای ما آن دسته از موجودیت ها هستند که پیوندهای بسیاری به هم دارند. همین دید ساده سرچشمه تعاریف ریاضی متفاوتی است که برای این موجودیت معنایی داده شده‌اند.

اصولا دو فاکتور برای ارائه یک مدل ریاضی برای اجتماعات وب در نظر گرفته می‌شود.

۱) تطابق با دید مبتنی بر وجود ارتباط قوی و ارجاع کننده و ارجاع شوندگی بین موجودیت ها

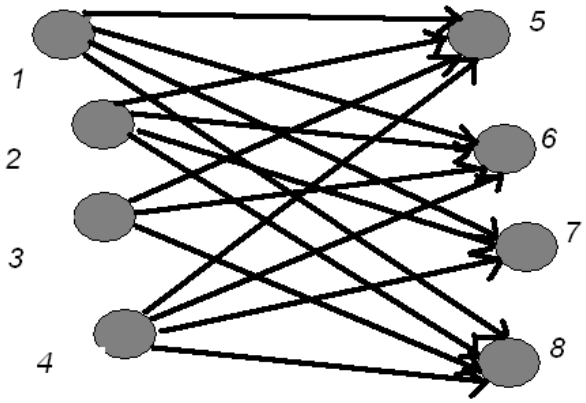
۲) سادگی برای تشخیص از طریق راه حل مدنظر

هم این پایان نامه بررسی و مقایسه راه های مختلف تشخیص این اجتماعات است. در این پایان نامه سعی شده‌است بیشتر به بررسی تفصیلی راه های جدیدتر و شاید بالقوه در تشخیص اجتماعات بپردازد. الگوریتم HITS و روش شبکه شاره Flake بطور اجمالی مورد بررسی قرار می گیرد و بیشتر به بیان راه حل درخت برش کمینه و الگوریتم تقریبی Charikar در شناسایی قسمت های شلوغ یک گراف و کاربرد آن در پیداکردن اجتماعات بپردازیم. به علاوه اینکه روش الگوریتم تقریبی را روی داده‌های حقیقی می‌سنجیم.

بطورکلی در روش هایی که برای شناسایی اجتماعات مطرح شده است همواره این واقعیت را مد نظر داشته اند که هیچ لزومی برای بیان یک تعریف دقیق از اجتماعات و دادن بها برای تخطی نکردن ازچنین تعریفی وجود ندارد. در اکثر روش‌ها یا تعریف بسیار دقیقی از این موجودیت نشده است و یا اگر شده همواره یک شعاع حول آن تعریف دقیق بعنوان مجموعه جواب برگردانیده شده است. در اینجا برداشت‌های مختلفی که از مفهوم اجتماع در هر کدام از روش‌ها شده است را می‌آوریم.

۱-۲-۱ نگاه هاب و مرجع

اگر مفهوم ارجاع دادن را در نظر بگیریم ارجاع دهنده و ارجاع شونده دو جزء اصلی می باشند. معمولا ارجاع دهنده به یک مطلب در سایت ارجاع شونده علاقمند است یا استناد می‌کند. بدین



هاب و مرجع :1-1

ترتیب مجموعه‌ای از سایت های محبوب و مرجع یا authority و مجموعه‌ای از سایت های هوادار یا هاب تشکیل می‌شود. شکل ۱-۱ به بیان این مطلب می‌پردازد.

شکل ۱-۱ ساختار یک گراف دوبخشی را یادآوری می کند. بنابراین در اجتماعات HITS مابدنبال چنین زیرگراف هایی در گراف وب می‌گردیم.

پس بصورت دقیق اگر $G(V,E)$ گراف وب با V مجموعه گره ها که نمایانگر صفحات وب و E مجموعه یال ها نمایانگر ارجاعات یا پیوندهای بین صفحات هستند یک هسته دو بخشی در وب یک زیرگراف است که شامل یک گراف دو بخشی k در l می‌شود. همانطور که گفته شد به صفحات ارجاع دهنده هاب و به صفحات ارجاع شونده در هسته دوبخشی، مرجع[۲] گفته می‌شود. اجتماعاتی که HITS پیدا می کند به اجتماعات دوبخشی[۳] مشهورهستند.

[۲] authority

[۳] Bipartitie Communities

۱-۲-۲ دیدن اجتماعات بصورت یک بخش متراکم یالی در گراف وب

اینگونه برداشت از اجتماعات که به اجتماعات متراکم یا شلوغ معروف است توسط الگوریتم های شاره و سایر الگوریتم های بررسی شده در این پایان نامه مدنظر گرفته شده است. در اینگونه اجتماعات انتظار عمومی وجود لینک های بسیار بین اعضای یک اجتماع نسبت به سایر موجودیت های وب است.

بطور دقیق تر یک اجتماع از نظر Flake یک زیرمجموعه راسی X از مجموعه V است بطوری که برای هر راس v از X تعداد یال های v به رئوس داخل X حداقل بتعداد یالهای خارج از X باشد. بیان دیگر این تعریف این است که حداقل ۵۰ درصد یال های v داخل X باشند. با این بیان دوم به راحتی می توان یک تعمیم برای این تعریف در نظر گرفت و آن پارامتر کردن عدد ثابت ۵۰ درصد است. البته Flake و بعدا Tsioutsiouliklis این مساله را از طریق شبکه شاره و با همان ثابت ۵۰ درصد حل کرده اند. پس یک پیشنهاد خوب تحقیق در مورد الگوریتم حل این مساله بصورت پارامتریزه می باشد.

دیدی که از کار Charikar از اجتماعات بدست می آوریم هم منطبق بر اجتماعات متراکم است. در حقیقت ما به دنبال آن زیر گراف از گراف وب به عنوان یک اجتماع هستیم که بیشترین نسبت یال به راس را دارد. در آینده از این نسبت به عنوان چگالی گراف G یاد می کنیم. روش دیگری نیز بعنوان PageRank از طریق استفاده از یک قدم زن تصادفی پیشنهاد شده‌است. انتظار این است که یک قدم زن تصادفی خوب بیشتر روی رئوس یک اجتماع قدم زند تا رئوس خارج؛ بخصوص هنگامی که شروع قدم زدن از یک راس داخل یک اجتماع باشد.

این پایان نامه در ادامه بدین صورت منظم شده است. در ابتدا خصوصیات وب و مدل کردن آن بصورت یک گراف را شرح می‌دهیم. در فصل ۳ به بیان روش های سنتی و قدیمی تر یافتن اجتماعات می‌پردازیم. در فصل ۴ روش درخت برش کمینه را بررسی می‌کنیم در فصل ۵ روش Charikar را شرح داده و آن را می‌آزماییم. در انتها به جمع بندی می‌پردازیم.

فصل ۲

ساختار گرافی وب در مقیاس صفحات و میزبان ها

۲-۱ مقدمه

نگاه کردن به وب به عنوان یک گراف جهت دار مارا قادر به بسیاری از تحلیل های معنایی می کند که بدون استفاده از چنین دیدی باید بهای فراوانی برای چنین اطلاعاتی بپردازیم . این نگاه می‌تواند در دو واحد صفحات و میزبان ها صورت گیرد .الگوریتم های متفاوتی مانند HITS مبتنی بر چنین دید گرافی می باشد .در این فصل به بررسی ساختار غول آسای وب می‌پردازیم و روش‌های مدل کردن وب به عنوان یک گراف را می‌بینیم .

دلایل بسیاری وجود دارد که ما علاقمندیم چنین تحلیلی را روی وب انجام دهیم از جمله دلایل اقتصادی، ریاضیاتی، اجتماعی و .. بطورکلی کارکردن با وب بعنوان یک داده ساختار پایگاه داده‌ای و نادیده گرفتن اطلاعات موثری که در پیوندهای بین صفحات وجود دارد کارعاقلانه‌ای نمی‌باشد .استفاده از ساختار گرافی وب به ما کمک می‌کند اطلاعات فراوانی در مورد محتوای صفحات و میزبان‌های وب پیدا کنیم .بسیاری از روش‌های جستجوی اجتماعات وب از چنین ساختاری بهره می‌برند . ساختار گراف وب بهره برداری به منظور بهبود جستجو در موتورهای

جستجو شده است.[۳،۱]

بطورکلی اگر موضوع صفحه v برای ما ناشناخته و یا غیرقابل دسترس باشد با استفاده از مجموعه صفحاتی که v به آنها اشاره کرده و یا مجموعه صفحاتی که به vاشاره می‌کند می‌توانیم اظهارات دقیقی در مورد v و محتوای آن بکنیم .توزیع توانی به نظر می‌رسد توزیع مناسبی برای پیوند بین صفحات باشد .جالب است که توزیع‌های مشابهی برای سایر موارد مشابه مانند مرجع دادن مقالات علمی توسط[۴] Gilbertپیدا شده است . Mendelzon در[۷] به بحث درباره این موضوع پرداختند که واسط تقاضاهای SQLبه پایگاه داده‌ها مناسب برای کشف خصوصیات جالب نهفته در ساختار گرافی وب نمی‌باشد .او و Wood زبان +Gرا پیشنهاد دادند به عنوان زبانی با قابلیت بیان بالاتر نسبت به SQL برای تقاضاهای متناسب با ساختار گرافی وب کارهای مشابهی در نگاه کردن وب بصورت یک پایگاه داده‌ی نیمه ساختار یافته انجام شده که در نهایت به برداشت مختلط از وب بصورت رابطه‌ای و گرافی منجر شد.

۲-۲ جستجوی عنوان

مساله جستجوی عنوان بشکل زیر تعریف می‌شود :اگر یک عنوان برای جستجو داده شود(که این عنوان بصورت یک Query مطرح می‌شود) بعنوان خروجی صفحاتی را برگردان که در مورد آن عنوان دارای کیفیت بالایی باشند.

چنین مساله‌ای کاربرد هر روزه‌ی موتورهای جستجوست .هر کاربری که حتی برای اندک باری با موتورهای جستجو کار کرده باشد می‌داند که این مساله در موتورهای جستجو بصورت بهینه حل نشده است .بطورکلی برای هر عنوان خاص مجموعه‌ای از صفحات مرجع اطلاعاتی (authority)و صفحات بسیاری که به آن‌ها اشاره دارند که ما از آنها به هاب نام می‌بریم وجود دارند .

الگوریتم HITS در حقیقت به جستجوی چنین ساختارهایی می‌پردازد. بطور خلاصه ما به هر صفحه دو وزن نامنفی (x_p, y_p) اختصاص می‌دهیم که x_p وزن مرجعیت و y_p وزن اشاره‌دهی است .ایده اصلی این است که اگر صفحه‌ای با مقدار بسیاری اشاره ده با وزن بالا اشاره شد ما تمایل به

افزایش اعتبار مرجعیت آن داریم و بالعکس.

$$x_p = \sum_{q|q\to p} y_p$$

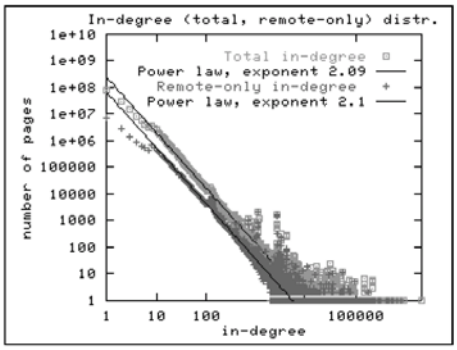
$$y_p = \sum_{q|p\to q} x_p$$

در حقیقت این درک ما را منجر به یافتن گراف‌های دارای زیر گراف‌های دوبخشی در وب می‌کند .پس الگوریتم ما بدین فرض کار می‌کند که برای یک عنوان خوش بیان در وب یک هسته در گراف وب وجود دارد .به جستجو و شمارش چنین هسته‌های دو در وب عمل trawlingگفته می‌شود .نتایج تجربی نشاندهنده این واقعیت است که چنین الگوریتم لازم است که دسترسی تصادفی به یال‌های داخل گراف داشته باشد که این مستلزم نگاه دارای کل گراف در داخل حافظه اصلی است که مانع از سرباره سنگین خواندن از حافظه جانبی برای هر بار دسترسی شود .برای پیداکردن چنین هسته‌هایی پارادایم حذف و تولید توسط[۹] پیشنهاد شده است .

۲-۲-۱ پارادایم حذف و تولید

در قسمت حذف بعنوان مثال اگر ما دنبال هسته $C_{۴,۴}$ هستیم کلیه گره‌های با درجه ورودی ۳یا کمتر نمی‌توانند در سمت راست چنین هسته‌ای باشند پس کلیه یال‌هایی که به چنین گره‌هایی ختم می‌شوند قابل حذف می‌باشند با حذف چنین یال‌هایی مکررا می‌توانیم عمل حذف را انجام دهیم .

عمل تولید نقطه مقابل حذف می‌باشد فرض کنید در پیداکردن $C_{۴,۴}$ گره u درجه ورودی ۴ را داشته باشد در این صورت این گره تنها در صورتی متعلق به یک هسته بصورت $C_{۴,۴}$ می‌باشد اگر و تنها اگر هر۴ گره که به u اشاره دارد حداقل ۴ همسایه داشته باشد پس چنین فیلتری حالت صلب و اثبات دارد .اینگونه اعمال بصورت خطی با سایز گراف انجام می‌شوند .



توزیع درجات ورودی (شکل از]۶[ ) :2-1

۲-۳ چند اندازه گیری

در این قسمت ما به بررسی اندازه گیری های انجام شده روی مشخصه‌های گراف وب می‌پردازیم .نتایج توزیع درجه در قسمت اول از کارهای Kumarو سایرین در [۸] Albert و سایرین در [۲] بدست آمده است .در نهایت تحلیل جزء همبند و تحلیل قطری از کارهای Broder بدست آمده است .

