دانشکده مهندسی کامپیوتر

پایان نامه کارشناسی ارشد
گرایش نرم افزار

عنوان
تحلیل انتشار در شبکه های اجتماعی مبتنی بر نظریه بازی ها

نگارش
میلاد افتخار

استاد راهنما
دکتر محمد قدسی

تیرماه ۱۳۸۸

به نام خدا

دانشگاه صنعتی شریف

دانشکده مهندسی کامپیوتر

رساله کارشناسی ارشد

عنوان: تحلیل انتشار در شبکه های اجتماعی مبتنی بر نظریه بازی ها

نگارش: میلاد افتخار

کمیته ممتحنین:

استاد راهنما: دکتر محمد قدسی **امضاء.....**

استاد ممتحن داخلی: دکتر علی موقر **امضاء.....**

استاد ممتحن خارجی: دکتر مهران سلیمان فلاح **امضاء.....**

تاریخ:.....

قدردانی

بدین وسیله مشتاقم که صمیمانه از دکتر محمد قدسی تشکر نمایم که تشویق‌ها، راهنمایی‌ها و حمایت ایشان از آغاز پروژه تا سطوح نهایی به من امکان داد تااین رساله را آماده نمایم.

با سپاس فراوان از:
کمیته داوری بخصوص جناب آقای دکتر علی موقر که به حقیر و این نوشته، توجه خاص مبذول داشته و کمال همکاری را در مراحل مختلف با من داشته‌اند.

.

تقدیم به:

پدر، مادر و برادر عزیز و مهربانم که بی هیچ شک و تردیدی بدون دلگرمی و کمک‌های بی شائبه‌شان، نگارش این رساله هرگز امکان پذیر نبود.

.

و سپاس پروردگار یکتا راست که چون رحمت واسعش شامل حالم نمی‌شد، هیچگاه به این درجات علمی دست نمی‌یافتم.

پروردگارا! سپاس بی‌کران تو راست که آدمی را آفریدی و آنچه را نمی‌دانست به او آموختی.

و یقین دارم که:
بنده همان به که ز تقصیر خویش، عذر به درگاه خدای آورد
. ورنه سزاوار خداوندیش، کس نتواند که به جای آورد

تحلیل انتشار در شبکه‌های اجتماعی مبتنی بر نظریه بازی‌ها

چکیده

آدمی از همان اوان پیدایش و شکل‌گیری اولین تمدن‌های بشری برای گذران زندگی روزمره خود به تجارت و خرید و فروش مشغول بوده‌است. سالهای سال است که قواعدی برای تجارت نگاشته شده و افراد برای افزایش سود خود در مبادلات تجاری به روش‌های متعددی روی آورده‌اند. در سالهای اخیر نیز علوم کامپیوتر و پیشرفت تکنولوژی در این عرصه وارد شده و با علم اقتصاد درآمیخته است و بویژه شبکه‌های اجتماعی تحت وب به عنوان ابزاری بسیار قدرتمند این ارتباط را استوارتر نموده است و به عنوان بستری برای پیشبرد پدیده‌ای با نام بازاریابی ویروسی که در آن موج محبوبیت یا اطلاع‌رسانی توسط خود افراد در سطح وسیعی پخش می‌شود، نقش اساسی ایفا می‌کنند. مهمترین مساله‌ای که در این زمینه به آن پرداخته شده است، مساله یافتن تاثیرگذارترین مجموعه k عضوی از نودهای درون یک شبکه است که یک مساله $NP-Complete$ بوده و بنابراین روشهای حریصانه و تقریبی برای حل آن پیشنهاد شده است. ما در این رساله سعی داریم تا با ایجاد گروه‌های منسجم در شبکه به سرعت و دقت پاسخ‌ها و افزایش موج انتشار کمک کنیم. بدین منظور در فصل ۲، روشی ارائه می‌دهیم که افراد درون شبکه را با بهره‌گیری از ایده‌های الگوریتم ROCK و سازگارسازی آنها برای شبکه‌های جهتدار وزندار و با استفاده از اطلاعات گروه‌های خام اولیه، گروه‌بندی کند و مدل فردی شبکه را به یک مدل گروهی تبدیل نماید. در فصل ۳، چگونگی تحلیل انتشار در این مدل گروهی را برای یافتن m گروه تاثیرگذار بررسی می‌کنیم و در انتهای فصل سرعت و سود حاصل از الگوریتم خود را نسبت به الگوریتم متداول Hill-Climbing مقایسه نموده و کارایی الگوریتم خود را نتیجه می‌گیریم. در فصل ۴، به این مساله می‌پردازیم که در دنیای واقع در شبکه‌های اجتماعی، نودی وجود ندارد که از تمامی اطلاعات درون شبکه که برای اجرای الگوریتم لازم است، مطلع باشد. جمع‌آوری این اطلاعات بسیار پرهزینه، زمانگیر و در بسیاری از اوقات غیرممکن است. بنابراین روشی پیشنهاد داده‌ایم تا هنگام نقص اطلاعات، مساله را بدون هزینه اضافی برای جمع‌آوری اطلاعات و بصورت نامتمرکز برای تامین محرمانگی اطلاعات حل کنیم. الگوریتم توزیع شده ما ویژگی مثبت دیگری نیز دارد و آن اینکه به علت مشارکت تمامی نودها در اجرا برای شناسایی k نود تاثیرگذار، سرعت افزایش می‌یابد. بعلاوه چون در الگوریتم‌های توزیع شده ممکن است که نودها برای افزایش سود خود در ارسال و تبادل اطلاعات و محاسبات دروغ بگویند، مکانیزمی طراحی می‌کنیم که راستگویی و درست‌کرداری بازیکنان را تضمین کند.

واژه‌های کلیدی: شبکه اجتماعی، بازاریابی ویروسی، نودهای تاثیرگذار، گروه‌بندی چند عضویتی، طراحی مکانیزم با نقص اطلاعات

فهرستِ مطالب

فهرستِ جداول

فهرستِ اشکال

فصل ۱

مقدمه

امروزه اقتصاد، دانشی است که به چارچوب‌ها و قواعد و اصول تجارت و راه‌کارهای موفقیت می‌پردازد. یکی از این راه‌کارها، انجام تبلیغات برای متقاعد نمودن مشتریان به خرید کالای موردنظر است که روش‌های متفاوتی بدین منظور پیشنهاد و به کار بسته شده است. در حال حاضر با گسترش تکنولوژی و دنیای کامپیوتر، علوم مختلف بهره‌های فراوانی از این علم نوپا در حل مشکلات خود و پیشرفت‌های روزافزون گرفته‌اند و خود نیز در اقدامی متقابل باعث گسترش و وسعت این علم شده‌اند. همکاری علم اقتصاد و علم کامپیوتر نیز نمونه‌ای از این قبیل ارتباطات دوجانبه است که در زمینه‌های متنوعی به پیشبرد هر دو کمک شایانی کرده است. از این جمله می‌توان به کمک‌های علم اقتصاد به کامپیوتر در تخصیص منابع به پردازه‌های متفاوت در محیطی که پردازنده‌ها و محیط‌های حاوی منبع متنوع است اشاره کرد و یا به تاثیر علم کامپیوتر در ایجاد تعادل مشتری–کالا در بازار اشاره نمود.

در سالهای اخیر، گسترش شبکه‌های اجتماعی[1] تحت وب سبب شده است که مجموعه‌های ارزشمندی از دانش و اطلاعات در سطح وسیعی فراهم شود. این عامل باعث شده تا در مقالات جدید شاهد آن باشیم که تلاش روزافزونی برای بکارگیری این مجموعه‌های غنی اطلاعات در تجارت انجام شده است. در حقیقت هدف نویسندگان این مقالات اینست که از این شبکه‌های اجتماعی برای انتشار موج محبوبیت کالای خود بهره گیرند. روشن است که این مفاهیم نه تنها در دنیای تجارت بلکه در زمینه‌های گوناگون دیگر نیز مورد استفاده فراوان است. به عنوان نمونه از ویژگی شبکه‌ها می‌توان در دنیای اطلاع‌رسانی برای مطلع‌کردن جمعیت بزرگی از افراد از اخبار خاصی استفاده کرد یا در دنیای سیاست می‌توان بدین روش سیل عظیمی از افراد موافق را برای یک تصمیم‌گیری واحد فراهم نمود.

گسترش شبکه‌های اجتماعی بر روی اینترنت، به شرکت‌های بزرگ این امکان را داده است که با استفاده از اطلاعات افراد و مخصوصاً ارتباطات میان آنها، استراتژی‌هایی برای افزایش سود شرکت اتخاذ کنند. بدین ترتیب جریان انتقال اطلاعات بر اساس ارتباطات و تاثیر افراد بر یکدیگر

[1] Social Networks

در این شبکه‌های اجتماعی از اهمیت بسزایی برخوردار است. در این ساختار، اطلاعات و یا کالاهای جدید توسط افراد به یکدیگر انتقال داده شده و در کل ساختار پخش می‌شود.

در مسائلی که در این حوزه بررسی می‌شود، شبکه اجتماعی معمولاً به صورت گرافی در نظر گرفته میشود که افراد رئوس این گراف را تشکیل می‌دهند و یال‌های این گراف در حقیقت میزان ارتباط میان افراد را مشخص می‌نماید. هدف نیز پیداکردن روشی است تا بتوانیم از این ساختار بیشترین استفاده را ببریم. به عنوان مثال، محبوبیت یک کالای خاص را در کل شبکه اجتماعی پخش نماییم تا بدین طریق سود افزون‌تری حاصل شود.

در این مسائل معمولاً نودها یک استراتژی قدیمی را گرفته‌اند و سعی بر اینست که بررسی کنیم اکتساب استراتژی جدید در بین این نودها چگونه خواهد بود. در ابتدای کار مجموعه کوچکی از نودها را به عنوان اولین اتخاذکننده‌های استراتژی جدید در نظر می‌گیریم. هر نود با توجه به رفتار نودهای همسایه خود تصمیم می‌گیرد که استراتژی جدید را اخذ نماید یا با همان استراتژی قدیمی ادامه دهد. بدین ترتیب در هر مرحله احتمالاً نودهای بیشتری استراتژی جدید را می‌گیرند و به تدریج این استراتژی در شبکه انتشار می‌یابد. این پدیده را در اصطلاح انتشار نوآوری[۲] گویند.

نکته‌ای که در چنین مسائلی در نظر گرفته می‌شود اینست که با افزایش نودهایی که استراتژی جدید را اتخاذ می‌کنند، میزان علاقه‌مندی نودهای دیگر به‌این استراتژی بیشتر می‌شود. به عنوان مثالی از این گونه مسائل می‌توان مجموعه‌ای از افراد را در نظر گرفت که دو گزینه برای انتخاب یک نرم‌افزار پیغام[۳] پیش رو دارند. مسلماً هر چه که تعداد بیشتری از آشنایان نود a نرم‌افزار دوم را استفاده نمایند، نود a به احتمال بیشتری نرم‌افزار دوم را ترجیح خواهد داد.

۱-۱ تعریف صوری مساله (مدل موریس[۴])

در اینجا می‌خواهیم مدلی از مساله را که در [۱] مطرح شده است، بیان نماییم. همانطور که در قسمت قبل گفتیم، شبکه اجتماعی با یک گراف با نام گراف اجتماعی[۵] مدل می‌شود. گراف فوق، یک گراف $G = (V, E)$ است که V مجموعه رئوس را نشان می‌دهد که نماینده افراد (نودها) در شبکه اجتماعی هستند. مجموعه یالها، E، نیز مجموعه‌ای از زوج‌های (a, b) است که در آن a و b به نحوی در شبکه اجتماعی با هم در ارتباط هستند. وضعیتی را در نظر می‌گیریم که دو استراتژی وجود دارند که نودها توانایی انتخاب یکی از آنها را دارند. در ابتدای کار همه نودها استراتژی اول را اتخاذ کرده‌اند و در ادامه می‌توانند بنا به شرایط استراتژی جدید را به عنوان استراتژی مورد علاقه خود اتخاذ نمایند. در حقیقت دو نوع استراتژی قدیمی و جدید متصورند که معمولاً به ترتیب با A و B

[2] Diffusion of Innovation

[3] Messenger

[4] Morris

[5] Social Graph

نمایش داده می‌شوند. بر روی هر یال (a,b) نیز تمایلی وجود دارد که در آن نودهای a و b علاقه دارند که استراتژی‌های یکسانی را اخذ کنند.

مساله به صورت یک بازی مشارکتی تعریف می‌شود که در حالتی که ما بررسی می‌کنیم، یک عدد ثابت $۰ \leq p \leq ۱$ تعریف می‌شود که برای هر دو همسایه a و b در گراف اجتماعی G، اگر هر دو همسایه استراتژی اول را انتخاب نمایند به هر دو سود p، اگر هر دو استراتژی دوم را انتخاب نمایند به هر دو سود $۱-p$ و اگر استراتژی‌های مختلفی انتخاب کنند به هر دو سود ۰ خواهد رسید.