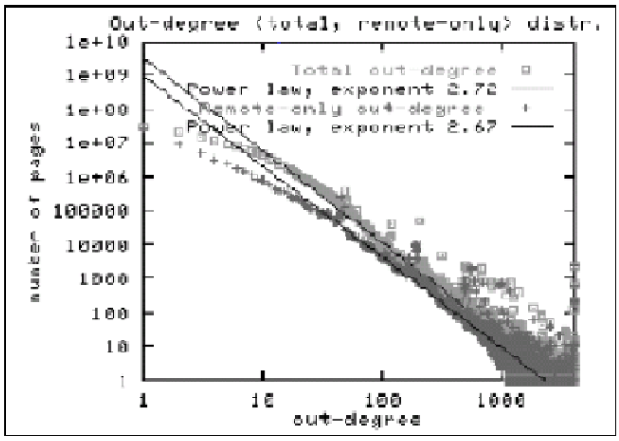
۲-۳-۱ توزیع درجه

با بررسی درجه ورودی و خروجی گره های گراف وب شروع می‌کنیم .کارهای اولیه kumar در [۸] به مشاهده وجود توزیع توانی در درجات ورودی بود :کسری از صفحات وب یا درجه ورودی i متناسب با $1/i^x$ برای عدد $x > 1$ می‌باشد .در ادامه کارهای Albertدر [۲] چنین نظری را تایید کرد و یک مقدار مناسب برای x را برابر ۲.۱ تخمین زد .

نمودار log-log (با محور منفی شده‌ی x ها) در ۲-۱از توزیع درجات ورودی می‌باشد[1].همانطور که

مشاهده می‌شود مقدار ۲.۱ به بهترین شکل به داده‌های ورودی ما می‌خورد .

[1] [۶]



توزیع درجات خروجی (شکل از]۶[) :2-2

شکل ۲-۲ نتیجه مشابه را برای درجات خروجی نشان می‌دهد [۲]. بهترین خط تطابق توزیع با $x = ۲.۷۲$را برای ما می‌دهد .میانگین درجه خروجی برابر **۷.۲** می‌باشد .

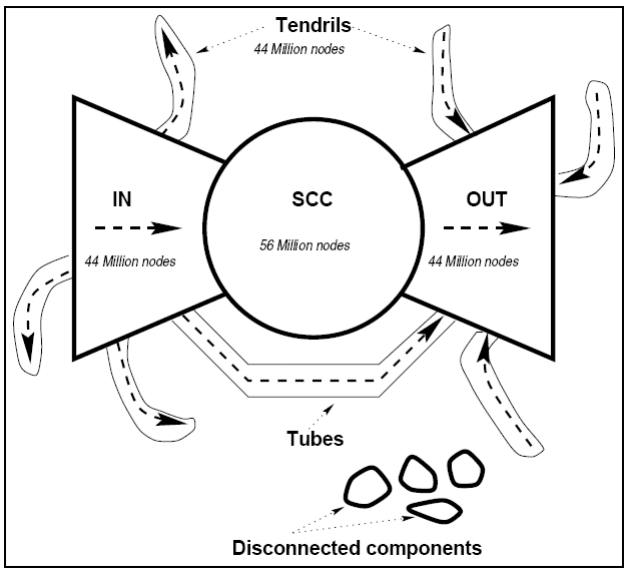
با استفاده از جستجوهای سطح اول و استفاده از گره‌های تصادفی Broder و سایرین توانستند که ساختار شکل ۲-۳ را برای گراف وب پیشنهاد کنند. [۳]

این نتیجه از پیمایش Altavisto درماه مه 1999 بدست آمده است که توسط پیمایش مشابه در همان سال جایگزین شده است .برای فهم این شکل ابتدا به بیان چند تعریف می‌پردازیم .

تعریف ۱. یک جزء همبند ضعیف مجموعه‌ای از صفحات است که به هم دسترسی دارند اگر پیوندها بدون توجه به جهت شان دنبال شوند.

وسیع ترین جزء همبند ضعیف در این پیمایش برابر۱۸۶ میلیون گره بوده است که بیش از۹۰ درصد از این پیمایش را شامل می‌شود .

[۲] اصل شکل از [۶]
[۳] اصل شکل از [۶]



نمایی از وب :2-3

تعریف ۲. بطور مشابه یک جزء قویا همبند یک مجموعه از صفحات است که برای دو جفت صفحه (u, v) در این مجموعه یک مسیر جهت دار از u به v موجود است .

به زبان وب این بدان معناست که یک شخص از u می‌تواند توسط Hyper Link ها به v برسد .وسیع ترین جزء قویا همبند حدود ۵۶ میلیون گره دارد .

Broder و Kumar[۶] شکل پیشنهاد شده را Bowtie می‌نامد و ناحیه‌های مختلفی را در داخل Bowtie شناسایی می‌کند .هسته Bowtie که SCC نامیده است نمایشگر جزءقویا همبند غول آسای وب شامل ۵۶ میلیون گره است .

سمت چپ bowtie نمایشگر ۴۴ میلیون صفحه به نام IN است که نشاندهنده صفحاتی خارج از SCC است که به SCC راه دارند .پس یک کاوشگر وب با شروع از یک گره در IN می‌تواند به کلیه گره‌های در SCC دسترسی داشته باشد. IN بطور شهودی آن دسته از صفحات جدید است که خود به مقاصد جالب پیوند دارند ولی هنوز توسط هسته وب کشف نشده‌اند .

بطور مشابه مجموعه دیگر ۴۴ میلیونی از صفحات که سمت راست bowtie را نشان می‌دهند و OUT خوانده می‌شود که خصوصیت آن این است که هر صفحه‌ای از OUT توسط هر صفحه‌ای

در SCC قابل دسترسی است . بطور شهودی چنین صفحاتی صفحات مشهور و پیشکسوت اینترنت هستند که فقط به خودشان دسترسی دارند .

پس مدل ما توانایی کاوش bowtie را از چپ براست نشان می‌دهد ولی عکس آن درست نیست.

نهایتا ناحیه چهارم که Tendrils خوانده می‌شود از صفحاتی تشکیل شده که به هسته راه ندارند و از هسته هم به آنها راهی نیست . این صفحات در حقیقت اجتماع عیوب مجموعه IN و OUT هستند .باز هم حدود ۴۴ میلیون از این گونه صفحات کشف شده است .

۲-۴ اندازه قطر وب

شکل Bowtie فاش کننده یک حقیقت غیرمنتظره در مورد همبندی وب است .برای اکثر صفحات v و u مسیری از u به v وجود ندارد .بطور دقیق تر اگر u در SCCاجتماع با INو v در SCCاجتماع با OUTباشد در اینصورت مسیری بین u و vموجود است .احتمال قرارگرفتن u در $\bigcup$ IN SCCحدود $\frac{1}{4}$ و احتمال قرار گرفتن v در SCC $\bigcup$ OUT هم برابر $\frac{1}{4}$ است پس احتمال این دو پیشامد مستقل حدود $\frac{1}{4}$ است.

پس در حدود ۷۵ درصداز صفحات وب به هم متصل نیستند .

آخرین تحلیل‌های Albert در[۲] قطر 19 را برای وب پیش بینی کرده بود در حالی که اندازه گیری های واقعی تصویر متفاوتی به ما نشان می‌دهد اکثر جفت صفحات وب فاصله بی نهایت از هم دارند . به عنوان یک گزینه ما می‌توانیم میانگین فاصله اتصالی را تعریف کنیم بشرطی که بین دو صفحه مسیری واقع باشد .چنین میانگین ۲-۱ در جدول آمده است .

نوع پیوند	ورودی	خروجی	بدون جهت
میانگین فاصله اتصالی	16.12	16.18	6.83

میانگین فاصله اتصالی با توجه به نوع پیوند :2-1

۲-۵ تحلیل صفحات میزبان

تا این جا ساختار گرافی وب را با واحدهای صفحات وب بررسی کردیم .در عمل وب یک ساختار سلسله مراتبی است با دامنه، صفحات مزبان و سایت‌ها که نمایانگر در جات مختلف کنترل و مدیریت می‌باشند .برای درک بهتر ساختار وب ما باید کلان ساختار وب را در مقیاس پیوندهای بین وب سایت ها بررسی کنیم .

تعریف ۳. وب سایت :یک مجموعه از صفحات وب است که نمایانگر یک موجودیت در وب می‌باشد .

تعریف ۴. دامنه :متشکل از تعدادی میزبان وب است که جزءآخر مشترک در اسم خود دارند(بطور مثال comیا ir)

دامنه‌ها بیشتر وابسته به کشورها هستند با این وجود دامنه‌هایی هست که مرز جغرافیایی نمی‌شناسند .جدول ۲-۲ آماره در مورد سایز صفحات وب است .

	1999 اکتبر	2000 آگوست	2001 ژوئن
صفحات وب به میلیون	128.99	604.37	1,292
پیوندها به بیلیون	1.27	5.54	19.94

سایز وب :2-2

	1999 اکتبر	2000 آگوست
بدون وزن	4.11	5.27
وزن دار	3.31	3.71

تخمین میانگین فاصله بین میزبان‌ها :2-3

با توجه به تعریف ما از میانگین فاصله اتصالی در بخش قبل جدول ۲-۳ فاصله میانگین در جزء SCC نشان می دهد .

البته برای میزبان ها با توجه به تحلیل Henzinger در [۵] توزیع درجات در بین میزبان‌ها نیز از توزیع توانی یا Zipfian پیروی می‌کند که i مناسب برای این توزیع ۱.۶۲ برای درجات ورودی و ۱.۶۷برای درجات خروجی است .

در نهایت مدلی که برای شبیه سازی گراف وب چه در مقیاس صفحه و چه در مقیاس میزبان‌ها پیشنهاد شده است به مدل نسخه برداری یا Copy Model مشهور است .

بطور خلاصه برای ساختن چنین گرافی در هر گام با احتمال β ما یک گره از قبل موجود را انتخاب می‌کنیم و لینک‌های جدیدی را به آن طبق قانون زیر اضافه می‌کنیم :

- ابتدا یک گره تصادفی v از میان گره‌ها موجود و d تا لینک تصادفی آن را انتخاب می‌کنیم .
- در ادامه برای d تا از $i = ۱, \ldots, d$ امین لینک جدید u به یک گره تصادفی موجود با احتمال αاشاره می‌کند یا به مقصد iامین لینگ vبا احتمال $۱ - \alpha$

در نهایت با احتمال$۱ - \beta$ یک گره جدید به گراف اضافه کرده و لینک‌های خروجی را بدان همانند مدل نسخه برداری اضافه می‌کنیم .

این مدل نشاندهنده دو تغییر با گذشت زمان در وب است .

۱) مروز زمان گره‌های جدیدی را به گراف اضافه می‌کند یعنی صفحات یا میزبان های جدید.

۲) صفحات یا میزبان‌های کنونی می‌توانند پیوندهای جدیدی به گره‌های دیگر ایجاد کنند .

فصل ۳

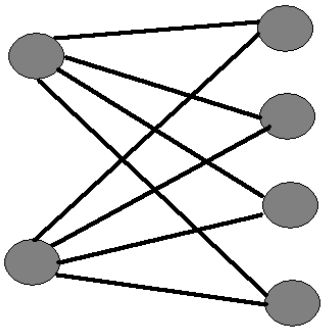
الگوریتم HITS و شاره در استخراج اجتماعات وب

۳-۱ الگوریتم HITS و هسته های دوبخشی

الگوریتم HITS بعنوان یک روش محلی در پیداکردن اجتماعات وب نگاه می شود. یک روش کاملا محلی آن روشی است که تنها خصوصیات یک همسایگی محلی حول و حوش دو گره را بررسی می کند تا تشخیص دهد که آیا این دو گره عضو یک اجتماع هستند. در مقابل روش های جهانی کلیه یال ها در گراف وب مورد بررسی قرار می دهد برای اینکه تصمیم بگیرد که آیا دو گره در داخل یک اجتماع قرار می گیرند یا خیر.

برای بررسی میزان شباهت یا ارتباط دو گره روش های سنتی و متریک های سنتی وجود دارد. اگر گراف وب را همراه با ماتریس مجاورت A در نظر بگیریم حاصلضرب $(AA^T)_{uv}$ و $(A^TA)_{uv}$نشان دهنده میزان رجوع شوندگی و رجوع کنندگی برای جفت صفحه u و v می باشد.

البته اینگونه متریک ها علی رغم ظاهر تمیز و ساده‌ی خود مشکلات فراوانی در عمل خواهند داشت. اولا هر دو ماتریس AA^T و A^TA بسیار متراکم تر از خود A می باشد و این مشکلات ذخیره سازی و خواندن از حافظه را بدنبال می آورد. ثانیا بعضی از صفحات ممکن است



نمای تصویری گراف دوبخشی $K_{۲۴}$:3-1

به (از) هزاران صفحه رجوع کنند(شوند). البته مشکل دوم با استفاده از نرمالسازی قابل حل می شود. پس تا این لحظه یک میزان برای تعیین شباهت میان دو صفحه با وجود تمامی مشکلات آن در پیاده سازی پیدا کرده ایم. ولی آنچه ما بدنبال آن هستیم حسی است که در دید کلی از اجتماعات وب بدست می آوریم.