در چنین مسائلی به این نتیجه می‌رسیم که یک نود a هنگامی استراتژی جدید را ترجیح خواهد داد که نسبت مشخصی از همسایه‌هایش این استراتژی را انتخاب کرده باشند. برای نشان دادن درستی این حقیقت به این صورت عمل می‌کنیم. فرض کنید که تمامی نودها به جز نود دلخواه a استراتژی خود را معین کرده‌اند. فرض کنید که نود a به اندازه n_a همسایه دارد که از این تعداد، $n_a^۱$ همسایه استراتژی اول و بقیه استراتژی دوم را انتخاب نموده‌اند. اگر نود a استراتژی اول را برگزیند، سود او برابر $p * n_a^۱$ خواهد بود. به همین ترتیب اگر استراتژی دوم انتخاب شود، سود $(۱-p) * (n_a - n_a^۱)$ به نود a خواهد رسید. بدیهی است که a تمایل دارد که استراتژی اول را انتخاب کند اگر $p * n_a^۱ > (۱-p) * (n_a - n_a^۱)$ باشد. یعنی استراتژی اول انتخاب خواهد شد اگر $\frac{n_a^۱}{n_a} > ۱-p$. به عبارت دیگر می‌توان گفت که نود دلخواه a استراتژی جدید را هنگامی انتخاب خواهد کرد که به نسبت p از همسایگانش استراتژی جدید را انتخاب کرده باشند.

به وضوح می‌توان به این نتیجه رسید که هنگامیکه p کوچک باشد، استراتژی جدید به سادگی درون شبکه منتشر خواهد شد. در اصطلاح به بزرگترین مقدار p که باعث شود استراتژی جدید در درون شبکه منتشر شود و همه یا بخش بزرگی از نودها استراتژی جدید را اکتساب کنند، مقدار آستانه سرایت[۶] گفته می‌شود.

در این مساله، یک مجموعه اولیه متصور است که فرض می‌شود تمامی اعضای آن استراتژی B را پذیرفته‌اند و سایر نودها دارای استراتژی A هستند. ما با شروع از این مجموعه اولیه و با در نظر گرفتن مقدار p بررسی می‌کنیم که آیا استراتژی B در نهایت در شبکه سرایت خواهد کرد (فراگیر خواهد شد) یا خیر؟ خاصیت سرایت به صورت صوری بدین گونه تعریف شده است.

فرض کنید که مجموعه کوچک S در ابتدای کار استراتژی جدیدی را اخذ کرده است. هر نود در هر لحظه از زمان می‌تواند با توجه به استراتژی همسایگانش، به طور مکرر استراتژی خود را تغییر دهد. برای ساده‌تر شدن مساله، فرض می‌کنیم که زمان به صورت گسسته جلو می‌رود، یعنی هر نود در لحظات زمانی $t = ۱,۲,۳,\cdots$ تصمیم می‌گیرد با توجه به استراتژی همسایه‌هایش در مرحله قبلی، $t-۱$، استراتژی خود را در مرحله t انتخاب کند. $h_p^k(S)$ نشان دهنده مجموعه رئوسی است که پس از گذشت زمان k استراتژی B را اخذ کرده‌اند با این پیش فرض که در ابتدای کار مجموعه اخذ کننده‌های استراتژی B، مجموعه S بوده است و p سود ناشی از اتخاذ استراتژی A توسط دو همسایه است. نود a را تغییریافته می‌نامیم اگر یک عدد صحیح i وجود داشته باشد که به ازای هر $j \geq i$، $a \in h_p^j(S)$ برقرار باشد. در اصطلاح مجموعه S را مسری[۷] گوییم اگر تمام رئوس

[6] Contagion Treshold

[7] contagious

گراف G توسط مجموعه S تغییر یابند.

مثال ۱.۱ فرض کنید که گراف G، از یک مسیر دو طرفه نامتناهی تشکیل شده است و رئوس آن با برچسب‌های $\{\cdots,-۲,-۱,۰,۱,۲,\cdots\}$ نشان داده شده‌اند. بعلاوه فرض می‌کنیم که مطالعه انتشار رفتار جدید B با میزان آستانه $p = ۱/۲$ هدف ماست. مجموعه $\{-۱,۰,۱\}$ را در نظر بگیرید. فرض می‌کنیم که این مجموعه، رفتار جدید B را اتخاذ کرده‌اند. در زمان $t = ۱$، این مجموعه همچنان روی B باقی می‌مانند و نودهای $۲-$ و ۲ نیز، رفتار خود را به B تغییر می‌دهند. در زمان $t = ۲$، مجموعه $\{-۲,-۱,۰,۱,۲\}$ روی B باقی می‌مانند و نودهای $۳-$ و ۳ نیز به B تغییر رفتار می‌دهند. به همین روال در زمان $t = k$، مجموعه $\{-(k+۱),-k,\cdots,-۱,۰,۱,\cdots,k,k+۱\}$ رفتار B را دارند. بنابراین هر نودی توسط مجموعه $\{-۱,۰,۱\}$ از زمانی به بعد به رفتار B تغییر می‌کند و بنابراین می‌توان گفت که این مجموعه یک مجموعه مسری است و میزان آستانه سرایت حداقل برابر ۱/۲ است.

هدف مسائل مطرح در این زمینه اینست که بیشترین مقدار آستانه سرایت را بیابیم که به ازای آن مجموعه‌های مسری موجود باشند و یا به ازای مقادیر مشخص آستانه مجموعه‌های مسری را پیدا نماییم.

مدل ما تابحال به این صورت بود که هر نود با توجه به استراتژی همسایگانش می‌توانست به طور متناوب استراتژی خود را از A به B و از B به A تغییر دهد. اما در بسیاری از مسائلی که ما علاقه‌مندیم مدل نماییم، این ویژگی وجود دارد که یک نود اگر استراتژی خود را از A به B تغییر داد، هرگز به استراتژی A باز نمی‌گردد یعنی تا انتهای کار استراتژی B را نگه می‌دارد. به این رفتار در اصطلاح پیشرو[8] می‌گویند. بنابراین مدل پیشرو که در اکثر مقالات و تحقیقات از جمله این رساله به عنوان مبنا مورد بررسی قرار می‌گیرد به این شکل تغییر می‌کند که هر نود در لحظات زمانی گسسته $t = ۱,۲,۳,\cdots$، نودهای دارای استراتژی A به B تغییر می‌کنند اگر به نسبت p از همسایگانشان دارای استراتژی B باشند و نودهای دارای استراتژی B همچنان روی همان استراتژی خواهند ماند.

۱-۲ مدل‌های مختلف شبکه‌های اجتماعی

برای نزدیک‌شدن هر چه بیشتر مدل‌ها به دنیای واقع، مدل‌های دیگری برای شبکه‌های اجتماعی تعریف شده است که در آنها مقادیر p برای نودهای مختلف و همچنین میزان تاثیرگذاری افراد بر یکدیگر متفاوت است. در ادامه مدل‌های عام‌تر موجود برای شبکه‌های اجتماعی را مطرح می‌کنیم. چنین مدلهایی توسط کمپه[9] و همکارانش در [۲] مطرح شده است. مدلهای مشابهی نیز توسط دادز[10] و واتز[11] در [۳] مطرح شده است.

[8] Progressive

[9] Kempe

[10] Dodds

[11] Watz

۱-۲-۱ مدل با آستانه خطی

در مدل آستانه خطی[۱۲]، گراف G یک گراف جهتدار وزندار است که در آن وزن یال (a,b) برابر $w_{ab} \geq ۰$ است که نشان‌دهنده میزان تاثیرگذاری نود a روی b است. واضح است که بایستی $\sum_{(b \in Neighbor(a))} w_{ba} \leq ۱$.

بعلاوه برای هر نود a یک مقدار آستانه Θ_a متصور است. در این مدل نود a استراتژی B را انتخاب خواهد کرد اگر $\sum_{(b|strategy(b)=B)} w_{ba} \geq \Theta_a$.

۱-۲-۲ مدل آستانه عمومی

در مدل آستانه عمومی[۱۳]، مدل قبلی جامع‌تر شده است. به این صورت که در مدل قبل فرض بر آن بود که تاثیری که از نودهای همسایه یک نود بر آن گذاشته می‌شود برابر جمع وزن‌ها است. اما در این مدل پا را فراتر گذاشته و توابع f_a را برای هر نود تعریف می‌کنیم. ورودی این تابع زیرمجموعه‌ای از همسایگان نود a است که استراتژی B را اتخاذ نموده‌اند و خروجی عددی بین ۰ تا ۱ است. بدین ترتیب نود a استراتژی B را خواهد پذیرفت اگر:

$$f_a(X) \geq \Theta_a$$

باشد که در آن X مجموعه همسایگان نود a است که B را پذیرفته‌اند. در این مسائل فرض می‌کنیم که توابع f، توابعی یکنوا هستند.

تعریف ۱.۱ تابع f را یکنوا[۱۴] نامند، اگر $f(\emptyset) = ۰$ و برای هر دو مجموعه $X \subseteq Y$ داشته باشیم: $f(X) \leq f(Y)$.

این نوع مدل، مدلی بسیار جامع است زیرا هر محدودیتی را می‌توان با اعمال تغییراتی در توابع f ارضا کرد.

۱-۲-۳ مدل آبشاری

در مدل آبشاری[۱۵]، شباهت بیشتری با دنیای واقع دیده می‌شود. در این مدل، هر نود پس از اینکه استراتژی جدید را برگزید، تلاش می‌کند تا این استراتژی را به همسایگانش که استراتژی A را دارند

[12] Linear Threshold Model

[13] General Threshold Model

[14] monotone

[15] Cascade Model

تبلیغ کند. پس از هر تبلیغ به احتمال مشخصی همسایگان استراتژی B را می‌پذیرند و خود شروع به تبلیغ آن می‌کنند و یا تبلیغ به شکست می‌انجامد. احتمال موفقیت هر تبلیغ به دو طرف و همچنین به نودهای پیشین که به این نود تبلیغ کرده‌اند و شکست خورده‌اند، بستگی دارد. به عبارت دقیقتر، برای هر نود a، یک تابع $g_a(b, X)$ تعریف می‌شود که در آن g احتمال این را بیان می‌کند که b استراتژی B را به a تبلیغ کرده و این تبلیغ موفقیت‌آمیز باشد، در حالیکه نودهای درون X در مراحل قبل استراتژی B را به a تبلیغ کرده‌اند. در مقاله [۴]، کمپه و سایرین نشان داده‌اند که مدل آبشاری با مدل آستانه عمومی معادلند.

۱–۳ مجموعه نودهای تاثیرگذار

یکی از مهمترین سوالاتی که در شبکه‌های اجتماعی مطرح شده است، یافتن تاثیرگذارترین مجموعه نودهاست. یکی از مهمترین موارد کاربرد این مساله در بازیابی ویروسی[۱۶] است. شرکت‌های مختلف سعی می‌کنند تا با بهره‌گیری از پدیده رواج تبلیغات کلامی و انتشار موج محبوبیت میان افراد، محصول خود را در بازار با خرج کردن کمترین هزینه برای تبلیغات عرضه کنند. بدین صورت که کالای خود را میان مجموعه تاثیرگذاری از افراد تبلیغ کنند و آنها نیز دوستان و آشنایان خود را متقاعد نمایند و بدین ترتیب محبوبیت کالا در میان افراد رفته رفته افزایش می‌یابد. مساله یافتن تاثیرگذارترین مجموعه نودها در [۵] مطرح شد.

بدین ترتیب مساله بدین صورت تعریف می‌شود که با داشتن یک مقدار ثابت k، مجموعه k–عضوی S را بیابیم که در نهایت با تاثیرگذاری آن مجموعه به نودهای درون شبکه، بیشترین تعداد افراد توسط آن مجموعه، استراتژی نو را اتخاذ نمایند. به عبارت دقیقتر با شروع از یک مجموعه اولیه S –مجموعه حاوی نودهایی است که استراتژی B را در آغاز کار پذیرفته‌اند– اگر $f(S)$ تعداد نودهایی را نشان دهد که در پایان انتشار، استراتژی B را پذیرفته‌اند، ما به دنبال مجموعه k–عضوی S هستیم که بیشترین مقدار تابع f را نتیجه دهد. ثابت شده است که مساله فوق برای همه مدل‌ها و حالات خاصی که پیشتر بیان کردیم یک مساله NP-hard است. بدین ترتیب افراد مختلف سعی نموده‌اند تا الگوریتم‌های تقریبی برای حل این مساله ارائه دهند.

تعریف ۲.۱ تابع f را زیرپیمانه‌ای[۱۷] گویند اگر افزودن یک عضو به مجموعه Y بهبود کمتری نسبت به افزودن همان عضو به زیرمجموعه‌ای از Y حاصل کند. به طور دقیق‌تر تابع f زیرپیمانه‌ای است اگر برای مجموعه‌های X و Y که $X \subset Y$ و هر نود $q \notin Y$ داشته باشیم:

$$f(X \cup \{q\}) - f(X) \geq f(Y \cup \{q\}) - f(Y)$$

[16] Viral Marketing

[17] Submodularity

زیرپیمانه‌ای بودن در حقیقت نوعی از خاصیت بازگشت کاهشی[۱۸] را می‌رساند.

تعریف ۳.۱ خاصیت بازگشت کاهشی بدین معناست که سود حاصل از افزودن اعضا هنگامی‌که مجموعه‌هایی که به آنها اضافه می‌شوند بزرگ می‌شود، کاهش می‌یابد.

قضیه‌ای زیر توسط نمهازر[۱۹] و سایرین در [۶] مطرح شده است که ما در اینجا بدون اثبات آن را عیناً نقل می‌کنیم.