تعریف ۵. یک گراف کامل دوبخشی یک گراف جهت دار است بصورتی که گره های آن قابل تقسیم به دو بخش L و R می باشند که $L \bigcup R = V$ و $L \bigcap R = \emptyset$ بدین صورت که‌ازهر گره در L یالی به هر گره در R وجود دارد. ما چنین گرافی را به K_{lr} نشان می دهیم که $l = |L|$ و $r = |R|$.

یک هسته دو بخشی یک زیرگراف از یک گراف است که اگر به تنهایی نگاه شود یک گراف کامل دوبخشی است. شکل ۳-۱ مثالی از $K_{۲۴}$ می باشد.

الگوریتم HITS از Kleinberg در[۱۰] یک زیرگراف از گراف وب را می گیرد و دو وزن برای هرصفحه دراین زیر گراف تولید می کند. این اوزان اغلب بعنوان امتیاز هاب و مرجعیت نامیده می شوند و ذاتا مربوط به خصوصیات ماتریسی آن بخش از گراف وب هستند که الگوریتم روی آن اجرا شده است.

بطور معنایی یک هاب یک صفحه از وب است که به تعداد زیادی از مراجع ارجاع می دهد و یک مرجع یک صفحه وب است که توسط تعداد زیادی از هاب ها ارجاع شده است. (به ماهیت بازگشتی این تعریف توجه کنید). دو امتیازی که HITS برای هر صفحه نشان می دهد نمایانگر این حقیقت است که تا چه حدی دو صفحه این دو خصوصیت را از خود نشان می دهد. اگر به شکل

۳-۱ مراجعه کنیم گره های سمت راست نمایانگر مراجع و سمت چپی ها نمایانگر هاب ها می باشند.

الگوریتم HITS در دو گام انجام می شود. مرحله اول پیش پردازشی است که معین می کند کدام زیرگراف از گراف غول آسای وب گزینش شود مادامیکه گام دوم یک پروسه عددی تکرار شونده است. راه معمول برای گزینش مجموعه دانه بصورت زیر است.

۱) یک پرسش مورد علاقه را به یک موتور جستجو دلخواه بفرست.

۲) مجموعه m صفحه اول جواب را انتخاب کن.

۳) مجموعه را با کلیه همسایگان بلاواسطه یا با یک واسطه گسترش بده.

در انتخاب مجموعه دانه هیورستیک های متفاوتی می تواند مورد استفاده قرار بگیرد. بعنوان مثال اینکه صفحات داخلی یک میزبان مورد استفاده قرار نگیرد یا اینکه یک حد بالایی برای تعداد صفحات گزینش شده بگذاریم و ...

با داشتن چنین مجموعه دانه ای اگر $G(V,E)$ این زیرگراف از گراف وب باشد و A ماتریس مجاورت G قسمت دوم تکرار شونده الگوریتم HITS دو بردار $|V| \times ۱$بعد ی به نام های $\mathbf{a},\mathbf{h}$ را بصورت زیر بروز می کند. مقدار اولیه این دو بردار می تواند ۱ یا هر عدد ثابت دیگری باشد.

$$a = A^T h \tag{۱}$$

$$h = Aa \tag{۲}$$

$$a = \frac{a}{||a||^۲}$$

$$h = \frac{h}{||h||^۲}$$

معادله۱ بیانگر این مطلب است که امتیاز مرجع بودن یک صفحه جمع امتیازات هاب بودن صفحاتی است که او ارجاع داده اند. و مشابها معادله ۲ نشان دهنده این است که امتیاز هاب بودن یک صفحه جمع امتیاز مراجعی است که او به آنها ارجاع داده است.

۳-۱-۱ شباهت HITS و Pagerank

الگوریتم HITS از جهت تحلیل ماتریسی و نتایج بسیار شبیه الگوریتم Page Rank می باشد هر چند روش Page Rank در نگاه اول خیلی شبیه HITS نیست. به اختصاریک قدم زدن تصادفی در الگوریتم PageRank موجود است که در هر لحظه یکی از این دو عمل را انجام می دهد.

- با احتمال ε می تواند به یک صفحه در وب نقل مکان کند.
- با احتمال $۱ - \varepsilon$ از روی یک یال تصادفی متصل به مکان فعلی اش به یک همسایه دیگر می رود.

در نهایت هر صفحه ای یک احتمال ثابت و قابل محاسبه ملاقات شدن توسط قدم زن تصادفی دارد که همان امتیازش است. بطور صریح محاسبه این امتیاز از طریق رابطه تکرار شونده زیر است.

$$r_v{}^t + ۱ = \frac{\varepsilon}{n} + (۱ - \varepsilon) \sum_{u:(u,v)\in E} \frac{r_u{}^t}{d_u{}^{out}} \qquad (۳)$$

kleinberg در [۱۰]با استفاده از الگوریتم خود توانست صفحاتی را که از نظرواژه‌ی کلیدی مشابه ولی معنای متفاوت بودند به رده های مناسب دسته بندی کند. بعنوان مثال صفحه حاوی لغت Jaguar به رده های مختلف مثل رده اتومبیل، حیوان، تیم فوتبال، بازی کامپیوتری، سیستم عامل و ... دسته بندی شد) با این مفهوم الگوریتم HITS قابل تبدیل به یک تشخیص دهنده اجتماع در وب است.

۳-۱-۲ سازگارشدن الگوریتم PageRank برای تشخیص اجتماعات

الگوریتم PageRank را نیز می توان براحتی برای تشخیص اجتماعات داخل وب سازگار کرد. معادله ی یکنواخت ۳ می تواند به یک معادله حساس به موضوع در روش PageRank تبدیل شود.

$$r_v{}^t + ۱ = \varepsilon p_v + (۱ - \varepsilon) \sum_{u:(u,v)\in E} \frac{r_u{}^t}{d_u{}^{out}} \quad (۴)$$

با این شرط که $\sum_v p_v = ۱$. مقدار p_v برای تمامی vهاطوری انتخاب می شود که نشان دهنده احتمال پاگذاشتن قدم زن تصادفی به صفحهv با توجه به موضوع باشد. این پروسه می تواند در تشخیص اجتماعات بکار آید بدین ترتیب که برای دو صفحه یک امتیاز حساس به موضوع را محاسبه می کند. در نهایت صفحاتی که به اجتماع متعلق می باشند احتمال ملاقات شدنشان از یک حد بالاتر است و سایر صفحات خارج از اجتماع می باشند.

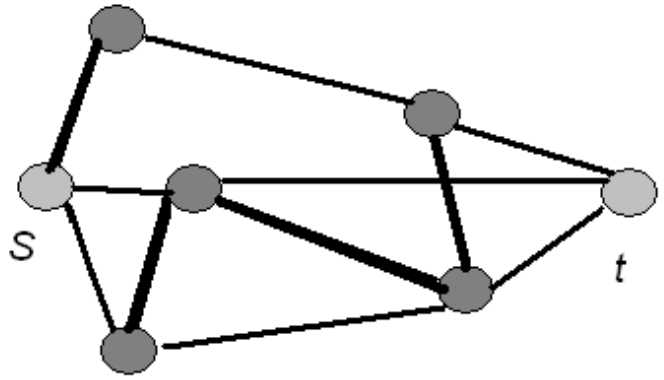
۳-۲ روش شاره بیشینه در بدست آوردن اجتماعات وب

روش شاره بیشینه از جمله روش هایی در بدست آوردن اجتماعات بحساب می رود که خصوصیات هر دو نوع روش های محلی و جهانی را دارد. این روش می تواند هم روی کل وب هم روی یک زیرگراف پیاده سازی شود. پیچیدگی این روش در بدترین حالت تابعی از سایز کل وب است با این وجود در عمل این پیچیدگی محدود به اندازه اجتماعی می شود که پیدا کرده است. همچنین اجتماعی که از این روش بدست می آید مطمئن تر و دقیق تر از سایر روش هاست.

همانطور که انتظار می رود الگوریتم پیداکردن اجتماع مساله را تبدیل به یک مساله پیداکردن شاره بیشینه یا معادل آن برش کمینه می کند. در اینجا توضیح متخصری در مورد مساله شاره بیشینه و الگوریتم پیداکردن آن می دهیم.

۳-۲-۱ مساله شاره بیشینه

یک شبکه آب رسانی را در نظر بگیرید. این شبکه از یک سری لوله های آب با قطرهای مختلف، یک سری گذرگاه و یک منبع و مقصد تشکیل شده است. قطرهای مختلف این لوله ها هر کدام ظرفیت شخصی برای گذردهی آب تعیین می کنند. شکل ۳-۲ نمونه ای از این شبکه را نشان می دهد.



یک شبکه شاره :3-2

ما علاقمندیم که بدانیم این شبکه بیشترین مقدار آبی که از خود عبور می دهد چه اندازه است . بنابراین گراف $G(V,E)$ اگر معادل چنین شبکه ای باشد و برای کلیه یال های $(u,v) \in E$ $c(u,v)$ نشان دهنده ظرفیت آن یال باشد و قرارداد می کنیم که $c(u,v) = 0$ در صورتی که بین u,v یالی موجود نباشد.

بنابراین با گرفتن دو راس s,t یک مساله شاره بیشینه $s-t$ پیداکردن بیشترین مقدار شاره ایست که از راس s به راس t می تواند جریان داشته باشد بشرطی که هیچ یالی بیشتر از ظرفیت خود شاره تحمل نکند و هیچ شاره ای هم در بین راه نابود نشود. طبیعی است که آنچه این مقدار را تعیین می کند آن قسمت هایی از شبکه هستند که حکم گردی بطری را دارند. Fulkerman, Ford در تئوری« شاره بیشینه برش کمینه» خود ثابت کردند که بیشترین شاره ممکن در یک گراف با منبع s و مقصد t برابر با برش با کمترین ظرفیتی است که s را از t جدا می کند. الگوریتم چند جمله ای فراوانی برای حل مساله شاره بیشینه پیشنهاد شده است که بعنوان مثال می توان :به[۱۱] اشاره کرد. درزیر یکی از ساده ترین این الگوریتم ها آمده است.

```
Max-Flow (graph G(V,E); vertex: s,t)
  set R = residual network of G
```

```
  while R contains a directed path from s to t
    identify shortest augmenting path, P, from s to t
    set gamma = min (r(u, v) : (u, v) is in P)
    for all (u, v) in P
      set r(u,v) = r(u,v) - gamma
      set r(v,u) = r(v,u) + gamma
return R
```

این الگوریتم روی گراف باقیمانده ها کار می کند. این گراف اطلاعات یال ها را در مورد ظرفیت مصرفی و ظرفیت باقیمانده یا پتانسیل نگاه می دارد. در نهایت در هر دور الگوریتم یک مسیر روی گراف باقیمانده ها پیدا می کند که بتوان با ظرفیت های غیرصفر از s به t رسید. کمترین ظرفیت این مسیر به شاره کنونی اضافه می شود. اگر چنین مسیری وجود نداشت کار الگوریتم به پایان رسیده است.

۳-۲-۲ تعریف اجتماعات از دیدگاه شاره ای و الگوریتم تشخیص آن

تعریف ۶. یک اجتماع یک زیرگراف بارئوس $X \subseteq V$ می باشد به نحوی که برای هر گره $v \in X$ تعداد یال هایی که v را به گره هال داخل X متصل می کند کمتر از تعداد یال هایی که v را به خارج وصل می کند نباشد.

توجه کنید که چنین تعریفی قدری بازگشتی است چون دارای غالبی مشابه غالب زیر است که یک ایرانی بیشتر ایرانیان را می شناسد تا غیر ایرانیان را. البته تشخیص فراگیر و دقیق اجتماعات با این تعریف یک مساله NPتمام‌است با این حال روش شاره بیشینه اکثر اجتماعات قوی را باز می گرداند.

ورودی الگوریتم تشخیص اجتماع گراف G و یک مجموعه دانه S و یک پارامتر α می باشد. رویه، یک گراف از روی G به انضمام دو گره منبع و مقصد مصنوعی می سازد از منبع با ظرفیت نامحدود به کلیه رئوس مجموعه دانه و از مقصد با ظرفیت تعیین شده توسط α به کلیه رئوس کل

گراف متصل می کنیم. پارامتر α در حقیقت یک پارامتر کوک کردن می باشد که کار آن تنظیم کردن سایز اجتماعات استخراجی می شود. بعد از ساخت رویه ما الگوریتم شاره بیشینه را به عنوان یک زیر برنامه فراخوانی می کند و در انتها آن بخش از گراف باقیمانده ها که در مولفه همبندی S قرار می گیرد به عنوان یک اجتماع منطبق با تعریف ؟ برگردانیده خواهد شد.

در زیر شبه‌کد الگوریتم فوق را آمده است. دو الگوریتم مرتبط به نام های Approximation-com و com-cluster داریم که هر دو آنها الگوریتم com را به عنوان یک زیربرنامه فراخوانی می کنند. الگوریتم com هنگامی است که کل گراف در حافظه اصلی گنجانیده شود.

```
com ( graph G(V,E), S, alpha)
  set V_alpha = V union {s, t}
  set E_alpha = E union {(v, t): v in V} union {(s, u) : u in S}
  set c(v, t) = alpha, for all v in V
  set c(s, u) = not_bounded for all u in S
  let R be the output of Max-Flow(G_alpha, s, t)
  set X be the member of smallest connected component about s in R
  return X - {s}
```

همانطور که می دانیم گراف غول آسای وب دارای چنین خصوصیتی نمی باشد. پروسه Approximation-com هنگامی به کار می رود که تنها بخش کوچکی از گراف بتواند در حافظه نگاه دار می شود. این الگوریتم از یک خزش[1] اندازه ثابت برای یافتن یک اجتماع تقریبی استفاده می کند و قوی ترین اعضای این اجتماع را به عنوان دانه های مرحله بعدی در نظر می گیرد.

```
Approximation-com (S, k)
  while number of iteration is less than an optional number
    set G to a crawl from S of depth k
```

[1] crawl

```
    let alpha be the size of S
    set X = com(G, S, alpha)
    rank all v in V by number of edges in X
    add highest ranked non-seed vertices in X to S
  return X
```

رویه com-cluster برای پیداکردن کلیه اجتماعات در یک گراف بکار می رود. این الگوریتم هم فرض می کند که کلیه گراف در داخل حافظه اصلی جا می گیرد.

```
com-cluster(graph G(V,E), alpha)
  set S = V
  while exists s in S
    set X = com(G, {s}, alpha}
    for all v in V
       set cluster(v) = s
    set S = S - X
  return cluster labels of v in V
```

۳-۲-۳ تحلیل

بخاطر آورید که ما از پارامتر به عنوان پارامتر کوک کردن نام بردیم. قضیه زیر در مورد پارامتر صادق است.