قضیه ۱.۱ اگر S^* مجموعه‌ای k عضوی باشد که بیشترین مقدار تابع f رابه دست می‌دهد و f تابعی زیرپیمانه‌ای و یکنوا باشد، آنگاه مجموعه S که از طریق hill climbing به دست می‌آید (با اجرای الگوریتم برای k مرحله و انتخاب یک نود در هر مرحله که بیشترین افزایش در تابع f را حاصل می‌کند) ویژگی زیر را خواهد داشت:

$$f(S) \geq ۰.۶۳ * f(S^*)$$

در مقالات دیگری سعی شده است تا الگوریتم‌های تقریبی برای مدل آبشاری ارائه دهند. کمپه و همکارانش در مقالات [۲] و [۴] به این نتیجه رسیدند که برای هر مدل آبشاری که در آن $g_a(b, X) = w_{ba}$، تابع تاثیر در مدل معادل، زیرپیمانه‌ای خواهد بود و در قضیه کلی‌تری این نتیجه را بیان کرده‌اند که هر مدل آبشاری که در آن توابع g خاصیت بازگشت کاهشی را دارند، معادل یک مدل آستانه عمومی است که تابع تاثیر آن زیرپیمانه‌ای است.

الگوریتم‌های گوناگونی برای مساله نودهای تاثیرگذار مطرح شده است که می‌توان به الگوریتم ناشیانه[۲۰] انتخاب k نود به صورت تصادفی و یا الگوریتم انتخاب k نود با بالاترین درجه خروجی اشاره کرد. بنا بر اطلاعات ما که با مطالعه حجم زیادی از مقالات به دست آمده است، الگوریتم hill-climbing کاراترین الگوریتمی است که در حال حاضر برای حل مساله افزایش سود در شبکه‌های اجتماعی استفاده می‌شود و مقالاتی که در سال‌های اخیر مطرح شده، به مدل‌های مساله و تغییراتی در مساله پرداخته‌اند و بر روی الگوریتم کاراتری برای حل این مساله تمرکز ندارند. در این رساله ما برآنیم که الگوریتم دیگری که مساله را کاراتر حل نماید، ارائه دهیم.

---

[18] diminishing returns

[19] Nemhauser

[20] naive

۱-۴ مدل‌های پیچیده‌تر

در مقالات اخیر شاهد مدل‌های پیچیده‌تر و مفیدتری هستیم که به دنیای واقع منطبق‌ترند. در مقالات [۵] و [۷]، ریچاردسون[۲۱] و دومینگس[۲۲]، روش جدیدی را ارائه دادند. مدلی که در این مقالات استفاده شده است، بدین ترتیب است که n مشتری در نظر گرفته می‌شوند و برای هر مشتری یک متغیر دودویی X_i متصور است. متغیر X_i برابر ۱ خواهد بود اگر مشتری i کالای تبلیغ‌شده را بخرد و در غیر اینصورت برابر ۰ خواهد بود. مجموعه حاوی متغیرهای X_i را با X نشان می‌دهیم. همچنین همسایه‌های مشتری i با $N_i = \{X_{i,1}, \cdots, X_{i,n_i}\}$ نشان داده می‌شوند. هر کالا نیز با یک سری از صفات به صورت $Y = \{Y_1, \cdots, Y_m\}$ معرفی می‌شود. استراتژی بازاریابی که ما برای مشتریان در نظر می‌گیریم را با M نمایش می‌دهیم که در آن M_i مشخص می‌کند که چه تصمیم بازاریابی برای مشتری i در نظر گرفته شده است. متغیر M_i می‌تواند همانند مقاله [۵] یک متغیر دودویی باشد که نشان می‌دهد که آیا به مشتری i تخفیف داده می‌شود (به او تبلیغ می‌کنیم) یا نه. همچنین این متغیر می‌تواند همانند مقاله [۷] یک مقدار پیوسته باشد که میزان تخفیف یا تبلیغی که به مشتری i می‌شود را مشخص کند. در حقیقت در مقاله [۷]، به جای اینکه مجموعه‌ای از نودها انتخاب کنیم و به آنها تبلیغ نماییم، مشخص می‌کنیم که چه مقدار هزینه تبلیغات روی هر شخص خاص خرج شود.

برای هر X_i خواهیم داشت:

$$P(X_i|X - \{X_i\}, Y, M) = P(X_i|N_i, Y, M) = \beta_i P_\circ(X_i|Y, M_i) + (1 - \beta_i) P_N(X_i|N_i, Y, M)$$

در رابطه بالا، $P_\circ$ مشخص می‌کند که احتمال درونی X_i برای خرید کالا چقدر است. P_N تاثیر همسایگان بر X_i را نمایش می‌دهد و β_i که عدد ثابتی به صورت $0 \leq \beta_i \leq 1$ است، میزان خوداتکایی[۲۳] X_i را نشان می‌دهد. بعلاوه N_i مجموعه همسایگان نود i است. در مقاله [۷] از مدل خطی استفاده شده است و بنابراین

$$P_N(X_i = 1|N_i, Y, M) = \sum_{X_j \in N_i} w_{ji} X_j \qquad (1-1)$$

از آنجا که وضعیت همسایگان مشخص نیست، از تمام وضعیت‌های ممکن برای محاسبات استفاده می‌شود. فرض کنید که $C(N_i)$ مجموعه تمامی حالات ممکن برای وضعیت همسایگان مشتری i باشد و $\hat{N}$ یک تخصیص وضعیت ممکن برای این همسایگان باشد. آنگاه از رابطه ۱-۱ خواهیم داشت:

$$\begin{aligned} P(X_i = 1|Y, M) &= \sum_{\hat{N} \in C(N_i)} P(X_i = 1|\hat{N}, Y, M) P(\hat{N}|Y, M) \\ &= \sum_{\hat{N} \in C(N_i)} \beta_i P_\circ(X_i = 1|Y, M_i) P(\hat{N}|Y, M) \end{aligned}$$

[21] Richardson

[22] Domingos

[23] self-reliant

$$+ \sum_{\hat{N}\in C(N_i)} (۱-\beta_i) \sum_{X_j\in N_i} w_{ji}\hat{N}_j P(\hat{N}|Y,M)$$

که $\hat{N}_j$ مقداری است که توسط $\hat{N}$ به X_j تخصیص یافته است. بنابراین:

$$\begin{aligned} P(X_i=۱|Y,M) &= \beta_i P_۰(X_i=۱|Y,M_i) + (۱-\beta_i) \sum_{X_j\in N_i} \sum_{\hat{N}\in C(N_i),\hat{N}_j=۱} w_{ji} P(\hat{N}|Y,M) \\ &= \beta_i P_۰(X_i=۱|Y,M_i) + (۱-\beta_i) \sum_{X_j\in N_i} w_{ji} P(X_j=۱|Y,M) \end{aligned}$$

بنابراین مشخص است که این احتمالات به صورت بازگشتی به هم مربوطند و می‌توان با شروع از یک تخصیص اولیه معقول به پاسخ نهایی دست یافت. یک تخصیص اولیه معقول می‌تواند احتمال درونی $P_۰$ برای اتخاذ کالا در هر مشتری باشد. هدف ما اینست که استراتژی بازاریابی M را طوری بیابیم که بیشترین سود حاصل شود.

در این مقاله دو مفهوم برای مشتریان تعریف شده است. مفهوم اول، مقدار شبکه‌ای[۲۴] و مفهوم دوم، تاثیر شبکه‌ای[۲۵] است. تفاوت این دو مفهوم در اینست که مقدار شبکه‌ای هر نود، سودی را که از طریق فعال‌شدن آن نود در شبکه حاصل می‌شود، مشخص می‌کند و به تاثیر شبکه‌ای، واکنش نود به بازاریابی و به هزینه و سود مرتبط با استراتژی بازاریابی بستگی دارد. اما تاثیر شبکه‌ای، میزان تاثیری را که یک نود بر شبکه دارد، مشخص می‌کند. تاثیر شبکه‌ای هر نود در این مقاله با Δ_i نمایش داده شده و ثابت شده است که:

$$\Delta_i(Y) = \sum_{j=۱}^{n} w_{ij}\Delta_j(Y) \qquad (۱-۲)$$

و بیان می‌کند که تاثیر شبکه‌ای هر نود برابر است با تاثیری که آن نود بر هر همسایه‌اش دارد ضربدر تاثیر آن همسایه در شبکه. بعلاوه برای آغاز محاسبات فرض می‌شود که در ابتدای کار برای تمامی مشتریان $\Delta_i = ۱$.

برای اینکه بتوان از مقادیر پیوسته به عنوان تخفیف مشتریان استفاده کرد، در مدل از توابع دلخواه مشتق‌پذیر $\alpha(z)$ استفاده شده است که:

$$P_۰(X_i|M_i=z) = \alpha(z)P_۰(X_i|M_i=۰)$$

و ما فرض می‌کنیم که $\alpha(z)$ طوری انتخاب می‌شود که $\alpha(۰)=۱$. در مقاله [۷]، آورده‌اند که انتخاب یک تابع نمایی مجانب‌دار برای $\alpha(z)$ منطقی است زیرا خاصیت بازگشت کاهشی را به خوبی مدل می‌کند. تابعی که نویسندگان مقاله از آن استفاده کرده‌اند، به صورت زیر است:

$$\alpha(z) = \alpha_\infty + (۱-\alpha_\infty)e^{-\lambda z}$$

روشن است که $\alpha(۰)=۱$ و $\alpha(z)\to\alpha_\infty$ هنگامیکه $z\to\infty$.

[24] network value

[25] network effect

به علاوه در این مقاله ادعا شده است که الگوریتم پیشنهادی به صورتی است که با اطلاعات ناقص نیز کار می‌کند زیرا این حقیقت در نظر گرفته شده است که اطلاعات ما از شبکه‌ها و ارتباطات افراد کامل نیست و تکمیل این اطلاعات خود هزینه‌بر است. بنابراین یکی از ویژگی‌های الگوریتم اینست که نسبت به نقص اطلاعات مقاوم باشد. نشان داده شده است که اگر هیچ اطلاعاتی از شبکه و ارتباطات میان افراد در دست نباشد، الگوریتم بازاریابی ویروسی همانند بازاریابی مستقیم[۲۶] جواب می‌دهد و با داشتن حتی حجم کمی از اطلاعات، بازاریابی ویروسی بهتر از بازاریابی مستقیم جواب خواهد داد. بازاریابی مستقیم روشی است که در آن هر مشتری بر مبنای علاقه درونی خود در مورد خرید کالا تصمیم می‌گیرد و ارتباطات میان همسایه‌ای در الگوریتم تاثیر داده نمی‌شود.

بعلاوه برای استخراج اطلاعات اضافی از شبکه، سعی شده که با خرج بودجه ثابت بتوان اطلاعات خوبی جمع‌آوری کرد. بدین منظور الگوریتم جمع‌آوری بدین صورت مطرح شده است که از نودهایی که در شبکه ناقص اطلاعاتی، دارای تاثیر خوبی بوده‌اند، در مورد همسایگانشان پرسش می‌کنیم، بدین دلیل که انتظار می‌رود که چنین نودهایی در شبکه کامل اطلاعاتی نیز دارای تاثیرگذاری بالایی باشند. با یافتن همسایگان این نودها می‌توان دو هدف را دنبال کرد. اول اینکه می‌توان به همسایگانی که تاثیرگذاری بالایی به این نودها دارند، تبلیغ کرد و دوم اینکه می‌توان تاثیرگذاری نودهایی را که در شبکه ناقص دارای اهمیت هستند و انتظار می‌رود که در شبکه کامل نیز تاثیرگذار باشند، بار دیگر با اطلاعات جدید محاسبه نمود. بدین ترتیب از نودهای تاثیرگذار در شبکه ناقص به ترتیب اهمیت پرسش می‌شود تا بودجه تخصیص داده شده، به پایان برسد. نشان داده شده است که این روش نسبت به روشی که در آن مجموعه‌ای از افراد را به صورت تصادفی انتخاب کنیم و از آنها در رابطه با همسایگانشان پرسش کنیم، بهتر جواب می‌دهد.

در مقاله [۸] مساله به این صورت بررسی شده است که هر نود ۳ انتخاب مختلف دارد، یا اینکه استراتژی A را انتخاب نماید، یا استراتژی B و یا هر دو استراتژی A و B را همزمان با صرف هزینه بیشتر انتخاب کند. در صورتیکه نودی هر دو استراتژی A و B را انتخاب نماید آنگاه از ارتباط با تمامی همسایگانش سود خواهد برد. به عنوان مثال فرض کنید که دو نرم‌افزار پیامی A و B موجود باشد. همسایگانی که دارای نوع مشترکی از این نرم‌افزارها هستند، قادرند با هم ارتباط برقرار کنند. و همسایگانی‌که از نرم‌افزارهای مختلف استفاده می‌کنند، امکان ارتباط ندارند. بدین معنا که اگر نودی نرم‌افزار A را انتخاب کند با همسایگانی که B را برگزیده‌اند، نمی‌تواند ارتباط برقرار کند. در حالیکه اگر این نود هر دو نرم‌افزار را همزمان داشته باشد، توانایی ارتباط را هم با همسایگان دارای نرم‌افزار A و هم با همسایگان دارای نرم‌افزار B دارد.

نشان داده شده است که اگر هزینه اخذ دو استراتژی به صورت همزمان خیلی کم باشد و یا خیلی زیاد باشد، استراتژی AB انتخاب نخواهد شد. زیرا در حالتی که هزینه کم باشد، همه تمایل دارند که استراتژی AB را اخذ کنند و در مرحله بعدی همه استراتژی A را انتخاب خواهند کرد و هزینه اتخاذ همزمان دو استراتژی را نخواهند پرداخت. در طرف مقابل اگر هزینه زیاد باشد، نودها تمایلی به انتخاب همزمان دو استراتژی نخواهند داشت. بنابراین یک بازه وجود دارد که اگر هزینه درون آن بازه نباشد استراتژی همزمان گسترش نمی‌یابد.

[26] direct marketing

در مقاله [۹] به مساله کمی عمومی‌تر پرداخته شده است. در این مساله برای هر کالا قیمتی نیز متصور است. هر نود برای خریدن یک کالا علاوه بر تاثیری که از همسایگانش می‌پذیرد به قیمت کالا نیز توجه می‌کند. برای این کار الگوریتم تقریبی ارائه شده است. استراتژی‌ای که در مقاله پیشنهاد شده است تاثیرگذاری بهره‌گیری[۲۷] نامیده می‌شود و اساس آن بدین صورت است که در ابتدا کالا به صورت رایگان به مجموعه‌ای از نودها ارائه می‌شود و سپس با اتخاذ یک استراتژی حریصانه، روی کالاها قیمت‌گذاری می‌شود و از خرید این نودها سود حاصل می‌شود.

۱–۵ ایده‌ها و نوآوری‌های ما

همانطور که تابحال اشاره کردیم، مساله تحت بررسی، چگونگی استفاده حداکثری از دانش موجود در شبکه اجتماعی برای افزایش سود یک شرکت تجاری است. به عبارت دقیق‌تر یک شرکت تولیدی برای اینکه سود حاصل از فروش کالای خود را بیشینه نماید، یک طیف تبلیغات را آغاز می‌کند. بر اساس تبلیغات انجام شده و استفاده از قابلیت انتشار در شبکه، افرادی کالا را خریده و به این ترتیب سودی نصیب شرکت می‌شود. هدف این است که اختلاف مبلغ دریافتی از فروش کالاها و هزینه صرف شده روی تبلیغات بیشینه شود. در بخش‌های پیشین دیدیم که مدل‌های مختلفی برای مساله ارائه شده و کارهای متعددی بر روی آن در چند سال اخیر انجام شده است. در ادامه ما برآنیم که با تکنیک‌هایی به کارایی و بهره‌وری از دانش موجود در شبکه بیفزاییم.

در فصول ۲ و ۳ الگوریتم جدیدی بر مبنای گروه‌بندی اعضای شبکه ارائه می‌دهیم و نشان می‌دهیم که این الگوریتم نتایج بهتری نسبت به $Hill-Climbing$ ارائه می‌دهد. بعلاوه الگوریتم با سرعت بالاتری اجرا می‌شود. در فصل ۴ مساله در حضور اطلاعات ناقص مورد توجه قرار خواهد گرفت. برای حل این مساله مکانیزم توزیع‌شده‌ای ارائه می‌دهیم و راستگویی آن را تضمین می‌کنیم. در فصل ۵ نیز نتیجه‌گیری کرده و ایده‌های نوینی برای کارهای آتی مطرح می‌کنیم.

[27] influence-exploit

فصلِ ۲

الگوریتم انتشار گروه محور: گروه‌بندی

همانطور که پیشتر دیدیم مساله پیداکردن تاثیرگذارترین مجموعه k-عضوی از نودها در یک گراف اجتماعی، یک مساله NP-Complete است. بدین ترتیب ما دنبال روشهایی هستیم تا جوابی را با تقریب خوب و در عوض در زمان اجرای معقول ارائه دهد. در روش اول که بخش عمده رساله را تشکیل خواهد داد، ما سعی می‌کنیم که به نحوی رئوس گراف را در گروه‌های شبیه هم قرار دهیم. هدف این است که این گروه‌ها چنان باشند که رفتار رئوس موجود در آنها تقریباً مشابه باشد. سعی می‌شود تا حد امکان این گروه‌ها منسجم باشند و میان رئوس در گروه‌های مختلف شباهت حداقل باشد. بدین روش می‌توان برای تمامی نودهای درون یک گروه، یک نماینده انتخاب کرد و رفتار انتشار را بر روی نماینده‌های گروه‌ها بررسی کرد. این مدل تا حدی به دنیای واقع نیز نزدیکتر است. زیرا در مدل‌های قبلی فرض بر این بود که تبلیغات روی اشخاص انجام می‌شود، اما در عمل اینچنین نیست. این سازمان‌ها و گروه‌ها هستند که مخاطب تبلیغات واقع می‌شوند. به عنوان مثال، مخاطب تبلیغاتی که درون یک روزنامه دانشگاهی چاپ شده است دانشجویان، اساتید و کارکنان آن دانشگاهند. تبلیغات درون یک روزنامه شهری، افراد یک شهرند و رسانه‌های ملی تبلیغاتی را در بر دارند که به کل یک کشور مربوط می‌شوند. بنابراین سنجیده‌تر اینست که گروه‌ها را هدف تبلیغات قرار دهیم. نکته مثبت دیگر این روش اینست که زمان اجرای الگوریتم را به شدت کاهش می‌دهد. به عنوان مثال فرض کنید که n راس در گراف داریم. همچنین فرض کنید که m گروه درون گراف تشخیص داده شده‌اند. با در نظر گرفتن گروه‌ها به جای رئوس زمان اجرای الگوریتم محاسبه انتشار از مرتبه $O(۲^m)$ خواهد بود. حال اگر $m = \log(n)$ باشد، آنگاه زمان کل اجرا نسبت به n خطی خواهد شد و این پیشرفت بزرگی در این زمینه محسوب خواهد شد، نظر به اینکه اگر گروه‌ها به خوبی تشخیص داده شده باشند، جواب حاصل با دقت بسیار خوبی نزدیک جواب بهینه خواهد بود. ویژگی دیگری که در این روش موجود است اینست که بسیاری از این گروه‌ها از قبل مشخص است. یعنی خود افراد در شبکه‌های مختلف عضو گروه‌های متفاوتی شده‌اند. مثلا در شبکه‌های اجتماعی مختلف مثل Facebook یا Orkut شاهد آن هستیم که دانشجویان دانشکده کامپیوتر عضو گروه دانشکده کامپیوتر دانشگاه شریف، عضو گروه دانشگاه صنعتی شریف و عضو

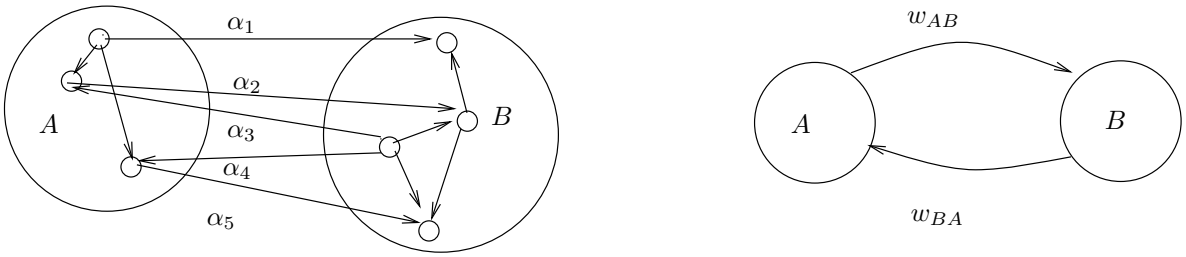
گروه ایرانیان شده‌اند. بدین ترتیب یک ساختار از پیش مشخص شده موجود است که این توانایی را دارد که در تشخیص و شکل‌دهی گروه‌ها کمک شایانی کند.

در این فصل به نحوه گروه‌بندی می‌پردازیم و در فصل بعد، چگونگی انتشار و به دست آوردن گروه‌های تاثیرگذار را مورد بحث قرار می‌دهیم.

۲-۱ مدل گروهی شبکه

حال به چگونگی انجام کار و الگوریتم در این ساختار می‌پردازیم. در ابتدا لازم است که ساختار کلی مساله تغییر یافته را تعریف نماییم. گراف $G = (V, E)$، نشان‌دهنده ساختار نوین شبکه اجتماعی است که در آن V نمایانگر مجموعه گروه‌ها است و E یالهای جهت دار میان گروه‌ها را نشان می‌دهد. یال جهتدار (A, B) نشان می‌دهد که گروه A (در حقیقت اعضای گروه A) تا چه حد بر روی گروه B تاثیرگذار است. وزن روی این یالهای میان گروهی باید به صورتی تعیین شود که هر چه تعداد یالها و وزنشان میان اعضا در دو گروه بیشتر باشد، وزن یال نماینده میان گروهی بیشتر شود.

همچنین به معیاری نیاز داریم تا با استفاده از آن بتوان جمعیت گروه‌ها را نسبت به یکدیگر نشان داد. بدین ترتیب برای هر گروه یک مقیاس جمعیت برابر تعداد نودهای درون آن مشخص می‌نماییم. این مقیاس لازم است زیرا اتخاذ استراتژی نو توسط گروه‌های با جمعیت‌های مختلف، سودهای متفاوتی را نصیب شرکت تولیدی خواهد کرد. مثال ۱.۲ یک مدل گروهی نمونه را نمایش می‌دهد.



(a) (b)

شکلِ ۲-۱: چگونگی مدل‌سازی گروهی یک شبکه اجتماعی. (a). یک شبکه اجتماعی نمونه و نودها و یال‌های میان نودها را نشان می‌دهد. (b). ساختار گروهی متناظر شبکه موجود در (a) را نشان می‌دهد که در آن نودها به ۲ گروه تبدیل شده‌اند و یال‌های میان نودها به یال‌های میان گروهی نظیر شده‌اند.

مثال ۱.۲ شکل ۲-۱، ۲ گروه و اعضا و یالهای میان آنها را نشان می‌دهد. ساختار گروهی حاصل در

قسمت (b) نشان داده شده است.

ما براساس ویژگی‌های ساختار شبکه و گروه‌ها، فرمول زیر را برای محاسبه وزن یال میان گروهی از گروه A به گروه B ارائه داده‌ایم:

$$w_{AB} = \frac{\sum_{i \in A, j \in B} w_{ij} \times w_j^B}{N_B} \qquad (۲-۱)$$

که در آن w_{ij} وزن یال جهتدار از نود i به نود j و w_j^B وزن نود j در گروه B و N_B جمعیت درون گروه B را نشان می‌دهد. وزن نود j در گروه B به طور مستقیم به وزن لینک‌های خروجی از j به سایر اعضای گروه B بستگی دارد. یک روش برای تخمین این وزن به صورت زیر است:

$$w_j^B = \sum_{i \in B} w_{ji} + ۱ \qquad (۲-۲)$$

که در آن ۱ تاثیر نود j بر خودش را نشان می‌دهد.

در مرحله بعد لازم است که استراتژی پذیرش در هر گروه را مشخص کنیم. دو نوع استراتژی مختلف را می‌توانیم بیان کنیم.

استراتژی اول یک استراتژی بر مبنای گسسته‌سازی است که مدل آستانه نیز بر اساس آن بناشده بود. در این روش گروه B یک استراتژی را خواهد پذیرفت اگر:

$$\sum_{A' \in \{\text{گروه‌های فعال همسایه } B\}} w_{A'B} \geq \theta_B \qquad (۲-۳)$$

یک گروه همسایه را فعال می‌خوانیم اگر استراتژی نو را پذیرفته باشد.

استراتژی دوم بر مبنای پیوستگی است. در این استراتژی یک معیار با نام درصد پیشرفت گروهی برای هر گروه تخصیص می‌دهیم. سوالی که ممکن است در این قسمت پیش آید اینست که اصولاً درصد پیشرفت گروهی یعنی چه و به چه دلیل مفید است؟ برای پاسخ دادن به این سوال بایستی به موارد زیر توجه نماییم. اول اینکه در هنگام تبلیغ کردن در یک گروه، عده‌ای از افراد موافق و عده‌ای مخالفند. راضی کردن تمام گروه شاید هزینه زیادی داشته باشد که سود حاصل را تحت تاثیر قرار دهد. دوم اینکه در هنگام تاثیرگذاری گروه‌ها بر یکدیگر، تعدادی از افراد در گروه‌های جدید تاثیر پذیرفته و استراتژی جدید را انتخاب می‌کنند، ولی گروهی از افراد تاثیر نمی‌پذیرند. این درصدها باید قابلیت تغییر داشته باشند. زیرا در طول زمان با افزایش یا کاهش اتخاذ استراتژی نو، درصد پذیرش در گروه‌های مختلف تغییر می‌کند. با استفاده از معیار درصد پیشرفت گروهی، هدف نهایی بیشینه کردن مقدار $\sum N_i \times \beta_i - I$ خواهد بود. N_i معیار جمعیت گروه i، β_i درصد پیشرفت گروهی در گروه i و I هزینه صرف شده برای تبلیغات را مشخص می‌نماید.