قضیه ۷. اگر X اجتماعی باشد که توسط الگوریتم com با پارامتر α پیدا شده است برای هر P و Q به نحوی که P و Q، X را افراز کنند حدود زیر برقرار است.

$$\frac{f(x, V-X}{|V-X|} \leq \alpha \leq \frac{f(P,Q)}{\min(|P|,|Q|)}$$

اثبات قضیه فوق در[۱۴] یافت می شود. در حقیقت α یک حد بالایی برای ظرفیت های برون اجتماعی و یک حد پایینی برای ظرفیت های درون اجتماعی است. بنابراین الگوریتم com تضمین می کند که اعضای اجتماع بطور متراکم به هم مرتبط و یال های کمی به خارج داشته باشند. در دو حالت اکسترمم انتخاب یک مقدار کوچک برای α مثلا نزدیک به صفر منتهی به پیداکردن کل گراف به عنوان یک اجتماع می شود در صورتی که انتخاب یک مقدار بزرگ برای α منجر برگرداندن n اجتماع تک عضوی[۲] خواهد شد که در اینجا منظور از یک مقدار بزرگ می تواند عددی بزرگتر از مجموع ظرفیت های گراف باشد. واقعیت دیگر در مورد الگوریتم Com-Cluster است. برای دو اجتماع تشخیص داده شده X_u و X_v در این الگوریتم یکی از سه حالت زیر برقرار است.

$X_u \bigcap X_v = \emptyset$ یا $X_u \subseteq X_v$ یا $X_v \subseteq X_u$

۳-۳ بحث و مقایسه

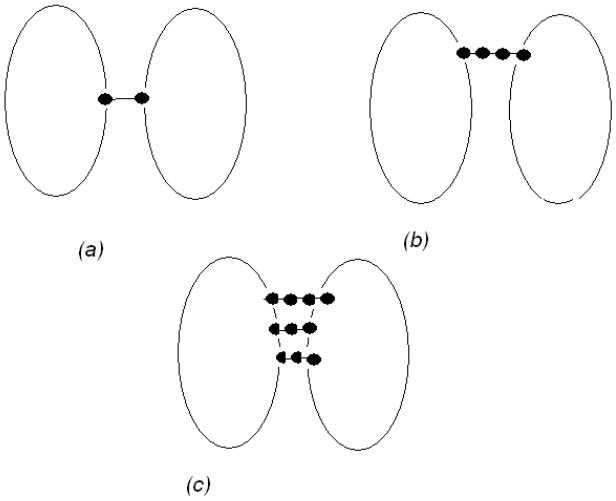
در الگوریتم درخت های برش کمینه در فصل آتی مفهوم کانون مطرح می شود. مقصد مصنوعی که ما در این الگوریتم برای گراف انتخاب کردیم در رابطه تنگاتنگی با کانون درخت برش کمینه قرار می گیرد. به فصل مربوطه مراجعه کنید.

در مقایسه با الگوریتم HITS باید گفت یک مزیت الگوریتم های مبتنی بر شاره نسبت به الگوریتم HITS و مشابها Pagerank عدم حساسیت الگوریتم شاره به فواصل می باشد. در شکل ۳-۳ این تفاوت به تصویر کشیده شده است. در حالی که اشکال a و b از دید الگوریتم شاره یکسان است. ولی یک قدم زن تصادفی بسیار بین این دو شکل تمییز قائل می شود چون در شکل b مسیر طولانی باید برای رسیدن به یک جزء از جزء دیگر طی شود در حالی که شکل a چنین خصوصیتی ندارد.

در مقابل شکل c با وجودی که از نظر یک قدم زن تصادفی با شکل a توفیری ندارد ولی الگوریتم شاره این دو شکل را بسیار متفاوت می یابد زیرا شاره های پیشینه در این شکل با هم تفاوت می کنند.

---

[۲] singleton



چند گراف مشابه با خصوصیات برشی و ماتریسی گوناگون :3-3

فصل ۴

الگوریتم‌های دسته‌بندی با درخت های برش کمینه

۴-۱ درخت های برش کمینه

در این بخش نگاه اجمالی به تئوری درخت های کمینه می کنیم. حتی الامکان از پرداختن به اثبات ها در بیان تئوری ها خودداری می کنیم. این بخش زمینه را برای معرفی روش Tsioutsiouliklis در دسته‌بندی فراهم می کند.

بیان درخت برش کمینه در وهله اول توسط Hu, Gomory در[۱۲] مطرح شد. از این بابت این داده ساختار بر درخت Gomory-Hu نیز خوانده می شود. فرض کنید $G(V,E)$ گراف بدون جهت و $V=\{v_1,\dots,v_n\}$ مجموعه گره ها و $f(i,j)$ یا f_{ij} شاره بیشینه بین دو گره $v_i,v_j\in V$ باشد.

قضیه ۸. مجموعه اعداد نامنفی $f_{ij}=f_{ji}$ که در آن $i,j\in\{1,\dots,n\}$ یک بیشینه شاره معتبر است اگر و فقط اگر

$$f_{ik}\geq\min\{f_{ij},f_{jk}\}$$

برای تمام $i, j, k \in \{1, \ldots, n\}$

نتیجه : با استفاده از استقرا می توان گفت برای هر دنباله $v_1, \ldots, v_n$ از گره ها

$$f_{1n} \geq \min\{f_{12}, f_{23}, \ldots, f_{(n-1)n}\}$$

علاوه بر این برای هر شبکه بدون جهت G یک درخت T وجود دارد که حاوی شاره های بیشینه آن شبکه است.

قضیه ۹. اندازه‌ی شاره بیشینه بین دو گره v_i و v_j از گراف G برابر با کمترین مقدار مجموعه $\{f_{ia}, f_{ab}, \ldots, f_{dj}\}$ است که $v_{ia}, v_{ab}, \ldots, v_{dj}$ وزن یال هایی هستند که مسیر یگانه بین دو گره v_i و v_j را در T تشکیل می دهند.

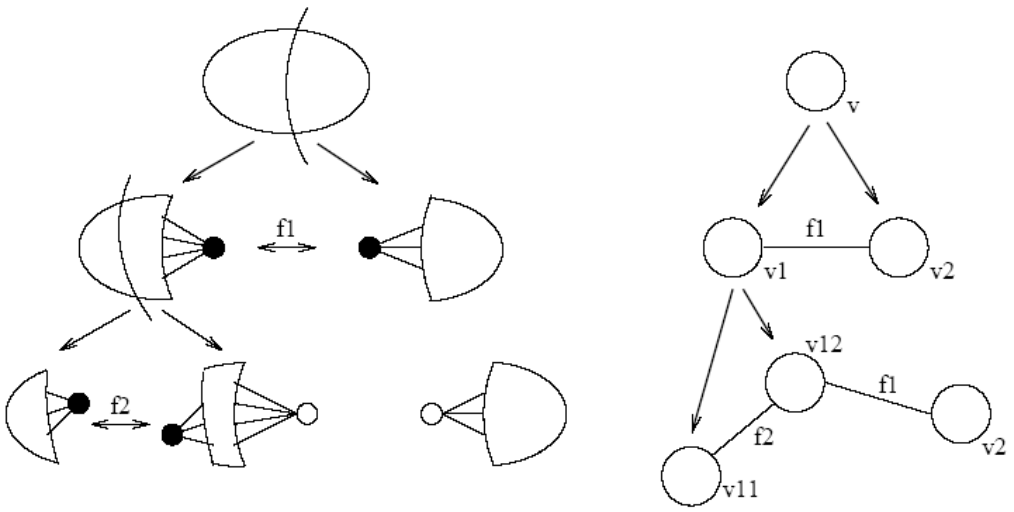
به T درخت معادل شاره بیشینه نیز می گویند. از بین درخت های با چنین خصوصیت حداقل یک درخت وجود دارد که حذف یال با کمترین وزن روی مسیر بین v_i و v_j به ما برش کمینه بین v_i و v_j را در G می دهد. به چنین درختی با این خصوصیت درخت Gomory-Hu, درخت برش کینه یا درخت برش می گویند. با در دست داشتن چنین درختی (که باز هم تاکید می کنم می تواند یگانه نباشد) ما شاره کمینه بین دو گره v_i و v_j را در گراف G با پیمودن مسیر بین v_i و v_j (که می دانیم چنین مسیر یگانه ای وجود دارد) و یافتن یال کمینه $e_{\min}$ پیدا می کنیم.

۴-۱-۱ الگوریتم ساختن درخت

نسخه‌ی اصلی

- قدم اول: الگوریتم شاره بیشینه را بین دو گره در گراف اجرا کن
- قدم دوم : خروجی قدم اول بعنوان وزن یال بکار رفته در درخت برش بکار می رود.

این الگوریتم بعد از ساختن $n - 1$ یال به اتمام می رسد. در مرحله یک چشمه و چاهک جدید برای قدم اول انتخاب می شود.



الگوریتم ساختن درخت برش کمینه :4-1

با این تفاوت که زیرمجموعه های شخصی در گراف اصلی بصورت یک گره نگاه می شوند. این کار باعث می شود که ورودی مساله ما کوچک تر از گراف اصلی شود و در نهایت مرتبه الگوریتم کاهش یابد. این دو مرحله بطور روشن در شکل۴-۱ به تصویر کشیده شده اند.[۱]

با گرفتن گراف اولیه به عنوان ورودی یک گره از درخت برش ایجاد می کنیم. آنگاه دو گره دلخواه از گراف انتخاب کرده و الگوریتم برش کمینه را روی آن دو اجرا می کنیم. با محاسبه برش کمینه درخت اولیه برش ما که یک گره داشت به دو گره جدید با یک یال که وزن آن برابر مقدار برش کمینه حاصله است تبدیل می شود. این دو گره نماینده دو بخش در دو سوی برش کمینه حاصله در گراف اصلی می باشد در دور بعدی الگوریتم ما روی یکی از این دو بخش برش کار می کنیم. این گراف جدید همان گراف اولیه است با این تفاوت که بخش دیگر برش را در یک گره منقبض کردیم و طبیعتا حالا باید دو گره ای که برای الگوریتم برش کمینه انتخاب می شوند شامل این گره انقباضی نباشند. حال شاره بیشینه جدید به‌ما دومین یال را از این درخت حاصل می کند. باز بطور مشابه گرهی که نماینده یک بخش از برش در گراف اولیه ما بود به دو گره جدید تقسیم و

[۱](اصل شکل از [۱۳])

یال بین این دو گره وزن شاره بیشینه را خواهد گرفت. اکنون گره سابق این درخت را باید به یکی از این دو گره جدید اتصال دهیم. طبیعتا اتصال ما به آن بخش از برش خواهد بود که در گراف واقعی هم گره ما بدان اتصال دارد.

با اجرای الگوریتم بدین منوال نهایتا به مرحله ای می رسیم که هر گره ما در درخت برش متناظر با زیرگرافی از گراف اصلی است که دقیقا یک گره اصلی دارد. این اتفاق دقیقا بعد از n-1 دور از الگوریتم می افتد. البته Gomory و Hu نشان دادند که این درخت براستی همان درخت برش کینه با همان خصوصیات دلخواه است.

نسخه Gusfield

نسخه های بهتری نیز از این الگوریتم ارائه شده است. در الگوریتم Gusfield در[۱۳] دیگر احتیاجی به انقباض یک گره وجود ندارد. با وجودی که این الگوریتم هم در n-1 محاسبه شاره بیشینه انجام می شود ولی تمام اعمال روی همان گراف اصلی انجام خواهد شد. در این حالت درخت برش ما با یک گراف ستاره اغاز می شود و در مراحل اجرای الگوریتم یال ها با توجه به مقادیر محاسبه شاره بیشینه بین گره‌ها جابجا خواهند شد.

در زیر این الگوریتم آورده شده است.

```
GusfieldCutTree(G(V, E))
   Initialize:
        Number all nodes in V from 1 to n
        Let p be an $n$ length vector initialized to 1
           // p corresponds to star tree T,
          // such that every node s points to p[s]
        For s=2 to n do
           Compute a minimum cut between nodes s and t = p[s] in G
           Let X be the set of nodes on the s side of the cut
```

```
        Let f(s,t) be the value of the minimum cut found
        Set fl[s] = f(s, t)
        For i = 1 to n do
           If ((i != s) and ( i in X) and (p[i] == t))
              Set p[i] = s
        If (p[t] in X)
           Set p[s] = p[t]
           Set p[t] = s
           Set fl[s] = fl[t]
           Set fl[t] = f(s,t)
     Output p and fl /* p corresponds to the minimum cut tree T
     with edges defined between all nodes i and p[i];
     the weights of edge (i, p[i]) is equal to fl[i].
```

به رغم ساختار ساده تر الگوریتم Gusfiled و پیچیدگی محاسباتی برابر این الگوریتم با الگوریتم اولیه Gomory-Hu نتایج تجربی در [۱۳] نشان می دهد که الگوریتم دوم با استفاده از هیوریستیک های مناسب قوی تر کار می کند.