در تحلیل انتشار استراتژی نو در شبکه، فرض می‌کنیم که اول کار درصد پیشرفت در تمامی گروه‌ها برابر صفر است یعنی در ابتدا همه استراتژی قدیمی را دنبال می‌کنند. سپس با اجرای کار تبلیغاتی در برخی از گروه‌ها درصد پیشرفت افزایش می‌یابد. در یک روش ناپخته می‌توان فرض کرد که در همه این گروه‌ها، درصد پیشرفت تا ۱۰۰ درصد افزایش می‌یابد. در میانه‌ی کار گروه‌های

با درصد پیشرفت‌های متفاوت خواهیم داشت. در مرحله بعد این گروه‌ها روی گروه‌های دیگر تاثیر می‌گذارند. اگر درصد پیشرفت گروه A از ۰ به β_A تغییر کند و بخواهد به گروه همسایه B با درصد پیشرفت ۰ تبلیغ کند، آنگاه درصد پیشرفت گروهی B برابر خواهد شد با:

$$\beta_B = w_{AB} \times \beta_A$$

۲-۲ تشکیل گروه‌ها

قبلًا بیان نمودیم که در شبکه‌های اجتماعی خود افراد عضو گروه‌هایی شده‌اند. گروه‌های بالا بسیار مفیدند چون توسط خود افراد مشخص شده و دقیق هستند. اما مشکلاتی در این گروه‌ها وجود دارد که استفاده مستقیم از آنها را بدون اعمال یک سری از تغییرات نامفید می‌نماید. اول اینکه بسیاری از این اطلاعات ناقص هستند، یعنی بسیاری از افراد عضویت خود را در گروه‌هایی اعلام نکرده‌اند. دوم اینکه این گروه‌بندی‌ها لزوماً براساس تاثیرگذاری افراد بر یکدیگر نیست و بهتر است بسیاری از این گروه‌ها ادغام، تجزیه و یا حذف شوند. مشکل مهمتر دیگر اینست که گروه‌های بسیاری مهمی در شبکه وجود ندارد. مثلًا مشخص است که اعضای یک خانواده تاثیر بسیار زیادی بر هم دارند، اما هیچ گروه خانوادگی در شبکه‌ها وجود ندارد. مشکل دیگر در رابطه با افرادی است که قبلًا عضو یک سازمان بوده‌اند ولی در حال حاضر عضویت آن سازمان را ندارند، اما بر روی همکاران پیشین خود همچنان تاثیرگذارند. بنابراین گاهی لازم می‌شود که گروه‌هایی در سلسله مراتب مکانی و زمانی مختلف برای یک سازمان تشکیل شود.

بدین ترتیب برای تعیین گروه‌های درون یک شبکه اجتماعی از یک ساختار اولیه خام گروهی که توسط خود افراد مشخص شده است، شروع نموده و با تغییراتی در این گروه‌ها به یک ساختار گروهی مفیدتر خواهیم رسید. گروه‌های نهایی باید دارای عضوهای فعال تاثیرگذار یا تاثیرپذیر باشند. در حقیقت، بصورت طبیعی ممکن است افراد عضو گروههایی شده باشند ولی بر روی افراد آن گروه نه تاثیر بگذارند و نه از آنها تاثیر چشمگیری بپذیرند. نمونه‌های این عضویت‌ها مخصوصاً در گروه‌های برخط[1] بسیار زیاد است. به عنوان مثال می‌توان گروه‌هایی که شامل هواداران یک هنرمند خاص در شبکه اینترنت وجود دارد را در نظر گرفت که ممکن است افراد زیادی عضو این گروه‌ها باشند در حالیکه اصلًا همدیگر را نمی‌شناسند و تاثیری روی هم نمی‌گذارند.

برای تشکیل گروه در شبکه‌هایی از نودها تکنیک‌هایی وجود دارد که در زیر به برخی از آنها اشاره می‌کنیم.[۱۱]

- reachability. روش اول اینست که گروه‌ها به صورتی تشکیل داده شوند که در آن برای هر دو نود درون یک گروه، میزان تاثیرگذاری بین آنها (با واسطه یا بدون واسطه) از یک حدی بیشتر باشد.

[1] online

- k-reachability. روش دوم اینست که گروه‌ها طوری باشند که در آنها بین هر دو نود مسیری وجود داشته باشد که اندازه آن حداکثر k باشد. این روش خود به ۲ نوع تقسیم می‌شود.
  مسیر تنها از یال‌های درون گروه عبور کند.
  مسیر می‌تواند از هر یالی درون شبکه استفاده نماید.

قبل از اینکه در خصوص چگونگی شکل‌گیری گروه‌ها بحث کنیم، لازم است تا تحلیلی در خصوص شبکه و تعداد گروه‌های موجود انجام دهیم. این امر به ما کمک خواهد کرد تا نگرش منطقی‌تری به چگونگی گروه‌بندی داشته باشیم.

فرض می‌کنیم که شبکه مورد بررسی ما در حدود ۱۰۰۰۰۰۰۰ نود دارد. این یک فرض معقولانه است زیرا شبکه‌ای که یک شرکت بزرگ قصد فروش کالا روی آن را دارد، بزرگی‌ای در این حدود خواهد داشت. گروه‌های خیلی کوچک و یا خیلی بزرگ ارزش زیادی برای ما ندارند زیرا یا توانایی انتشار کالا را در حجم بالا ندارند و یا از انسجام قابل قبولی در خود گروه برخوردار نیستند. بنابراین فرض می‌کنیم که گروه‌های مورد نظر ما در حدود ۵۰۰ عضو داشته باشند. از طرف دیگر هر نود می‌تواند در گروه‌های زیادی عضویت داشته باشد ولی بیشتر این گروه‌ها، همان‌گونه که پیشتر اشاره کردیم، گروه‌های مبتنی بر تاثیر نیستند. به عنوان مثال یک فرد می‌تواند عضو گروه‌های حامیان هنرمندان خاص، برخی از چهره‌های سیاسی و ورزشی، از علاقه‌مندان کالاها و نرم‌افزارهای خاص باشد و یا حتی عضو گروه‌هایی مانند شهروندان یک شهر یا کشور مشخص باشد، اما هیچکدام از سایر اعضای گروه را نشناسد. بدیهی است که عضویت در چنین گروه‌هایی نمی‌تواند هدف ما را که در حقیقت انتشار موج محبوبیت کالا توسط تاثیرگذاری افراد گروه بر یکدیگر است برآورده سازد و این گروه‌ها از گروه‌های تاثیری شناخته نشده و در تحلیل‌های ما دخیل نخواهند بود. بنابراین اعضا علیرغم عضویت در گروه‌های فراوان تنها در تعداد محدودی گروه تاثیرپذیری یا تاثیرگذاری دارند و به عبارت دیگر عضویت فعال دارند. فرض می‌کنیم که هر فرد به طور متوسط در ۵ گروه عضویت فعال داشته باشد. با استفاده از اطلاعات بالا به این نتیجه می‌رسیم که برای یک تحلیل قابل قبول در حدود $\frac{۱۰٬۰۰۰٬۰۰۰\times۵}{۵۰۰} = ۱۰۰٬۰۰۰$ گروه نیاز خواهیم داشت.

نکته‌ای که در این خصوص وجود دارد اینست که برای تبدیل زمان NP به زمان خطی، ما نیاز داریم که در کل $log(n)$ گروه داشته باشیم که در آن n تعداد نودهای شبکه را نمایش می‌دهد. به صورت واضح‌تر، اگر بخواهیم به زمانی برابر $T = O(n^k)$ برسیم، بایستی حدود $O(k\log(n))$ گروه داشته باشیم. اما با توجه به تحلیل صورت گرفته برای حل مساله به صورت کارا ۱۰۰٬۰۰۰ گروه قابل تشخیص‌اند. فرض کنید که یک پردازنده $۲GHz$ در اختیار داریم. در اینصورت زمانی در اردر $O(۲^{۱۰۰۰۰۰-۳۱})$ ثانیه نیاز داریم که ابداً قابل قبول نیست. می‌دانیم که زمان قابل قبول در حدود حداکثر چند روز است. بنابراین با توجه به اطلاعاتی که داریم، به این نتیجه می‌رسیم که حداکثر تعداد گروه‌ها بایستی در حدود ۵۰ گروه باشد. زیرا با یک پردازنده $۲GHz$ می‌توان در حدود $۲^{۳۱} = ۲^{۳۰} * ۲ \cong ۱۰^۹ * ۲$ عمل در یک ثانیه انجام داد. بعلاوه هر روز برابر $۲^{۴۷} \cong ۸۶۴۰۰$ ثانیه است. بنابراین در زمانی برابر ۸ روز می‌توان $۲^{۵۰}$ عمل انجام داد. بدیهی است که اگر بخواهیم کل شبکه را به ۵۰ گروه تقسیم نماییم، به گروه‌های بسیار بزرگی خواهیم رسید که ابداً انسجامی در آنها

نمی‌بینیم و از این گروه‌ها برای تحلیل و تبلیغات نمی‌توانیم استفاده نماییم.

با این اوصاف می‌توان نتیجه گرفت که یک tradeoff میان زمان اجرا و دقت الگوریتم وجود دارد. یعنی هرچه تعداد گروه‌ها بیشتر باشد، انسجام گروهی بیشتر شده و دقت بالا می‌رود و همانطور زمان اجرا هم بیشتر شده و سرعت کاهش می‌یابد. از طرف مقابل با کاهش تعداد گروه‌ها، سایز هر گروه افزایش یافته و انسجام گروهی کم می‌شود و بنابراین اگرچه سرعت افزایش می‌یابد ولی دقت الگوریتم به شدت پایین می‌آید. اما اشکال کار اینجاست که نمی‌توان نقطه بهینه‌ای در بین این دو مقدار تعیین کرد که سرعت و دقت قابل قبولی را نتیجه دهد. در حقیقت برای حل مساله ما نیاز داریم که همزمان هم سرعتی بالا و هم دقتی بالا داشته باشیم. سرعتی برابر سرعت اجرای الگوریتم در حضور ۵۰ گروه و دقتی برابر دقت الگوریتم با ۱۰۰٬۰۰۰ گروه.

در زیر بخش ۲-۲-۱ روشی را برای حل این مشکل و رسیدن به هر دو خواسته ارائه داده‌ایم.

۲-۲-۱ گروه‌بندی سلسله مراتبی

هدف ما در این بخش اینست که روشی ارائه دهیم که سرعت و دقت بالا رادر اجرای الگوریتم برای ما به ارمغان بیاورد. به همین علت از روش گروه‌بندی سلسله‌مراتبی بهره می‌گیریم. برای بیان موضوع به صورت جزئی‌تر به این روش عمل می‌کنیم که در ابتدای کار کل شبکه نودها را به $m = O(1)$ گروه تقسیم می‌نماییم. در هر سطح از گروه‌بندی، الگوریتم چند گروه را از میان گروه‌های آن سطح به عنوان گروه‌های مهم انتخاب نموده و آنها را در سطح بعد به گروه‌های کوچکتر تقسیم می‌نماید. این کار تا سطح آخر ادامه می‌یابد تا به تعداد مورد نظر گروه با سایز دلخواه برسیم. گروه‌هایی که در سطح آخر حاصل می‌شوند، گروه‌های مورد نظر ما خواهند بود که با بررسی آنها به جواب نهایی خواهیم رسید. به عنوان مثال فرض کنید که می‌خواهیم به ۱۰ گروه در نهایت تبلیغ کنیم. در ابتدای کار، کل شبکه را به ۵۰ گروه تقسیم می‌کنیم و از میان این گروه‌ها با اجرای الگوریتم ۱۰ گروه تاثیرگذارتر را انتخاب می‌نماییم. در مرحله دوم، هرکدام از ۱۰ گروه انتخابی را به ۵ گروه تقسیم کرده و به ۵۰ گروه می‌رسیم که از میان آنها بایستی ۱۰ گروه تاثیرگذارتر را انتخاب نماییم. این کار را تا آخرین سطح ادامه می‌دهیم. بدین ترتیب اگر در کل m سطح داشته باشیم، آنگاه $T = O(m * ۲۵۰)$ خواهد بود. برای مثال بالا، تعداد سطوح m برابر ۷ است. زیرا:

$$۱۰۰۰۰۰ = ۵^m \Rightarrow m = \log_۵ ۱۰۰۰۰۰ = ۷$$

می‌توان سرعت الگوریتم را به ۲ روش بهبود داد:

- با کم کردن branching factor از ۵ به ۴. بدین ترتیب هر گروه را در سطح پایین‌تر به ۴ گروه تقسیم می‌کنیم.
- انتخاب گروه‌های تاثیرگذار کمتر در سطوح میانی. مثلًا در هر سطح میانی به جای انتخاب ۱۰ گروه تاثیرگذارتر، ۸ گروه را انتخاب کنیم و در مرحله بعد از بین ۴۰ گروه موجود دنبال گروه‌های تاثیرگذار باشیم.

با اجرای هریک از موارد بالا، زمان الگوریتم در هر بخش $T = O(۲^{۴۰})$ خواهد شد که با توجه به اجرای $۲^{۳۱}$ عمل در ثانیه، زمانی کمتر از ۱۰ دقیقه مورد نیاز خواهد بود.

نکته‌ای که در انتها بایستی بدان اشاره کنیم اینست که گروه‌بندی سلسله مراتبی می‌تواند به ۲ صورت دستی و یا خودکار انجام شود. در روش دستی، گروه‌ها به ترتیب بر اساس یک معیار مشخص گروه‌بندی خواهند شد. به عنوان مثال گروه‌بندی می‌تواند براساس معیار جغرافیایی و یا براساس معیارهای کاری باشد. مثلًا در گروه‌بندی جغرافیایی می‌توان در ابتدا گروه‌ها را براساس قاره‌ها تقسیم بندی کرد. آسیا، اقیانوسیه، امریکا، اروپا و آفریقا. سپس براساس کشورها، شهرها و ... . اما مشکل این راه‌کار اینست که برخی از گروه‌ها بعد جغرافیایی ندارند و یا بین‌المللی هستند. در گروه‌بندی‌های دستی بعلاوه نمی‌توان تضمین کرد که گروه‌ها تقریباً هم سایز باشند. بنابراین ما در این رساله گروه‌بندی خودکار را به عنوان روش گروه‌بندی انتخاب می‌نماییم.