به علت اعمال انقباضی پیاده سازی بهینه برای الگوریتم Gomory-Hu به بداهت الگوریتم Gusfield نمی‌باشد. یک پیاده سازی ناظریف حافظه جدیدی به گره انقباضی و یالش اختصاص می دهد. ممکن است که $\Omega(n^2)$ اختصاص حافظه حتی برای یک گراف خلوت بیانجامد. Tsioutsiouliklis در [۱۳] پیاده سازی پیشنهاد کرد که از مرتبه $O(\lg n)$ رکورد برای گره های اضافی و $O(n)$ رکورد برای یال های اضافی بود.

۴-۲ الگوریتم‌های دسته‌بندی و استخراج اجتماعات

یک درخت برش حاوی اطلاعات ارزشمندی هم درباره برش های کمینه در یک گراف G و هم ساختار درونی G می باشد.

لم ۱۰. فرض کنید u,v دو گره از درخت برش T باشد و $P(u,v)$ مسیر (واحد) بین آن دو در T. آنگاه برای هر گره w روی این مسیر شاره بیشینه $f(u,w)$ کمتر از $f(u,v)$ نخواهد بود.

لم ۱۱. فرض کنید u,v , $P(u,v)$ به ترتیب دو گره و مسیر بین آن دو در درخت برش کمینه باشد و $p = P(u,v)$ اندازه یا طول این مسیر برابر با تعداد برش های کمینه یگانهs,t (متفاوت) در گراف است.

بطور کلی در درخت برش کمینه گره های با اتصال شدید به هم کنار یکدیگر قرار می گیرند. البته چنین حقیقتی اگر بخواهد برای کلیه گره ها صدق کند در تعارض با همسایه های ایجاد شده قرار می گیرد و ممکن است منتهی به ساختار ضعیف درختی شود. در زیر الگوریتم پایه دسته بندی را می آوریم و توضیح می دهیم که براساس ساختار درخت برش دسته بندی می کند.

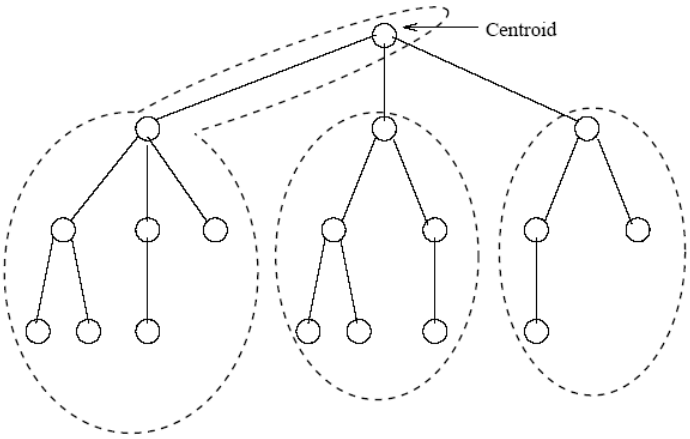
۴-۲-۱ الگوریتم

تعریف **کانون** (centroid) درخت برش :

تعریف ۱۲. کانون درخت T با مجموعه گره های V به صورت زیر تعریف می شود:
اگر $v \in V$ و $T_1, T_2, \ldots, T_d$ اجزا همبند ایجاد شده با حذف v از T باشند در اینصورت اگر $|T_i|$اندازه زیردرخت T_i تعریف شود آنگاه تعریف می کنیم برای هر گره:

$$N(v) = \max\{|T_i|\}$$

که در آن $1 \leq i \leq d$



تشخیص کانون در درخت :4-2

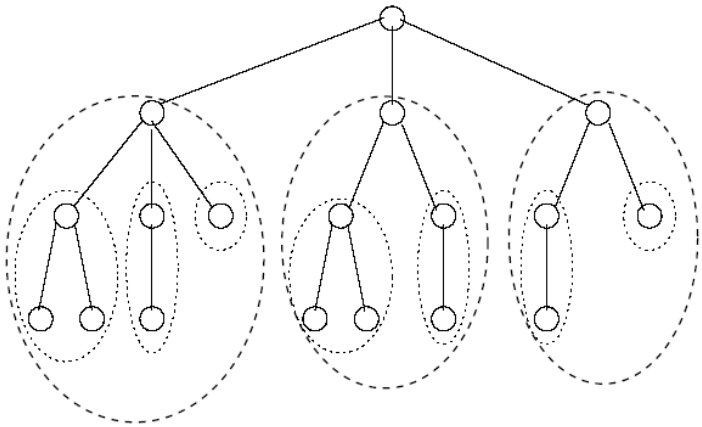
حال کانون درخت T تعریف می شود ([50]) به عنوان آن گرهی از درخت که می نیمم می کند مقدار $N(v)$ را روی تمام گره ها در V یا

$$N(v_c) = \min\{N(v)\}$$

که در آن $v \in V$

بطور شهودی آنچه از کانون یک درخت انتظار می رود آن است که با حذف کانون از درخت اجزا همبند باقیمانده بصورت حتی الامکان عادلانه تقسیم شده باشند. درخت T می تواند تا دو کانون داشته باشد که در اینصورت انتخاب هر کدام در الگوریتمی که در ادامه توضیح می دهیم امکان پذیر است. پس اگر $G(V, E)$ گراف ما برای عمل دسته بندی باشد و T درخت برش کمینه مربوط به آن باشد الگوریتم در ابتدا با پیداکردن کانون درخت شروع می کند. با تشخیص کانون ما درخت را از کانون آویزان می کنیم حال هر زیردرخت زیر کانون که هم اکنون ریشه درخت است یک دسته (cluster) است که توسط الگوریتم برگردانده می شود. حال باید ریشه را به یکی از این دسته ها منصوب کنیم. اگر e_c یال با بیشترین وزن متصل به v_c باشد ما v_c را به آن زیردرخت منسوب می کنیم که یک سر e_c در آن است.

شکل۴-۲ مثالی از الگوریتم ماست.



دسته بندی سلسله مراتبی (اصل شکل از]۱۳[) :4-3

۴-۲-۲ دسته‌بندی سلسله‌مراتبی

این الگوریتم به ما قدرت دسته بندی سلسله مراتبی را نیز فراهم می کند بدین ترتیب که می توانیم هر زیردرخت را خود به عنوان یک مجموعه جدید برای دسته بندی بگیریم. فقط بخاطر داشته باشید که دیگر لازم نیست کانون برای این زیردرختان تعیین کنید. شکل۴-۳ یک نمونه از دسته بندی سلسله مراتبی را نشان می دهد. [۲]

۴-۲-۳ نرمالسازی یالی

همانطور که Benezur [۱۵] نشان داد تعریف درخت برش کمینه فقط درباره گراف های بی جهت است. برای تصمیم این تعریف به گراف های جهت دار که موضوع مورد علاقه ما از جهت صفحات وب است از نرمالسازی یال های خروجی استفاده می کنیم.

اگر $G(V,E)$ گراف جهت دار و $v \in V$ یکی از گره های این گراف باشد آنگاه فرض کنید که $E_{\text{in}}, E_{\text{out}} \subseteq E$ به ترتیب مجموعه یال های ورودی و خروجی باشد و برای هر یال , w(x,y) $(x,y) \in E$ وزن مربوطه باشد.

[۲](اصل شکل از [۱۳])

حال نرمالسازی یال خروجی $(u,v) \in E_{\text{out}}$بصورت زیر انجام می شود.
یال (u,v) را بدون جهت فرض کرد و وزن جدید$\hat{w}(u,v)$ را به صورت زیر به آن نسبت می دهیم

$$\hat{w}(u,v) = \frac{w(u,v)}{\sum_{x \in V} w(u,x)}$$

لم ۱۳. این نحوه نرمالسازی G را به یک گراف بدون جهت با یال های نرمال شده تبدیل می کند.

اثبات. اثبات .هر گرهی وزن های یال های خروجی خود را فقط نرمال می کند و چون هر یال خروجی یک و فقط یک گره است حکم ثابت می شود. چنین نحوه نرمالسازی ما را به یاد روش های قدیمی تر مانند Pagerank , HITS می اندازد. □

۴-۲-۴ کاربرد در WWW

حال آماده ایم که الگوریتم پیشنهادی را روی وب اجرا کنیم. برای این کار یک مجموعه اولیه دانه از صفحات وب که مربوط به کلمه کاوشی (Search term) ما هستند پیدا می کنیم. این کار می تواند از طریق یکی از موتورهای جستجو انجام گیرد. حال برای ساختن گراف ورودی مساله ما به تمام بیوندهای ورودی و خروجی مجموعه دانه نگاه می کنیم. گره های این گراف مجموعه دانه بعلاوه همسایگان بلاواسطه‌ی آنها می باشد. حال می توان با نرمالیزه کردن و اجرای الگوریتم قسمت ۴-۲-۱ دسته ها را محاسبه کرد.

در اینجا خلاصه ای از نتایج تجربی که Tsioutsiouliklis در[۱۳] در مقایسه با Kleinberg برای عبارات gates, search engine, censorship, Java بدست آورد می آوریم. هر دو از یک مجموعه دانه با همان استراتری همسایه با یک فاصله (بلاواسطه) استفاده کردند. بعنوان مثال دسته هایی که kleinberg برای عبارت search engine پیدا کرده بود ۵ تای اولش همان موتورهای جستجوی معروف بودند در حالیکهTsioutsiouliklis در ۵ تای اول و عموما ۱۰ الی ۲۰ تای اول صفحاتی در مورد موتورهای جستجو و فراموتورهای جستجویی که قادر به پشتیبانی جستجوهای چندگانه و همزمان هستند بود. بنابراین نتیجه الگوریتم حداقل در این مثال بهتر و کلی تر از آنچه kleinberg با الگوریتم خود پیدا کرده بود بدست آمد.

۴-۲-۵ الگوریتم دسته بندی اجتماعات

بهتر است در ابتدا یک مساله بسیار مشابه که NP-complete می باشد را مطرح کنیم. این الگوریتم به دسته بندی k- برشی کمینه یا (Minimum k-cut clustering) معروف است و بصورت زیر بیان می شود.

اگر $G(V,E)$ یک گراف با مجموعه گره V و یال E باشد آنگاه طوری این گراف را به k زیرمجموعه افراز کنید که جمع یال های بین زیرمجموعه ها کمینه در بین تمام افرازها باشد. با استفاده از درخت برش کمینه می توان یک راه حل تقریبی با ضریب تقریب $(۲ - \frac{۲}{k})$ برای این مساله ارائه کرد. بدین صورت که اگر T درخت برش گراف بدون جهت $G(V,E)$ باشد با برداشتن $k - ۱$ یال با کمترین وزن در T k جزء همبند بدست می آید که G را به k دسته افراز می کند. مجموعه وزن یال های بین این دسته ها یک تقریب $(۲ - \frac{۲}{k})$ از مساله دسته بندی k برش کمینه‌است. با استفاده از این روش Wu, Lealy در[۱۶] ثابت کردند که هر برش درون دسته ای حداقل به بزرگی هر برش بین دسته ای در G می باشد. با وجود اینکه این روش بسیار راحت و سرراست می باشد ولی عیب اساسی دارد. با توجه به این حقیقت که معمولا یال های سبک در درخت برش ما به سمت برگ ها می روند (به تعریف کانون در بخش ۴-۲-۱ مراجعه کنید) دسته بندی با استفاده از این روش دسته های بسیار نامتعادلی تولید می کند که بیشتر آنها گره های تکی هستند. این همان نتیجه ایست که tsioutsiouliklis روی مجموعه داده هایش نیز بدست آورد.

۴-۲-۶ اجتماع ازوتریک

حال آماده می شویم که منظور خود را از یک اجتماع در وب بیان کنیم.

تعریف ۱۴. یک اجتماع (اجتماع ازوتریک) یک مجموعه S است بنحوی که بازای هر $s \in S$

$$\sum_{v \in S} w(s,v) \geq \sum_{v \in V-S} w(s,v)$$

به بیان دیگر **یک عضو اجتماع بیشتر به اجتماع خود تعلق دارد تا به بیگانگان!!**
در نهایت ما به دنبال دسته هایی هستیم که شبکه ما را به اجتماعات ازوتریک افراز کند. Tsioutsiouliklis در [۱۳] نشان می دهد که این مساله در حالت عمومی NP- تمام می باشد (کاهش به مساله افراز متعادل) با این وجود برای بسیاری از درختان برش می‌توان جواب‌های خوبی از این مساله یافت.

۴-۲-۷ تعمیم تعریف اجتماع Esoteric

S یک اجتماع p-esoteric است هرگاه به ازای هر $s \in S$

$$\sum_{v \in S} w(s,v) \geq p \sum_{v \in V} w(s,v)$$

یا معادلا

$$(۱-p) \sum_{v \in S} w(s,v) \geq p \sum_{v \in V-S} w(s,v)$$

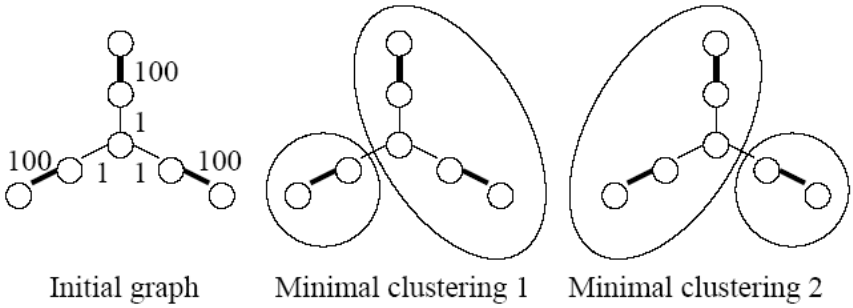
دقت کنید که با چنین تعریف تعریف قبلی ما از اجتماع ۱/۲-esoteric است . به پارامتر p که همواره از ۱/۲ کوچکتر پارامتر همبندی گفته می شود.