نکته‌ای که بایستی بدان توجه کنیم اینست که شبکه متقارن نیست. به عبارت دیگر وزن یال از نود دلخواه u به نود دلخواه v با وزن یال متناظر آن از نود v به u برابر نیست. بعلاوه نمی‌توان معیار فاصله که اساس کار الگوریتم‌های خوشه‌بندی است را میان نودها تعریف نمود. بنابراین نمی‌توان از تکنیک‌های رایج خوشه‌بندی استفاده کرد. بنابراین ما در این رساله برای گروه‌بندی نودها از ایده‌های الگوریتم ROCK استفاده می‌کنیم.

۲-۲-۲ الگوریتم ROCK

الگوریتم ROCK که مخففی است از RObust Clustering using linKs یک الگوریتم خوشه‌بندی سلسله مراتبی است که برای مجموعه‌ای از داده‌ها با مقادیر گسسته غیرعددی طراحی شده است. در این الگوریتم برای اولین بار از مفهوم لینک که در حقیقت همسایه مشترک میان دو نود است، برای اجرای خوشه‌بندی بهره گرفته شد. پیش از طرح این الگوریتم، برای خوشه‌بندی داده‌های غیرعددی نیز همانند داده‌های عددی از توابع فاصله استفاده می‌شد، اما در اکثر مواقع خوشه‌های حاصل از چنین روش‌هایی قابل اعتماد نبود. در این روش‌های سنتی معمولًا از شباهت میان نودها به تنهایی برای ادغام خوشه‌ها استفاده می‌شد و این روش‌ها بسیار خطاپذیر بود. به عنوان مثال به راحتی دو خوشه بسیار متفاوت و بعضاً متضاد با هم ادغام می‌شدند، تنها بدین دلیل که هر گروه یک نود داشته که به یک نود گروه دیگر نزدیک بوده است که در بسیاری از مواقع این نودها، نقاط دورافتاده[۲] بوده‌اند. الگوریتم ROCK با دخالت دادن مفهوم لینک‌ها میان نودها، حالت عمومی‌تری به شرایط لازم برای ادغام خوشه‌ها داده است.

اکنون به جزییات بیشتری از الگوریتم ROCK می‌پردازیم. در ROCK، ۲ نود همسایه‌اند اگر $sim(u,v) \geq \theta$ باشد که θ مقداری آستانه است که مشخص می‌کند دو نود تا چه اندازه‌ای بایستی به هم نزدیک باشند تا به عنوان همسایه در نظر گرفته شوند. همانطور که گفتیم لینک همان همسایه‌های مشترک میان نودهاست که می‌تواند به عنوان معیاری علاوه بر معیار شباهت در افزایش

[2] outlier

دقت به ما کمک کند. هدف از خوشه‌بندی همواره اینست که خوشه‌های نهایی دارای ارتباطات و انسجام بالایی باشند و میان خوشه‌های مختلف کمترین ارتباطات ممکن برقرار باشد. بنابراین ما در الگوریتم ROCK در نظر داریم که مجموع لینک‌های میان جفت نودهای هم گروه را در گروه‌های مختلف بیشینه کنیم و به طور همزمان لینک‌های میان جفت نودها در دو گروه متفاوت را کمینه نماییم. بدین ترتیب بیشینه کردن معیار زیر به عنوان هدف الگوریتم شناخته می‌شود:

$$E_l = \sum_{i=1}^{k} n_i \times \sum_{p_q, p_r \in C_i} \frac{link(p_1, p_r)}{n_i^{1+2f(\theta)}} \qquad (۲-۴)$$

که در آن خوشه C_i دارای n_i عضو است.

چشم انداز کلی الگوریتم ROCK به این صورت است که در ابتدا بخشی از داده‌ها به عنوان داده‌های نمونه انتخاب می‌شود و الگوریتم سلسله مراتبی بر روی آنها با استفاده از مفهوم لینک شروع به خوشه‌بندی می‌کند. سپس کل نودها به خوشه‌هایی که براساس این نودهای نمونه به دست آمده‌اند، متناظر می‌شوند.

خوشه‌بندی روی نقاط نمونه بدین ترتیب اجرا می‌شود که در ابتدای کار هر نود به عنوان یک خوشه در نظر گرفته می‌شود. در مراحل بعدی در هر مرحله از میان تمام جفت خوشه‌ها، دو خوشه‌ای را که بیشترین شباهت را با هم دارند ادغام می‌کنیم. این کار تا زمانیکه به تعداد مورد نظر خوشه‌ها برسیم ادامه می‌یابد. برای یافتن دو خوشه‌ای که در هر مرحله بایستی با هم ادغام شوند، معیاری با نام معیار خوبی[3] به صورت زیر تعریف شده است:

$$goodness(i,j) = \frac{link(c_i, c_j)}{(n_i + n_j)^{1+2f(\theta)} - n_i^{1+2f(\theta)} - n_j^{1+2f(\theta)}} \qquad (۲-۵)$$

بنابراین در هر مرحله جفت خوشه‌هایی که دارای بیشترین مقدار خوبی هستند، با هم ادغام می‌شوند.

در انتها نیز همانطور که اشاره کردیم بایستی تمام نودها به خوشه‌های تشکیل شده، متناظر شوند. براین اساس هر نود به خوشه‌ای اختصاص می‌یابد که دارای بیشینه مقدار $\frac{N_i}{(|L_i|+1)^{f(\theta)}}$ باشد. مقدار N_i، تعداد همسایه‌های نود را در خوشه C_i نشان می‌دهد.

در مقاله [۱۲] نشان داده شده است که پیچیدگی زمانی این الگوریتم برابر $T = O(n^2 + nm_m m_a + n^2 \log(n))$ است که در آن m_m بیشینه مقدار همسایه‌های یک نود است و m_a میانگین تعداد همسایه‌های نودهاست.

لازم به ذکر است که برای اینکه بتوان از الگوریتم ROCK در مساله انتشار نوآوری استفاده کرد، بایستی تغییراتی در آن اعمال کرد. در بخش‌های بعدی به سازگارسازی الگوریتم ROCK برای شبکه‌های اجتماعی خواهیم پرداخت.

---

[3] goodness

۲-۲-۳ گروه‌بندی چند عضویتی با بهره‌گیری از الگوریتم ROCK

همانگونه که پیشتر بیان کردیم، ماتریس وزنی یال‌ها حاصل از شبکه اجتماعی یک ماتریس غیر متقارن است و همچنین معیار تعریف شده‌ای برای فاصله بین نودها وجود ندارد. در چنین ساختاری استفاده از تعداد همسایه‌های مشترک میان نودها برای خوشه‌بندی که در الگوریتم ROCK مطرح شده است، بسیار سودمند خواهد بود. در این روش نودهایی را همسایه می‌نامیم که تاثیرگذاری میان آنها از یک مقدار آستانه θ بیشتر باشد. تاثیرگذاری میان دو نود را برابر مجموع وزن یالهای یکطرفه میان آنها تعریف می‌کنیم.

$$w(u,v) + w(v,u) = Inf(i,j) = v \text{ و } u \text{ تاثیرگذاری میان نودهای} \qquad (۲-۶)$$

در ابتدای کار به علت حجم بالای شبکه و محدودیت فضای حافظه لازم است تا از روش نمونه‌برداری[۴] استفاده نماییم. چگونگی نمونه‌برداری اولیه از شبکه از موارد مورد بحث این رساله نیست و ما همانطور که در [۱۲] اشاره شده از یکی از روش‌های موجود در [۱۳] استفاده می‌نماییم. همانطور که در این مقالات آمده است، اتخاذ تکه‌ای از اطلاعات به عنوان نمونه با اندازه مناسب نه تنها باعث کاهش کیفیت خوشه‌بندی نمی‌شود، بلکه با حذف نودهای دورافتاده[۵] به خوشه‌بندی کمک می‌کند. بعلاوه در مقاله [۱۴] تحلیلی در رابطه با اندازه مناسب نمونه اتخاذ شده برای داشتن یک خوشه‌بندی با کیفیت بالا آمده است.

در ابتدا با استفاده از نودهای نمونه، خوشه‌بندی را با درنظرگرفتن تعداد خوشه‌های مورد نیاز با استفاده از الگوریتم ROCK انجام می‌دهیم. با توجه به این خاصیت که هر نود در چندین گروه می‌تواند عضویت فعال داشته باشد، بایستی الگوریتم ROCK را برای شبکه‌های اجتماعی و مساله خود سازگار کنیم زیرا در الگوریتم‌های خوشه‌بندی از جمله ROCK، روال کار اینگونه است که هر نود در یک گروه قرار می‌گیرد.

بر این اساس نودهای نمونه توسط الگوریتم ROCK معمولی در k گروه، خوشه‌بندی می‌شوند. (یعنی هر نود نمونه در یک خوشه قرار خواهد گرفت) و بدین روش خوشه‌های مورد نظر در این سطح حاصل می‌شوند. قدم بعدی اختصاص هر یک از نودها به این خوشه‌هاست. یعنی نودها بایستی به خوشه‌هایی که اتصال[۶] بالایی با آنها دارند تخصیص داده شوند.

برای این کار از ایده برچسب‌گذاری[۷] داده‌های درون دیسک که در [۱۲] مطرح شده است، بهره می‌گیریم. بدین ترتیب در ابتدا کسری از نودهای موجود در هر خوشه i برای استفاده در عملیات برچسب‌گذاری اخذ می‌شوند. سپس نودهای شبکه از روی دیسک خوانده شده و هر نود به تمامی خوشه‌هایی که با آنها اتصالی بیش از یک مقدار آستانه δ دارد، تخصیص داده می‌شود. لازم

[4] Sampling

[5] outlier

[6] Interconnectivity

[7] Labeling

به ذکر است که نودهای نمونه نیز بایستی بتوانند عضو چندین گروه متفاوت باشند، بنابراین برای ارضای این خاصیت در انتهای کار نودهای نمونه نیز بار دیگر با تمام خوشه‌ها بررسی شده و به تمامی خوشه‌هایی که اتصال بالایی با آنها دارند، متناظر می‌شوند.

۲-۲-۴ سازگارسازی الگوریتم ROCK با مساله

همانطور که پیشتر بیان شده است، الگوریتم ROCK برای خوشه‌بندی مجموعه‌ای از داده با استفاده از مفهوم لینک‌های مشترک است. اما نکته‌ای که در ROCK وجود دارد، اینست که الگوریتم ROCK فرض بر متقارن بودن و دودویی بودن (یال‌های بی‌وزن) ماتریس داده‌ها دارد. به عبارت دقیق‌تر الگوریتم ROCK روی گراف‌های ساده عمل می‌کند. واضح است که در شبکه‌های اجتماعی، ماتریس نامتقارن بوده و یال‌ها وزن دار هستند. بدین ترتیب در شاخه‌های زیر بایستی الگوریتم ROCK را تغییر دهیم تا بتوانیم بر روی شبکه‌های اجتماعی از آن بهره بگیریم.

تعیین همسایه‌ها و لینک ها

گفتیم که از الگوریتم ROCK هنگامی می‌توان استفاده کرد که گراف g حاصل از داده‌ها یک گراف ساده باشد. در اینحالت دو نود همسایه خواهند بود اگر یک یال میان آنها برقرار باشد.

$$Simple\ Graph: \ (u,v) \in g \Leftrightarrow \text{همسایه‌اند } v,u \tag{۲-۷}$$

بعلاوه در ROCK معمولی، مقادیر θ و f طوری تعیین می‌شد که در انتها هر نود در خوشه خود دارای $n_i^{f(\theta)}$ همسایه باشد که n_i تعداد اعضای خوشه iام را نشان می‌دهد.

در گراف جهت دار وزن دار دو نود همسایه‌اند اگر مجموع یال‌های یک طرفه میان آنها از یک میزان آستانه بیشتر باشد. به عبارت دقیق‌تر:

$$Inf(u,v) = w(u,v) + w(v,u) \geq \theta \Leftrightarrow \text{همسایه‌اند } v,u \tag{۲-۸}$$

هدف اینست که مجموع تاثیرگذاری هر نود به همسایه‌های درون خوشه‌اش برابر مقدار زیرین باشد:

$$\sum_{v \in \{Neighbors(u)\ in\ C_i\}} Inf(u,v) = n_i^{f(\theta)} \times \alpha \tag{۲-۹}$$

که در آن α میانگین میزان وزنی مورد نظر هر یال است. در گراف جهت دار وزنی همانند گراف ساده هر نود یک لینک برای دو همسایه‌اش می‌باشد.

معیار خوبی

در الگوریتم رایج ROCK برای گراف‌های ساده، معیار خوبی دو خوشه که برای تعیین بهترین خوشه‌های قابل ادغام مورد استفاده قرار می‌گیرد به صورت زیر است:

$$goodness(i,j) = \frac{link(i,j)}{(n_i+n_j)^{۱+۲f(\theta)} - n_i^{۱+۲f(\theta)} - n_j^{۱+۲f(\theta)}} \qquad (۲-۱۰)$$

بدین ترتیب برای گراف‌های جهت‌دار وزنی، معیار خوبی را به صورت زیر تعریف می‌کنیم:

$$goodness(i,j) = \frac{\text{مجموع تاثیرگذاری لینک‌ها میان اعضای } j,i}{(n_i+n_j)^{۱+۲f(\theta)}\delta' - n_i^{۱+۲f(\theta)}\delta' - n_j^{۱+۲f(\theta)}\delta'} \qquad (۲-۱۱)$$

یک تعبیر مفید در مقاله [۱۲] برای لینک‌ها به شرح روبرو مطرح شده است. هر لینک میان دو نود i و j متناظر است با مسیری با یک نود میانی بین i و j. به عنوان مثال اگر نود k همسایه مشترک نودهای i و j باشد، مسیر i,k,j یک لینک میان i و j محسوب می‌شود. ما از این مفهوم استفاده می‌کنیم تا تاثیرگذاری لینک‌ها و پارامتر δ' را معرفی نماییم. با استفاده از تناظر لینک با مسیر، به راحتی مشخص می‌شود که تاثیرگذاری لینک i,k,j برابر حاصلضرب تاثیرگذاری هر یک از یال‌های مسیر است. به عبارت دیگر:

$$\Omega_{ikj} = Inf(i,k) \times Inf(k,j) \qquad (۲-۱۲)$$

عبارت Ω_{ikj}، نماد تاثیرگذاری لینک i,k,j است.