خصوصیات

۱) هر اجتماع باید حداقل دو گره داشته باشد. این ادعا نتیجه مستقیم تعریف اجتماع است.

۲) اگر C یک دسته بندی G باشد و $\hat{C}$ از اجتماع اجتماعات C تشکیل شده باشد آنگاه $\hat{C}$ هم یک دسته بندی مجاز برای G است. علت این خصوصیت نیز واضح است زیرا عمل اجتماع فقط وزن های داخل یک اجتماع راافزایش می دهد.

۳) اگر یک رده بندی غیرقابل کاهش را آن طور تعریف کنیم که رده بندی است که نتوان در آن دسته ها را کوچکتر کرد آنگاه می توان بجای یک دسته بندی کمینه چندین دسته بندی



دو دسته‌بندی غیر قابل کاهش متفاوت (شکل از]۱۳[:4-4

غیرقابل کاهش داریم. شکل۴-۴ چنین دسته بندی را نشان می دهد. [۳]

۴-۳ جستجو برای اجتماعات

برای یافتن اجتماعات ازوتریک از طریق شاره بیشینه باید پارامتر همبندی را برابر ۱/۲ قرار دهیم. اگر t,s به ترتیب منبع و چاهک برای گراف G باشد برش کمینه $s-t$ را در G در نظر بگیریم. اگر S,T به ترتیب زیر مجموعه های دو سوی این برش باشد بنحوی که $s \in S$ و $t \in T$ آنگاه بردن هر گره از $S-s$ به T اندازه برش را فقط می تواند کاهش دهد. این حقیقت برای عکس قضیه نیز درست است. پس هر گره در T,S بیشتر به خودشان متصل هستند تا قسمت متقابل پس T,S هر دو از قانون اجتماع پیروی می کنند. استثنا می تواند خود t,s باشد. براساس چنین مشاهده ای ما می توانیم با استفاده از درخت برش کمینه اجتماعات ازوتریک را بیابیم.

تعریف ۱۵. یک یال e در درخت برش کمینه T یک یال کمینه محلی است اگر هر دو س‌ر v_1 و v_2از e حداقل یک یال در T داشته باشند که وزنش بزرگتر از e باشد.

[۳]اصل شکل از [۱۳]

برای پیداکردن چنین یال هایی یک راه ساده وجود دارد. برای تمامی گره های داخل درخت یال با بیشترین وزن مجاورشان را علامت بزنیم. حال تمام یال های بدون علامت یال کمینه محلی هستند.

لم ۱۶. اگر T یک درخت برش کمینه برای گراف G باشد و e یک یال کمینه محلی آنگاه حذف e دو اجتماع از وتریک به ما می دهد.

اثبات. برای اثبات فرض کنید v_1 و v_2 دو گره دو سر e می باشند. یال e معادل برش کمینه v_1, v_2 در G می باشد. بنابراین دو طرف این برش دو اجتماع از وتریک هستند اگر
الف) هیچ کدام از این دو طرف یک گره تنها نباشند.
ب) اگر انتقال هر کدام از v_1 و v_2 طرف دیگر یک برش با اندازه کمتر به ما ندهد.
الف هیچ گاه اتفاق نمی افتد چون در اینصورت e به عنوان پروزن ترین یال v_1 یا v_2 از کاندیداهای یال های کمینه محلی خط می خورد.
ب) فرض کنید بدون از دست دادن کلیت که انتقال v_1 به قسمت برش v_2 با مقدار کمتری به ما بدهد. اگر $e_{\max} \neq e$ یال مجاور v_1 با بزرگترین وزن باشد و $v_{\max}$ و v_1 دو سر این یال باشند با انتقال v_1 به قسمت دیگر برش v_1 از $v_{\max}$ جدا می شود پس نمی تواند وزنی کمتر از $e_{\max}$ پیدا کند که می دانیم بیشتر از وزن e می باشد پس ثابت شد که حذف e اجتماعات از وتریک معتبر برای ما پدید می آورد براساس لم قبل با الگوریتمی ارائه می کنیم که اجتماعات ازوتریک را در G پیدا می کند. □ ایده اصلی الگوریتم بدین صورت است که ابتدا دو اجتماع پیدا می کنیم و یکی از این دو را به یک گره منقبض می کنیم بعنوان مثال X در تکرارهای بعدی باز دو اجتماع دیگر پدید می آید و هر دفعه یکی از این اجتماعات جذب X می شود. در نهایت الگوریتم هنگامی خاتمه می یابد که هیچ یال کمینه محلی در درخت برش کمینه ما وجود نداشته باشد.
در زیرالگوریتم شبکه مربوطه آمده است.

```
Community Cluster (G(V,E))
  Construct T as the minimum cut tree of G
  if (T has no local minimum edge)
   Return single, trivial cluster covering entire G
```

```
  else
   let X={ } be empty set
   let e be a local minimum edge in T
   while (e non-empty)
     let S1, S2 be the two communities formed after removing e
     if (X= { })
        output S1, S2 as next community and
        contract if into X
     else if (X is a member of S1)
        output Sz as next community and contract it
        into X
     else { // if X S2
        Output S1 as next community and contract it
        into X}
  Calculate new minimum cut tree T
  Let e be a local minimum edge in new T
 Return
```

Tsioutsiouliklis در [۱۳] نشان داد که این الگوریتم اجتماعات ازوتریک ۱/۲ تولید می کند. توجه کنید که این الگوریتم لروماگراف را افراز نمی کند بلکه تنها اجتماعات را از G استخراج می کند.

۴-۳-۱ تسهیل شرایط

همانطور که توضیح داده شد الگوریتم در مراحل مختلف خود یال های کمینه محلی جدید را از روی درخت محاسبه شده پیدا می کند. یک ایده جایگزین این است که در همان ابتدا تمام یال های کمینه محلی را از درخت اولیه برش حذف کنیم. البته دسته هایی که با این عمل حاصل می شوند

ممکن است لزوما اجتماعات از وتریک نباشند ولی با این حال می توانیم لم زیر را برای آنها ثابت کنیم.

لم ۱۷. اگر T درخت کمینه برش از گراف G باشد و C دسته بندی حاصله توسط حذف کلیه یال های کمینه محلی از T باشد آنگاه فرض کنید v_1 و v_2دو گره در یک دسته مشابه از C باشد و $c(v_1)$و $c(v_2)$ بزرگترین مقدار برش کمینه که v_1 و v_2 را از سایر گره ها در G جدا می کند باشد آنگاه برای برش کمینه $c(v_1, v_2)$ بین v_1 و v_2ما داریم

$$c(v_1, v_2) \geq \min c(v_1), c(v_2)$$

اثبات. می دانیم گره های روی مسیر بین v_1 و v_2 متعلق به C_1 یعنی دسته ایست که v_1 و v_2 بدان تعلق دارند. نظر به وزن های روی یال های $P = p(v_1, v_2)$ (مسیر) ادعا می کنیم که گره $v \in P$ وجود دارد به نحوی که یال های از هر دو گره v_1 و v_2 اوزان غیر نزولی به v دارند چرا که در غیر اینصورت یک یال کمینه محلی در مسیر پیدا می شد که باید حذف می شد.

اماv حداقل با یکی ازv_1 و v_2 متفاوت است فرض کنید $v \neq v_1$ حال گره همسایه v_1 (مثلا v_3) یال (v_1, v_3) را علامت نزده است چون اوزان یال ها به سمت v فقط زیاد می شوند. پس توسط v_1 علامت زده شده است. پس (v_1, v_3) همان $c(v_1)$ طبق تعریف است. در عین حال (v_1, v_3)یک حد بالایی برای برش کمینه مابین v_1 و v_2 است پس لم ثابت می شود.

$$c(v_1, v_2) \geq \min c(v_1), c(v_2)$$

□

فصل ۵

الگوریتم های تقریبی برای پیداکردن اجتماعات

در این فصل ما به روش Charikar در[۱۷] اشاره می کنیم و این روش را روی گراف وب با خصوصیات خاص خودش (رجوع کنید به فصل ۲) بررسی می کنیم. تا اینجا روشن است که

۱) تعریف مشخصی برای اجتماعات وب وجود ندارد و هر تعریف ارائه شده جنبه هایی از اجتماعات را تاکید و جنبه هایی را نادید می گیرد.

۲) لزومی به حل مساله اجتماعات بصورت دقیق وجود ندارد و راه حل های تقریبی با ضریب تقریب مناسب و مرتبه پایین به راه حل های دقیق مرتبه بالا ارجح می باشند.

در این بخش به بیان راه حل های تقریبی برای هر دو نوع گراف (بی جهت و جهت دار) می پردازیم و یک پیاده سازی کارا را برای راه حلمان شرح می دهیم.

۵-۱ گراف بدون جهت

تعریف ۱۸. اگر $G(V,E)$ گراف بدون جهت و $S \subseteq V$ باشد. مجموعه یالی القایی S یا $E(S)$ بصورت زیر تعریف می شود:

$$E(S) = \{ij \in E : i \in S, j \in S\}$$

تعریف ۱۹. اگر $S \subseteq V$ باشد. چگالی $f(S)$ این زیرمجموعه S از V بصورت زیر تعریف می شود:

$$f(S) = \frac{|E(S)|}{|S|}$$

در نهایت چگالی یک گراف بدون جهت G(V,E) تعریف می شود:

$$f(G) = \max_{S \subseteq V}\{f(S)\}$$

لم ۲۰. $۲f(S)$ میانگین درجات زیر گراف القایی توسط S و $۲f(G)$ بیشینه میانگین درجات روی تمام زیرگراف های القایی در G می باشد.

اثبات. جمع تمام درجات روی رئوس برابر $۲E$ می باشد. پس حکم ثابت می شود. □

مساله محاسبه $f(G)$ به مساله شلوغ ترین زیرگراف مشهوراست که توسط ترفندهای الگوریتم شاره قابل حل است. یک مساله مشابه مساله شلوغ ترین k-زیرگراف است که روی تعداد گره ها در بیشینه $f(G)$ قید k می گذارد. این دو مساله ی بسیار جالب برای ارائه راه حل های تقریبی می باشد و نسبتا باز هستند.

۵-۱-۱ محاسبه $f(G)$

محاسبه $f(G)$ بوسیله ترفندهای برنامه نویسی خطی (Linear programming) قابل انجام است. Charikar در[۱۷] علاوه بر این محاسبه یک الگوریتم 2-approximation برای این کار ارائه کرده

است. این کار بوسیله حذف متناوب گره با کمترین درجه صورت می گیرد. در ادامه مشابه همین بحث را برای گراف های جهت دار انجام می دهیم.
LP زیر را در نظر بگیرد.

$$\max \sum_{ij} x_{ij}$$

$$\forall ij \in E : x_{ij} \leq y_i$$

$$\forall ij \in E : x_{ij} \leq y_j$$

$$\sum_i y_i \leq ۱$$

$$x_{ij}, y_i \geq ۰$$

لم ۲۱. برای هر زیرمجموعه مقدار LP فوق حداقل برابر $f(S)$ است.

اثبات. اگر ما یک جواب امکانپذیر به این LP با مقدار $f(S)$ بدهیم حکم ثابت است. اگر $x = \frac{۱}{|S|}$باشد برای هر $i \in S$ قرار می دهیم $\hat{y_i} = x$ و برای هر $ij \in E(S)$ قرار می دهیم $\hat{x_{ij}} = x$. و سایر متغیرها را برابر صفر قرار می دهیم حال $\sum_i \hat{y_i} = |S|.x = ۱$پس $(\hat{x}, \hat{y})$ شرایط مساله را ارضاء می کنند. جواب این LP برابر می باشد با

$$|E(S)|.x = \frac{|E(S)|}{|S|} = f(S)$$

□ لم زیر نشان می دهد که همواره $S \in V$ وجود دارد که چگالی آن بیشتر از هر جواب امکانپذیر برای LP مذکور باشد.

لم ۲۲. با داشتن هر جواب امکانپذیر از LP فوق با مقدار v می توان $S \in V$ ساخت بشرطی که

$$f(S) \geq v$$

اثبات با پارامتریزه کردن مجموعه S و برهان خلف بدست می آید که در [؟] بطور کامل آمده است.

قضیه ۲۳. اگر مقدار جواب بهینه LP فوق را با $OPT(LP)$ نشان دهیم.

$$\max_{S \subseteq V}\{f(S)\} = OPT(LP)$$

همچنین زیرمجموعه S برای جواب بهینه از روی راه حل بهینه LP قابل استخراج است.

۵-۲ الگوریتم تقریبی با ضریب ۲ برای محاسبه $f(G)$

هدف ما این است که در گراف G یک زیرگراف شلوغ بدست آوریم. طبیعتا یک راه حل بدیهی این است که آن گره هایی از گراف که درجه کمتری دارند را بدور اندازیم تا شانس خود را برای بهبود چگالی بیازماییم. این ایده یک الگوریتم حریصانه را پیشنهاد می کند.

مشابه چنین الگوریتمی برای میانگین درجه بزرگ برای تعداد گره های با عدد k توسط Asahiro و سایرین در[۱۸] تحلیل شده است.

برای این مساله ما از تحلیل Charikar در اثبات ضریب ۲ آن و پیاده سازی پیشنهادی او استفاده می کنیم. الگوریتم همواره یک زیرمجموعه $S \in V$ از گره های گراف را نگاه می دارد. در ابتدا $S \longleftarrow V$ در هر دور الگوریتم $i_{\min}$ یعنی گره با کمترین درجه تشخیص داده می شود الگوریتم را $i_{\min}$ از مجموعه S حذف کرده و به دور بعدی می رود. شرط پایان الگوریتم زمانی است که S تهی می شود. برای تمام مجموعه های S که در طول اجرای الگوریتم ساخته شد آن مجموعه ای که بیشترین مقدار $f(S)$ را داشت بعنوان خروجی برگردانیده خواهد شد. در زیراین الگوریتم بصورت شبه که آمده است.

```
ApproxCom( Graph G(V, E))
 initialize S = V;
 S_max = V;
 f_max = E/V;
 While(notEmpty(S))
   Let i_min be the vertex with the minimum degree in S;
   Remove i_min from S;
   Update S and E(S);
   f = E(S)/S;
   if (f > f_max)
      f_max = f;
      S_max = S;
 Return S_max;
```

قضیه ۲۴. الگوریتم فوق یک راه حل ۲ تقریب به مساله $f(G)$ می باشد.

اثبات. در حقیقت باید ثابت کنیم که

$$\max_{S \subseteq V}\{f(S)\} \leq ۲v$$

که v مجموعه ایست که ما در نهایت در الگوریتم بر می گردانیم. در وهله اول ما گراف بدون جهت را به صورت جهت دار نگاه می کنیم و $d(i)$ را درجه ورودی راس i در نظر می گیریم. پس در عمل به هر یال ij راس i یا j را نسبت داده ایم حال تعریف می کنیم.

$$d^{\max} = \max_{i}\{d(i)\}$$

حال اگر $S_{\max}$ آن مجموعه S باشد که $f(S)$ را بیشینه می کند.

$$|E(S_{\max})| \leq |S_{\max}|.d^{\max}$$

$$f(G) = \frac{|E(S_{\max})|}{|S_{\max}|}$$

حال انتساب یال ها به یکی از دو سرشان در حین الگوریتم بدین صورت انجام می شود: در ابتدا تمامی یال ها دست نخورده می باشند. وقتی $i_{\min}$ در S حذف شد؛ تمامی یال هایش که به سایر رئوس S وصل هستند برچسب او را می خورند. ثابت حلقه بدین صورت است که کلیه یال ها بین رئوس در مجموعه کنونی S بدون برچسب می باشند در حالی که باقی یال ها تماما منتسب شده اند. پس در انتهای الگوریتم تمامی یال ها منتسب شده اند چون مجموعه S تهی شده است. حال ثابت می کنیم که

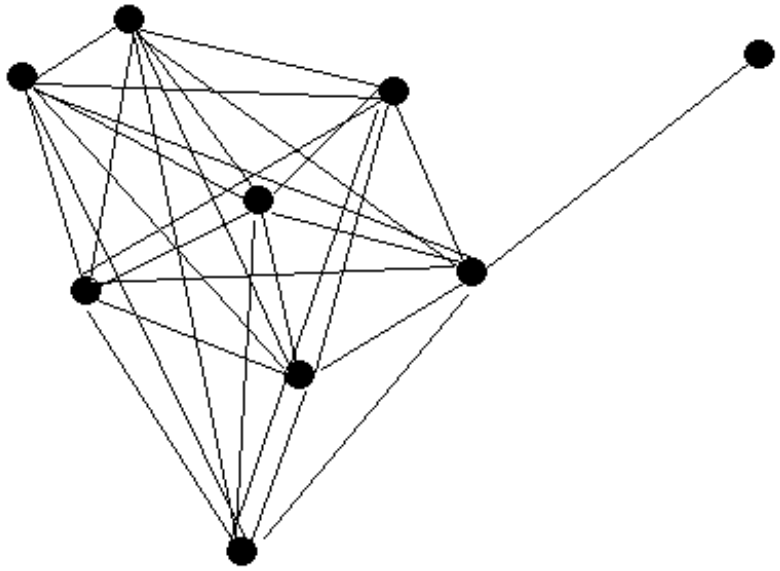
$$d^{\max} \leq ۲v$$

که v بیشینه مقدار $f(S)$ است که توسط الگوریتم حریصانه بدست می آید. برای نشان دادن این مساله الگوریتم را در یک دور آن در نظر می گیریم. از آنجایی که $i_{\min}$ طوری گزینش شده است که راس با کمترین درجه در S باشد پس درجه آن نمی تواند از $۲|E(S)|/|S|$ بیشتر باشد و می دانیم که این مقدار هم از $۲v$ بیشتر نیست. همچنین در نظر داشته باشید که یک راس خاص تنها هنگامی یال بخود منتسب می کند که در حال حذف شدن باشد. پس الگوریتم ما یک تقریب ضریب ۲ برای $f(G)$ می باشد. □

۵-۳ پیاده سازی الگوریتم

براحتی می توان یک پیاده سازی $O(n^۲)$ برای الگوریتم فوق که n تعداد گره های گراف و m تعداد یال های آن باشد پیشنهاد کرد. بدین منوال که ما درجه رئوس را در زیر گراف القایی S نگاه می داریم. در هر دور از الگوریتم راس با کمترین درجه تشخیص و حذف می شود و درجه سایر رئوس باقیمانده بروز می شوند. هر دو این اعمال در $O(n)$ قابل انجام هستند پس الگوریتم در $O(n^۲)$به اتمام می رسد.

در اینجا ما پیشنهاد Charikar را برای اجرای این الگوریتم در $O(n)$ ارائه می کنیم. برای این کار n لیست از گره ها تهیه می کنیم که لیست iام حاوی عناصر از درجه i می باشد. در هر دور از



گراف ناهنجار :5-1

الگوریتم عنصر با کمترین درجه حذف شده و درجات عناصر همسایه اش بروز می شوند. این کار باعث انتقال این عنصر به لیست های دیگر می شود. توجه کنید که اگر حذف این راس باعث شدن $i_{\min}$ گرهی در دور بعدی شده باشد ما کافی است که فقط با فاصله یک از درجه $i_{\min}$ همین دور بدنبال $i_{\min}$ دور بعد بگردیم به بیان دیگر اگر d درجه راس $i_{\min}$ می باشد در دور بعد ابتدا در لیست d و $d+۱$ و $d-۱$ را نگاه می کنیم. البته می تواند این سه لیست خالی باشد ولی هنوز کار ما به اتمام نرسیده باشد به عنوان مثال گراف را ۵-۱ نگاه کنید.

با حذف $i_{\min}$ در دور اول هر کدام از ۸ راس باقیمانده با درجه ی ۷ می تواند انتخاب شود. پس لزوما جستجو در $d-۱, d$ و $d+۱$ کافی نیست. بلکه باید درجه اولین لیست غیرتهی را ثبت کنیم که اگر $d-۱$ و d و $d+۱$ باشد با جستجو حاصل می شود. در نهایت اگر مرتبه اینگونه جستجوها را ثابت بگیریم جواب در $O(m+n)$ حاضر می شود.

۵-۴ انتخاب $d^{\max}$ بعنوان حد بالایی

در ارائه اثبات ضریب تقریب ایده حضور نابدیهی است در حالی که خود را در دوگان LP مساله ظاهر می کند. LP زیر دوگان LP مساله فوق است:

$$\min \gamma$$

$$\forall ij \in E : \alpha_{ij} + \beta ij \geq ۱$$

$$\forall i : \gamma \geq \sum_i \alpha_{ij} + \beta_{ji}$$

$$\alpha_{ij}, \gamma \geq ۰$$

۵-۵ مساله‌ی معادل برای گراف جهت دار

ابتدا به بیان تعاریف معادل می پردازیم. اگر $G(V,E)$ یک گراف جهت دار و $S,T \subseteq V$ ، $E(S,T)$مجموعه ای از یال هاست که از S به T می رود به بیان دیگر

$$E(S,T) = \{ij \in E : i \in S, j \in T\}.$$

پس بدین ترتیب چگالی جفت مجموعه S,T که آن را با $d(S,T)$ نمایش می دهیم بصورت زیر تعریف می شود.

$$d(S,T) = \frac{|E(S,T)|}{\sqrt{|S||T|}}$$

و مشابه گراف های بدون جهت چگالی $d(G)$ را برای یک گراف جهت دار $G(V,E)$ را تعریف می کنیم.

$$d(G) = \max_{S,T \subseteq V} \{d(S,T)\}$$

LP که برای مساله جدید تعریف می شود بصورت زیر است.

$$\max \sum_{ij} x_{ij}$$

$$\forall ij \in E : x_{ij} \leq s_i$$

$$\forall ij \in E : x_{ij} \leq t_j$$

$$\sum_i s_i \leq \sqrt{c}$$

$$\sum_j t_j \leq \frac{۱}{\sqrt{c}}$$

$$x_{ij}, s_i, t_j \geq ۰$$

برای گراف جهت دار هم خصوصیات مشابه قبل قابل اثبات است. در اینجا ما این خصوصیات را بدون بیان اثبات می آوریم.

۱) اگر $S,T \subseteq V$ و $c = \frac{|S|}{|T|}$ باشند. مقدار بهینه $LP(c)$ تعریف شده در بالا حداقل برابر $d(S,T)$ می باشد.

۲) با داشتن یک جواب امکان پذیر از $LP(c)$ با مقدار v ما می توانیم زیرمجموعه های S,T را از V طوری بیابیم که $d(S,T) \geq v$

۳) $\max_{S,T \subseteq V}\{d(S,T)\} = \max_c\{OPT(LP(c))\}$

۴) مجموعه S,T که بیشترین مقدار $d(S,T)$ را به ما می دهد از روی حل بهینه $LP(c)$ قابل محاسبه است.

۵-۶ الگوریتم حریصانه برای محاسبه $d(G)$

مشابه الگوریتمی که برای $f(G)$ توضیح داده شد برای یافتن $d(G)$ بکار می رود با این تفاوت که درجه‌ی یک گره در S یال هایی است که از آن گره به مجموعه T متصل است و درجه‌ی یک گره در T یال هایی است که از گره های موجود در S به آن اشاره شده است.

در نهایت راس حذفی ما باید با یک استراتژی از بین گره های S یا T انتخاب شود. جفت زیرمجموعه S,T که مقدار $d(S,T)$ را بیشینه می کند بعنوان خروجی الگوریتم برگردانده می شود.

Charikar نشان می دهد که الگوریتم فوق یک راه حل تقریبی با ضریب ۲ برای محاسبه $d(G)$ می باشد. البته یک حدس خوب برای مقدار c برای کارکرد مناسب الگوریتم لازم است. برای یک c خاص الگوریتم قابل اجرا در $O(m+n)$ می باشد. جستجو از مرتبه $O(\frac{\log n}{\varepsilon})$ برای C لازم است که یک جواب $2+\varepsilon$ تقریب بدست آید.

فصل ٦

نتایج تجربی روی الگوریتم تقریبی Charikar

در ارتباط با سایر روش های شناسایی اجتماعات وب نتایج تجربی به تفصیل در کارهای اخیر یافت می‌شود. به عنوان نمونه به [۱۳] مراجعه کنید. با این وجود، روش تقریبی Charikar به منظور شناسایی اجتماعات وب تا کنون آزمایش نشده است.

الگوریتم تقریبی هر چند تضمین مناسبی از جهت زمان اجرا در اختیار ما می‌نهد، تنها هنگامی بکار شناسایی اجتماعات می‌آید که متریک‌های یک اجتماع را برآورده سازد.

در این بخش نتایجی که از اجرای الگوریتم تقریبی Chariker در فصل ۵ روی مجموعه داده واقعی وب بدست آمده را گزارش و بررسی می‌کنیم. در ابتدا تغییرات اعمالی روی الگوریتم تقریبی را شرح داده، سپس به تحلیل نتایج می‌پردازیم.

۶-۱ الگوریتم تقریبی استخراج اجتماعات وب

۶-۱-۱ اگر کل گراف درحافظه اصلی برنامه گنجانیده نشود

الگوریتم توضیح داده شده در فصل پیشین با فرض گنجانیده شدن کل گراف ورودی در حافظه اصلی برنامه می‌باشد. همانطور که نشان دادیم، چنین فرضی لزوما در مورد گراف غول‌آسای وب صحیح نیست. پس در عمل الگوریتم زیر را برای شناسایی اجتماعات گراف وب پیشنهاد می‌کنیم.

```
Approx-Huge-Web (set: S, integer: k)
  while number of iteration is less than desired do
        set G to a crawl from S of depth k
        set X = Approx-com ( G );
        Score all the vertices in X due to their degree
        Add non-seed high-scores in X to S
 return X
```

رویه فوق بدین صورت است که در ابتدا یک مجموعه دانه که قابل بارگذاری در حافظه اصلی است را در نظر می‌گیریم. تا عمق k این مجموعه را خزش می‌کنیم و گراف ورودی ما به رویه Approx-com در فصل ۵ چنین ساخته می‌شود.

حال اگر X خروجی رویه Approc-com باشد، رئوس با درجه بالا که در S نبودند به S در مرحله بعد اضافه می‌شوند. در پیوست کد کامل به زبان جاوا آمده است.