در گراف‌های ساده که اساس کار الگوریتم رایج ROCK است، می‌دانیم که تصور می‌شود که هر نود متعلق به خوشه i دارای $n_i^{f(\theta)}$ همسایه است. به همین ترتیب هر نود برای هر یک از جفت همسایه‌هایش نقش یک لینک را بازی می‌کند، پس هر نود در $n_i^{f(\theta)} \times n_i^{f(\theta)}$ لینک نقش دارد و بنابراین در یک خوشه با n_i نود، انتظار داریم که $n_i \times n_i^{f(\theta)} \times n_i^{f(\theta)} = n_i^{۱+۲f(\theta)}$ لینک میان اعضا برقرار باشد. به صورت متناظر در گراف‌های جهت‌دار وزنی، گفتیم که انتظار می‌رود که در هر خوشه، مجموع وزن هر نود با همسایه‌هایش درون آن خوشه برابر $n_i^{f(\theta)}\alpha$ است. به روش بالا مشخص است که مجموع تاثیرگذاری لینک‌های درون یک خوشه برابر $n_i \times n_i^{f(\theta)} \times \alpha \times n_i^{f(\theta)} \times \alpha$. بنابراین مجموع مطلوب تاثیرگذاری لینک‌های درون خوشه i برابر $n_i^{۱+۲f(\theta)}\alpha^۲$ است. پس به این نتیجه می‌رسیم که $\delta' = \alpha^۲$. بار دیگر اشاره می‌کنیم که θ میزان آستانه تاثیرگذاری برای پذیرش همسایگی است و α میانگین تاثیرگذاری میان همسایه‌هاست.

برچسب‌گذاری داده‌ها به خوشه‌ها

طبق الگوریتم ROCK، یک نود به خوشه‌ای اختصاص دارد که در آن $\frac{N_i}{(|L_i|+۱)^{f(\theta)}}$ بیشینه باشد. برای گراف‌های جهت‌دار وزنی ابتدا تاثیر میانی هر دو نود را به صورت $Inf(u,v) = w(u,v) + w(v,u)$

تعریف کرده بودیم. اگر این مقدار کوچکتر از میزان آستانه θ باشد، تاثیر میانی برابر $Inf(u,v) = ۰$ خواهد بود. برای این گراف‌ها به این صورت عمل می‌شود که یک نود به یک خوشه C_i تعلق دارد اگر $\frac{\sum_{v \in C_i, v \in u's\ neighbors} Inf(u,v)}{(|L_i|+۱)^{f(\theta)}\alpha}$ بیشینه باشد یا از حدی بیشتر باشد.

۲-۲-۵ استفاده از گروه‌های اولیه در خوشه‌بندی

یکی از امتیازاتی که برای روش گروه‌بندی قائل شدیم، وجود گروه‌های اولیه و اعلام‌شده توسط خود نودها در شبکه بود. گفتیم که در بیشتر شبکه‌های اجتماعی افراد گروه‌هایی را تشکیل داده و خود عضو این گروه‌ها می‌شوند. استفاده صحیح از این اطلاعات سودمند می‌تواند در ایجاد گروه‌های کارا و انتشار هرچه بیشتر موج محبوبیت کالا موثر باشد. اما همانطور که گفتیم این گروه‌های اولیه خود مشکلاتی دارند که نمی‌توان آنها را به عنوان گروه‌های نهایی درنظرگرفت. بنابراین هدف اینست که الگوریتمی برای خوشه‌بندی طراحی کنیم که این گروه‌های اولیه را گرفته و عضویت‌های نامطلوب را از آنها حذف کند، عضویت‌هایی میان نودها و گروه‌ها اضافه کند، گروه‌هایی را حذف و اضافه کند و گروه‌هایی را ادغام یا تجزیه نماید. و در کل در حقیقت در گروه‌بندی خود از اطلاعات و همسایگی‌های میان اعضای چنین گروه‌هایی استفاده بهینه را داشته باشد.

بدین منظور با تغییراتی در الگوریتم سازگار شده ROCK که در بخش ۲-۲-۳ مطرح شد، سعی می‌نماییم تا گروه‌های اولیه را نیز در خوشه‌بندی دخیل کنیم. همانطور که در بخش ۲-۲-۲ گفتیم، برای ادغام خوشه‌ها در مراحل میانی در الگوریتم ROCK، معیاری با نام خوبی به صورت زیر تعریف می‌شود :

$$goodness(i,j) = \frac{link(C_i, C_j)}{(n_i + n_j)^{۱+۲f(\theta)} - n_i^{۱+۲f(\theta)} - n_j^{۱+۲f(\theta)}} \qquad (۲-۱۳)$$

که در آن $f(\theta)$ نشان می‌داد که انتظار داریم که هر نود در خوشه C_i تقریباً $n_i^{f(\theta)}$ همسایه درون C_i داشته باشد. برای ادغام گروه‌بندی‌های اولیه در تصمیم‌گیری معیار جدیدی به صورت زیر تعریف می‌کنیم.

$$fg(i,j) = \frac{BG(i,j)}{n_i * n_j} \qquad (۲-۱۴)$$

و

$$mergeVal(i,j) = (۱ - \alpha) goodness(i,j) + \alpha fg(i,j) \qquad (۲-۱۵)$$

که در آن $BG(i,j)$ تعداد زوج نودهایی از ۲ گروه i و j است که در گروه‌های اولیه هم‌گروه بوده‌اند و $n_i * n_j$ تعداد کل زوج نودها در دو گروه است. بعلاوه α ضریبی است بین ۰ و ۱ ($۰ < \alpha < ۱$) که مشخص می‌کند گروه‌بندی‌های اولیه تا چه حد قابل اعتمادند.

با تعریف این معیار جدید، الگوریتم بدین ترتیب عمل می‌کند که در فاز خوشه‌بندی هر بار دو خوشه‌ای که $mergeVal$ بالاتری دارند با هم ادغام شده و یک خوشه جدید به وجود می‌آورند و این عملیات تکرار می‌شود تا به تعداد خوشه‌های موردنظر برسیم.

همانطور که بالاتر از این اشاره کردیم، تعداد همسایه‌های هر نود در یک خوشه برابر تعداد دوست‌هایش در خوشه است که‌تاثیرگذاری بین آنها بیش از مقدار آستانه θ باشد. در حقیقت $f(\theta)$ در این فرمول‌ها مشخص می‌کند که ما علاقه‌مندیم یک نود در یک خوشه n عضوی، چند همسایه داشته باشد. به عنوان یک مثال می‌توان گفت که در یک گروه ۵۰ عضوی، اگر بخواهیم که هر نود با ۷ نفر همسایه باشد (دوستی با تاثیرگذاری قابل قبول)، آنگاه $f(\theta) = \frac{1}{7}$ خواهد بود.

نکته‌ای که در حین اجرای الگوریتم بایستی بدان توجه کنیم اینست که کار خوشه‌بندی در سلسله مراتب مختلف انجام می‌شود و در هر یک از این سلسله مراتب‌ها، اندازه گروه‌ها متفاوت است. به عنوان مثال در سطوح ابتدایی، گروه‌هایی داریم با اندازه حدود ۱٬۰۰۰٬۰۰۰ در حالیکه در مراحل پایانی، گروه‌هایی با اندازه حدود ۵۰۰ عضو خواهیم داشت. مسلم است که ثابت بودن $f(\theta)$ در تمامی طول اجرای الگوریتم بسیار غیرمنطقی است. برای روشن‌تر شدن بیشتر موضوع، تصور کنید که $f(\theta) = \frac{1}{7}$ باشد. در اینصورت در مراحل اولیه انتظار داریم که هر نود در یک خوشه ۱۰۰۰۰۰۰ عضوی حدود $1000000^{0.5} = 1000$ همسایه داشته باشد و در مراحل پایانی هر نود در یک خوشه ۵۰۰ عضوی در حدود $500^{0.5} = 22$ همسایه داشته باشد. هرچند که داشتن ۲۲ همسایه در خوشه‌های نهایی می‌تواند فرض معقولی باشد، اما بدیهی است که نمی‌توان انتظار داشت یک نود در حدود ۱۰۰۰ همسایه حتی در کل شبکه داشته باشد چه برسد به خوشه‌هایی تشکیل شده در مراحل اولیه. بنابراین واضح است که $f(\theta)$ بایستی در سلسله‌مراتب مختلف اجرای الگوریتم تغییر نماید. دو روش می‌توان برای این تغییر ارائه داد. به عبارت دیگر با دو روش می‌توان $f(\theta)$ را در هر سلسله‌مراتب خوشه‌بندی تغییر داد. در روش اول، می‌توان توابع متفاوتی برای f در هر مرحله تعریف کرد و بدین ترتیب باعث تغییرات در $f(\theta)$ شد. اما در روش دوم که با اساس مساله و مدل ما تطابق بیشتری دارد و ما از آن استفاده می‌کنیم، به این روش عمل می‌شود که در تمامی طول اجرای الگوریتم تابع f ثابت خواهد بود و θ تغییر می‌کند. در حقیقت در مراحل اولیه خوشه‌بندی که نیاز به تشکیل خوشه‌های بزرگ داریم، بایستی θ کوچک انتخاب شود و بدین ترتیب نودهای بیشتری با هم همسایه شده و هم گروه خواهند شد و گروه‌های بزرگتری خواهیم داشت. به مرور در سلسله‌مراتب‌های بعدی، گروه‌ها کوچک‌تر می‌شوند و این موضوع وقتی اتفاق می‌افتد که در تعیین همسایه‌ها سخت‌گیرتر بوده و θ را بزرگتر انتخاب کنیم و بدین ترتیب نودهای کمتری با هم همسایه شده و گروه‌های کوچکتری تشکیل می‌شوند. با این توضیحات به این نتیجه می‌رسیم که در سلسله‌مراتب بالایی مقدار θ کوچک و در سلسله‌مراتب نهایی به مرور θ افزایش می‌یابد.

نکته دیگری که بایستی در خصوص تفاوت ROCK اصلی و الگوریتم مورد استفاده ما در شبکه اجتماعی بدان توجه شود، اهمیت وزن یال‌ها در تعیین همسایه‌ها و عضویت نودها در خوشه‌هاست. گفتیم که دو نود همسایه‌اند اگر مجموع دو یال جهت‌دار میان آنها از θ بیشتر باشد. در الگوریتم سنتی ROCK یال‌ها وزن‌دار نیستند و بنابراین تعداد یال‌ها در محاسبات مهم است، اما استفاده از تعداد یال در شبکه اجتماعی بهینه نیست. یعنی ممکن است نودهایی بسیار مهم وجود

داشته باشند که تعداد یال‌های کم ولی با وزن بالا داشته باشند. این دلیلی است که ما را مجبور کرد به جای استفاده از تعداد از وزن در محاسبات استفاده کنیم. در یک جمع‌بندی، بار دیگر اشاره می‌کنیم که چگونه از وزن‌ها در الگوریتم ROCK در عوض تعداد استفاده کردیم.

۱. در ROCK سنتی هدف این بود که خوشه‌ها طوری تعیین شوند که تعداد همسایه‌های هر نود در یک خوشه حدوداً برابر $n_i^{f(\theta)}$ باشد. در الگوریتم ROCK سازگار شده برای شبکه اجتماعی هدف اینست که مجموع وزن یال‌ها میان یک نود با همسایه‌هایش حدود $n_i^{f(\theta)}\alpha$ باشد که در آن α یک ضریب نرمال‌سازی است که مشخص می‌کند به طور متوسط هر یال چه وزنی داشته باشد.

۲. در الگوریتم ROCK پس از مرحله خوشه‌بندی، تمام نودهای شبکه قرارگرفته در دیسک برچسب‌گذاری می‌شوند. برای برچسب‌گذاری در الگوریتم ROCK سنتی، تعداد همسایه‌های هر نود مورد آزمایش با نقاط نمونه درون هر خوشه بررسی شده و نود به خوشه‌ای که تعداد بیشینه همسایه‌ها را در آن دارد، تخصیص داده می‌شود. در ROCK سازگار شده برای شبکه اجتماعی برای برچسب‌گذاری، اساس مجموع وزن یال‌ها با همسایه‌هاست نه تعداد همسایه‌های درون یک خوشه. بنابراین هر نود مورد بررسی به خوشه‌هایی اختصاص می‌یابد که مجموع وزن یال‌های میان نود و همسایه‌هایش در این خوشه‌ها از حد قابل قبولی بیشتر باشد.