۶-۱-۲ اگر بخواهیم همه اجتماعات را شناسایی کنیم

الگوریتم Approx-com توضیح داده شده در فصل قبل، تنها شلوغ ترین زیرگراف را برمی‌گرداند. برای شناسایی کلیه اجتماعات می‌توان الگوریتم زیر را ارائه داد.

```
Approx-all-communities ( Graph (G,V), int n )
 while i<n
   compute the ith community from Approx-com (G)
   let S_i be the output of Approx-com (G)
   remove S_i from G
 return S_1, S_2, ... , S_n
```

در الگوریتم فوق فرض بر این است که کل گراف در حافظه اصلی نگهداری می‌شود.

۶-۲ داده های واقعی

برای محک الگوریتم تقریبی Charikar از یک مجموعه ۶۰۰۰۰ راسی که خود یکی از ۱۰ مجموعه خزش و جمع‌آوری شده از صفحات وب، در دانشگاه شریف است. این مجموعه در سال ۲۰۰۴ توسط تیم طراحی موتور جستجو در آزمایشگاه دکتر قدسی، دانشکده مهندسی کامپیوتر خزش شده است.

برای اجرای الگوریتم، از یک مجموعه ۸۰۰ گرهی به منظور S یا مجموعه دانه استفاده کردیم. k را در الگوریتم برابر ۱ قرار دادیم. گراف شکل ۶-۱ به عنوان خروجی الگوریتم بدست آمد.

صفحات متناظر با هر گره در زیر آمده است.

1 http://editors.dmoz.org/editors/editcat.cgi?cat=Computers/Security

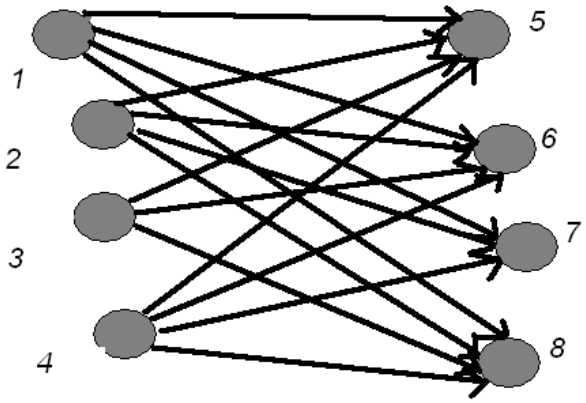
2 http://p2p.wrox.com/default.asp?CAT_ID=27

3 http://s.teoma.com/search?q=books

4 http://www.altavista.com/cgi-bin/query?q=programming\%20languages

5 http://www.pcisystems.com

6 http://www.phytec.com/



گراف خروجی الگوریتم تقریبی روی مجموعه دانه‌ی ۸۰۰ تایی :6-1

7 http://www.pond.ie

8 http://www.square1industries.com

الگوریتم در دور دوم با حذف گره های دور اول به کار خود ادامه می‌دهد. اجتماع تشخیص داده شده در دوم در گراف ۶-۲ نمایش داده شده است.

صفحات متناظر با هر گره در زیر آمده است.

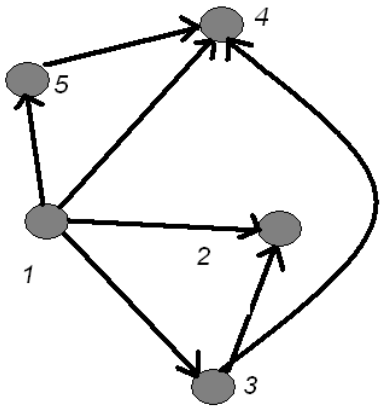
1 : dmoz.org/Games/Video_Games/
Console_Platforms/Sony/PlayStation/index.html

2 : www.vidgames.com

3 : www.playstationultra.com

4 : directory.google.com/Top/Games/
Video_Games/Console_Platforms/Sony/PlayStation/

5 : dmoz.org/cgi-bin/apply.cgi?where=
Games2FVideo_Games2FConsole_Platforms2FSony2FPlayStation



گراف خروجی الگوریتم تقریبی روی مجموعه دانه‌ی ۸۰۰ تایی در دور دوم :6-2

۶-۳ تحلیل نتایج

نتایج بدست آمده توسط الگوریتم تقریبی، انتظارات ما را در شناسایی درست اجتماعات برآورده می‌سازد. میزان های bibliometric و co-citaion ارائه شده در الگوریتم HITS را بر روی مجموعه جواب محاسبه می‌کنیم. برای زیرگراف شکل ۶-۱ متریک ها بصورت زیر می‌باشد.

$$Auth-score(۵) = \sum_{i=۱,۲,۳,۴} Hub-score(i)$$

$$Auth-score(٦) = \sum_{i=۱,۲,۳,۴} Hub-score(i)$$

$$Auth-score(۷) = \sum_{i=۱,۲,۴} Hub-score(i)$$

$$Auth-score(۸) = \sum_{i=۱,۲,۳,۴} Hub-score(i)$$

$$Hub-score(۱) = \sum_{i=۵,٦,۷,۸} Auth-score(i)$$

$$Hub-score(۲) = \sum_{i=۵,۶,۷,۸} Auth-score(i)$$

$$Hub-score(۳) = \sum_{i=۵,۶,۸} Auth-score(i)$$

$$Hub-score(۴) = \sum_{i=۵,۶,۷,۸} Auth-score(i)$$

در دور دوم هم یک اجتماع بامعنی استخراج شده است.
چند نکته قابل توجه در شکل‌های ۶-۱ و ۶-۲ در زیر آمده است.

- نسبت تعداد یال به راس در گراف ۶-۱ به مقدار $\frac{۱۵}{۸}$ یا**۱.۸۷۵** می‌باشد.
- در گراف ۶-۱ یک هسته‌ی دوبخشی چهار در سه وجود دارد.
- گراف ۶-۱ با دریافت یک یال دیگر به گراف کامل دوبخشی $K_{۴,۴}$ تحویل می‌شود.
- نسبت تعداد یال به راس در گراف ۶-۲ تا مقدار $\frac{۶}{۵}$ یا**۱.۲** کاهش یافته است.
- با توجه به URL های ذکر شده، هر دو جواب، دو اجتماع صحیح است که گرد دو موضوع معنایی پدیدار شدند.
- اعمال متریک های دوبخشی در مورد ۶-۲ هم نتایج قابل قبولی می‌دهد.

در دورهای بعدی الگوریتم، باز هم $f(G)$ کاهش پیدا می‌کند. با وجودی که الگوریتم تقریبی اساسا بر پایه اجتماعات متراکم عمل می‌کند، در این مثال مشاهده کردیم که یک جواب خوب به مساله اجتماعات دوبخشی نیز می‌تواند باشد که مرتبه زمانی بسیار مطلوبی نسبت به روش‌های پیشین دارد.

فصل ۷

نتیجه گیری

در این پایان نامه به بررسی روش های مختلف تشخیص اجتماعات موجود در وب پرداخته شده است. در ابتدا اهمیت این مساله، جنبه های مختلف آن توضیح داده شده و چند مساله مشابه معرفی شده اند. در ادامه خصوصیات اصلی گراف وب را بر شمرده ایم و یک مدل ریاضی برای شبیه سازی این گراف ارائه کردیم.

در ادامه به تشریح الگوریتم های مختلف تشخیص اجتماعات وب پرداختیم که‌چندی از نتایج باختصار در زیر آمده است.

- متریک ها و میزان هایی برای تعیین میزان شباهت دوصفحه ای که به یکدیگر اتصال مستقیم ندارد تعریف می شود.
- هسته های دو بخشی براساس دید هاب و مرجع بودن در یک گراف اساس تعریف اجتماعات وب در روش HITS است.
- هر دو روش HITS و PageRank تفسیرهای ماتریسی و قدم زدن تصادفی دارند و به یکدیگر قابل تبدیل هستند.
- روش بیشینه شاره دید متفاوتی نسبت به HITS و PageRank از اجتماعات وب دارد.
- روش درخت برش کمینه یک روش ساخت یافته براساس الگوریتم شاره است.
- الگوریتم تقریبی نتایج قابل قبولی با پیچیدگی خطی برای پیداکردن اجتماعات می دهند.

مسائل فراوانی را هر کدام از روش های تشخیص اجتماعات پیش روی ما باز می گذارند. در روش تقریبی Charikar یک زمینه تحقیقات فراتر بکارگیری خصوصیات گراف وب معرفی شده در بخش نخست است برای کاهش ضریب تقریب تئوریک و یا بهبود الگوریتم در عمل. حل مساله با تعمیم تعریف اجتماعات درروش شاره می تواند یک چالش جدی در این عرصه باشد. همچنین استفاده از اطلاعات متنی علاوه بر ساختار پیوندی گراف وب یک پیشنهاد عملی خوب است.

فصل ۸

واژه‌نامه

اجتماعات ..... Communities

امکان پذیر ..... feasible

انقباض ..... contraction

پرسش ..... query

تقریب ..... approximation

تک عضوی ..... singleton

توزیع توانی ..... power low distribution

جزء قویا همبند ..... strongly connected component

جزء همبند ضعیف ..... weakly connected component

چاهک ..... sink

چشمه ..... source

حریصانه ..... greedy

خزش ..... crawl

دامنه ..... domain

دسته ..... cluster

دسته بندی ..... clustering

دانه ..... seed
ساختار پیوندی ..... link structure
شاره ..... flow
صفحه وب ..... Webpage
غیر قابل کاهش ..... minimal
قدم‌زن تصادفی ..... random walker
کلمه کاوشی ..... search term
گره ..... node
متراکم ..... dense
مرجع ..... authority
میانگین فاصله اتصالی ..... average connected distance
میزبان ..... host
هسته دوبخشی ..... bipartite core
یال ..... edge

کتاب‌نامه

[1] K. Bharat and M.R. Henzinger. Improved algorithms for topic distillation in a hyperlinked environment. *Proc. ACM SIGIR, 1998.*

[2] R. Albert, H. Jeong, and A. Barabasi. Diameter of the World Wide Web .Nature 401:130-131, 1999.

[3] S. Brin and L. Page. The anatomy of a large-scae hyper textual Web search engine. *Proc. 7th WWW Conf. 1998*

[4] N. Gilbert. A simulation of the structure of academic science. Sociological Research Online, 2(2), 1997.

[5] K. Bharat, B. Chang, M. Henzinger, M. Ruhl Who links to whom: Mining Linkage between Web Sites

[6] R. Kumar, P. Raghavan, S. Rajagopalan, D. Sivakumar, A. Tomkins, E. Upfal. The Web as a graph.

[7] A. Mendelzon, G. Mihaila, and T. Milo. Quering the World Wide Web. *J. of Digital Libraries, 1(1):68-88, 1997.*

[8] S. R. Kumar, P. Raghavan, S. Rajagopalan, and A. Tomkins. Trawling emerging cyber-communities automatically. Poc. 8th WWW Conf, 1999.

[9] S. R. Kumar, P. Raghavan, S. Rajagopalan, and A. Tomkins. Extracting large-scale knowledge bases from the web. *Proc. VLDB, 1999.*

[10] J. M. Kleinberg. Authorative sources in a hyperlinked environment. In Proceedings of the ninth annual ACM-SIAM Symposium on Discrete Algorithms, pages 668-677, 1998.

[11] A. V. Goldberg, E. Tardos, and R. E. Tarjan. Network Flow algorithms. In B. Korte, L. Lovasz, H. J. Promel, and A. Schrijver, editors, *Paths, flows, and VLSI-layout, volume 9 of Algorithms and Combinatorics, , pages 101-164. Springer-Verlag, 1990.*

[12] R. E. Gomory and T. C. Hu. Multi-terminal network flows. J. SIAM, 9:551-570, 1961.

[13] K. Tsioutsiouliklis. *Maximum Flow Techniques for Network Clustering. PhD thesis, Princeton University, Princeton, NJ, June 2002.*

[14] G. W. Flake, R. E. Tarjan, and K. Tsioutsiouliklis. Graph clustering and minimum cut trees. Internet Mathematics, 2003.

[15] A. Benczur. *Counterexamples for directed and node capacitated cut-trees. SIAM J. Conput. 24:3 (1995), pp. 505-510.*

[16] Z. Wu and R. Leahy. An optional graph theoretic approach to data clustering: Theory and its application to image segmentation. IEEE Transaction on Pattern Analysis ans Machine Intelligence. Novermber 1993.

[17] M. Charikar. *Greedy Approximation Algorithms for Finding Dense Components in Graphs, in Proceedings of APPROX, 2000.*

[18] Y. Asahiro, K. Iwama, H. Tamaki and T. Tokuyama. Greedily Finding a Dense Subgraph. Journal of Algorithms, 34(2):203-221, 2000.

Web Communities and Algorithms

Abstract

Communities are those sets of entities in Web that have a strong relationship to each others. They appear in the shape of Web pages or Host pages. It is necessary to examine the link structure of the induced graph of Web in order to infer the communities. Recognizing Web communities provide us with a precious information about what pages are about.

Given a graph $G(V, E)$ with V representing the Web pages and E representing the links among them, the problem is to extract some subsets from G, which are either in the form of hubs and authorities, or dense in edges.

In this B.Sc thesis, we probe the web characterisitcs and models. We explain the traditional methods to identify Web communities. We will see the minimum cut tree method. Then we examine the Charikar's procedure and then we conclude.

Keywords: *Communities, hub, authority, flow, cut*

Web Communities and Algorithms

by

Mahyar Salek

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Bachelor of Science
in
Computer Engineering (Software)

Under supervision of
Prof. Mohammad Ghodsi

August 2005

Computer Engineering Department

Sharif University of Technology

Tehran