فصلِ ۳

الگوریتم انتشار گروه محور: انتشار

۳-۱ انتشار نوآوری در مدل گروه محور

در بخش ۲-۱ درباره مدل گروهی شبکه، یال‌های میان‌گروهی و وزن متناظر هر یال بحث کردیم. بعلاوه در انتهای بخش درباره استراتژی‌های پذیرش نوآوری، ۲ روش ارائه دادیم. روش اول یک روش گسسته بود که با استراتژی رایج در مدل آستانه خطی و عمومی برای مسئله یافتن k نود تاثیرگذار هماهنگ بود. در آن بخش، قسمتی از نقص‌های روش گسسته برای مدل گروهی را مطرح نموده و روش دیگری تحت عنوان روش پیوسته با تعریف معیاری با نام درصد پیشرفت ارائه دادیم. اما بایستی توجه کنیم که استفاده صرف از روش پیوسته‌سازی، نتایج ناپخته‌ای به ارمغان خواهد آورد. علت اینست که در این روش در عمل فرض می‌شود که هر نود پس از اینکه کوچکترین تاثیری در رابطه با نوآوری جدید گرفت، شروع به تبلیغ به نودهای همسایه خود می‌کند و این نودها نیز بسته به میزان تاثیری که نود تبلیغ‌کننده پذیرفته و وزن یال میانی، تاثیر خواهند پذیرفت. اما در دنیای حقیقی در بسیاری از موارد افراد در تبلیغات خود با شکست مواجه می‌شوند و بعلاوه تا هنگام متقاعد شدن کامل به تبلیغات نمی‌پردازند. بنابراین نیاز است تا برای شبیه‌سازی یک انتشار قابل اعتماد و مطابق با دنیای واقع، روش پخته‌تری برای انتشار معرفی نماییم. در ادامه با ۲ روش، اصلاحات لازم را در نحوه انتشار نوآوری پیاده می‌کنیم.

روش ترکیبی

در روش اول با تغییری در مدل پیوسته، آن را برای منظور خود اصلاح می‌کنیم. برای این کار از ایده‌های روش گسسته در روش پیوسته استفاده می‌نماییم. گفتیم که در روش پیوسته، معیاری با نام درصد پیشرفت محاسبه می‌کنیم که در هر مرحله نشان می‌دهد که چقدر از هر گروه نوآوری را پذیرفته‌اند. سپس برای محاسبه انتشار فرض می‌کردیم که اگر درصد پیشرفت یک گروه α باشد و

وزن میان آن گروه و همسایه‌اش برابر w باشد، درصد پیشرفت گروه همسایه از طریق گروه اول برابر $\beta = \alpha w$ خواهد بود. در روش ترکیبی برای هر گروه یک میزان آستانه پذیرش δ متصور است. بنابراین اگر مقدار درصد پیشرفت αw از این مقدار آستانه کمتر باشد، گروه برای اتخاذ نوآوری قانع نشده و درصد پیشرفت آن برابر ۰ خواهد بود و در این صورت خواهیم داشت:

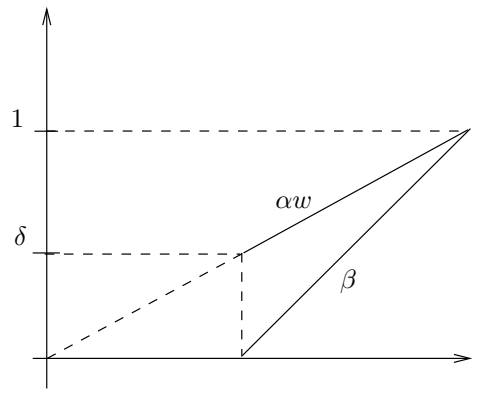
$$\alpha w < \delta \Rightarrow \beta = ۰$$

اگر درصد پیشرفت به میزان آستانه برسد، گروه برای اتخاذ نوآوری متقاعد شده و به تدریج آن را اخذ می‌کند. بدین منظور می‌توان به ۲ روش عمل کرد.

- در حالت اول:

$$\alpha w >= \delta \Rightarrow \beta = \alpha w$$

- در حالت دوم، هنگامیکه αw به میزان آستانه رسید، گروه شروع به اخذ نوآوری می‌کند به این طریق که گروه از مقدار درصد پیشرفت $\beta = ۰$ شروع کرده و هنگامیکه $\alpha w = ۱$ شد، مقدار درصد پیشرفت β هم به ۱ خواهد رسید و این افزایش به صورت خطی یا براساس یک تابع از پیش تعیین شده صورت می‌گیرد. شکل ۳-۱، روش را برای حالت خطی نشان می‌دهد.



شکلِ ۳-۱: چگونگی انتشار نوآوری در روش ترکیبی حالت دوم

هرچند که در بالا ۲ حالت متفاوت برای انتشار در روش ترکیبی مطرح شد، اما تاکید ما بیشتر در این روش بر روی حالت اول است. در حقیقت در حالت اول ویژگی گسستگی بیشتر خودنمایی می‌کند. درصد پیشرفت گروهی پس از رسیدن تبلیغات به مقدار آستانه به یکباره از ۰ به $\beta = \alpha w$ تغییر می‌کند. این حالت گسسته یک عملکرد کاملًا منطقی است، زیرا پس از متقاعد شدن گروه برای اخذ نوآوری، مجموعه‌ای قابل توجه از اعضای گروه همزمان نوآوری را می‌پذیرند و بدین ترتیب درصد پیشرفت به یکباره افزایش می‌یابد و این حقیقتی است که معمولًا در واقعیت رخ

می‌دهد. بنابراین اگر بخواهیم از روش ترکیبی بهره بگیریم، استفاده از حالت اول منطقی تر بوده و نتایج قابل قبول‌تری خواهد داد.

در بخش بعد روش دیگری با استفاده از ایده‌های مدل آبشاری ارائه می‌دهیم که اساس کار این رساله و الگوریتم ارائه شده برای مدل‌سازی گروهی خواهد بود.

روش آبشاری

در این روش سعی می‌کنیم تا با استفاده از ایده‌های مدل آبشاری که در [۲] و [۴] مطرح شده است، روش کارایی برای چگونگی انتشار نوآوری در شبکه‌های اجتماعی ارائه کنیم. بنابراین قصد ما اینست تا به نحوی میزان آستانه δ را از مدل انتشار حذف نماییم و در عوض احتمال فعال‌سازی یال‌ها را در محاسبات دخیل کنیم. بدین روش مدل انتشار را به صورت زیر معرفی می‌کنیم.

همانطور که می‌دانیم در مدل گروهی شبکه اجتماعی، یک سری گروه داریم که یک سری یال‌های یک طرفه میان آنها متصور است و هر یک از این یال‌ها داری وزنی تعیین شده می‌باشد. هرگاه یک گروه فعال می‌شود، می‌تواند توسط هر یک از یال‌های خود، برای فعال‌سازی گروه‌های همسایه اقدام کند. در هر تلاش، گروه تبلیغ‌کننده به یک احتمال مشخص موفق شده و گرنه تلاش او به شکست می‌انجامد. همینطور هرگاه که درصد پیشرفت یک گروه افزایش یافت، می‌تواند بار دیگر برای فعال‌سازی هرچه بیشتر گروه‌های همسایه اقدام نموده و به احتمال مشخصی درصد پیشرفت آنها را افزایش دهد. این اقدام برای فعال‌سازی دوباره، می‌تواند نودهای جدیدی از گروه‌های همسایه را برای اتخاذ استراتژی متقاعد کند و یا تمایل نودهای قبلی را افزایش دهد. روند تبلیغات گروهی تا هنگامی ادامه می‌یابد که در هیچ یک از گروه‌ها اختلاف درصد پیشرفت جدید با درصد پیشرفت پیشین از حدی بیشتر نباشد. به عبارت دیگر فرض کنید که گروه g_i در زمان آخرین اقدامات تبلیغی خود دارای درصد پیشرفتی برابر old_i بوده است و اکنون درصد پیشرفت این گروه بر اثر تاثیرپذیری از گروه‌های همسایه افزایش یافته و به new_i ارتقا یافته است. گروه g_i می‌تواند دوباره به فعال‌سازی و تبلیغ گروه‌های همسایه‌اش بپردازد، اگر:

$$new_i - old_i \geq \gamma$$

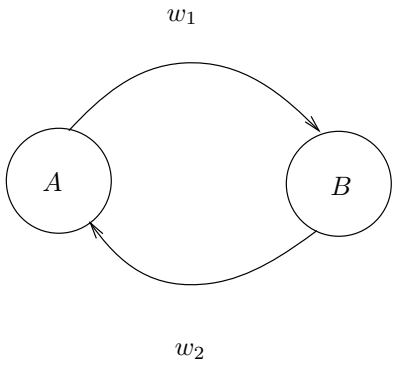
چگونگی تعیین γ خود می‌تواند بر روند انتشار استراتژی نو موثر باشد. در این مقاله γ به صورت زیر تعیین می‌شود:

$$\gamma = \%\frac{۱۰۰}{N_i}$$

منطقی که در این روش تعیین γ موجود است، اینست که مقدار $\%\frac{۱۰۰}{N_i}$ معادل یک نود حقیقی در گروه g_i است و این بدین معناست که منطقاً هنگامی گروه می‌تواند دوباره به فعال سازی بپردازد که حداقل یک نود جدید در گروه فعال شده باشد و در حقیقت نود جدید شروع به تبلیغات به همسایگانش نماید. مثال زیر روش انتشار مورد نظر را روشن‌تر بیان می‌کند.

مثال ۱.۳ فرض کنید که درصد پیشرفت گروه A برابر p_1 بوده و اکنون به p_2 افزایش یافته است. بعلاوه فرض کنید که درصد پیشرفت گروه B برابر q_1 است. همچنین تصور می‌کنیم که یال $A \to B$ در حال حاضر می‌تواند با احتمال Pr به فعال‌سازی گروه B بپردازد. بنابراین درصد پیشرفت گروه B به صورت زیر تغییر خواهد کرد.

$$\left\{ \begin{array}{lll} q_2 = q_1 & : & \text{به احتمال } 1 - Pr \\ q_2 = q_1 + w_1 * (p_2 - p_1) & : & \text{به احتمال } Pr \end{array} \right\} \qquad (۳-۱)$$



شکلِ ۳-۲: یک شبکه کوچک برای مثال انتشار با روش آبشاری

نکته‌ای که بایستی بدان توجه کنیم درخصوص چگونگی تعیین احتمال موفقیت یال‌های میان گروهی در فعال‌سازی‌هاست. در مدل آبشاری مقالات [۲] و [۴] احتمال موفقیت تاثیرگذاری هر نود به نود همسایه علاوه بر دو نود، به تاریخچه فعالیت‌هایی که تابحال برای فعال‌سازی نود جدید صورت گرفته و موفق نشده مرتبط شده است. نکته‌ای که در مدل گروهی وجود دارد، اینست که در هر گروه، اعضای زیادی وجود دارند که در هنگام فعال‌سازی بر اعضای متفاوتی از گروه‌های دیگر تاثیر می‌گذارند. به عبارت دیگر، در دو مرحله مختلف تاثیرگذاری گروه‌ها بر یکدیگر، ممکن است در مرحله دوم نودهای متفاوتی نسبت به مرحله اول در گروه اول فعال شده و بعلاوه بر نودهای متفاوتی در گروه دوم تاثیر بگذارند. یعنی اینگونه نیست که هر گاه گروه A به گروه B تاثیر می‌گذارد، معادل این است که یک عضو ثابت از A، مانند a، به عضو مشخصی از B، مانند b، اثر می‌گذارد. بنابراین هر مرحله از فعال‌سازی هرچند میان گروه‌های یکسانی باشد ولی ممکن است نودهای کاملًا متفاوتی را دخیل نماید. این بدان معناست که هرچند می‌توان در شرایط و مسائل خاص توابعی بر حسب وزن میان گروهی و تاریخچه تلاش‌های صورت گرفته برای فعال‌سازی برای تعیین احتمال موفقیت یال‌ها ارائه داد، اما یک روش کارا در حالت عمومی برای تعیین احتمال موفقیت هر یال در هر مرحله تلاش فعال‌سازی اینست که مستقل از این پارامترها عمل کرده و در حقیقت از توزیع یکنواخت تصادفی برای تعیین احتمالات استفاده نماییم.

شایان ذکر است که در هنگام فعال‌سازی آبشاری میان گروه‌ها، بایستی به ارتباطات درون

گروهی نودها نیز توجه کرد. در انتهای بخش بعد با اصلاح روابط مدل آبشاری، این مهم را نیز در محاسبات دخیل می‌کنیم.

۳-۲ تعیین درصد پیشرفت گروه‌های تاثیرگذار پس از فاز تبلیغات

طبق الگوریتمی که ارائه دادیم، ما در مدل گروهی در ابتدا به دنبال گروه‌های تاثیرگذار می‌گردیم و با تبلیغ به آنها امید داریم تا ارتباطات میان افراد باعث شود که موج محبوبیت استراتژی جدید در کل شبکه گسترش یابد. اما سوال اینست که درصد پیشرفت این گروه‌های تاثیرگذار پس از اجرای فاز تبلیغات چگونه تعیین می‌شود.

اولین حقیقت ملموس اینست که درصد پیشرفت این گروه‌ها بایستی به نحوی به هزینه تبلیغاتی که بر روی آنها صورت گرفته، رابطه داشته باشد. به عبارت دیگر انتظار داریم که هرچه هزینه تبلیغاتی بر روی یک گروه افزایش یابد (البته مسلماً تا یک مقدار محدود)، اعضای گروه به اتخاذ استراتژی مشتاق‌تر شوند و درصد پیشرفت گروهی افزایش یابد.

حقیقت دیگری که در خصوص تعیین درصد پیشرفت بایستی بدان توجه کنیم، میزان جمعیت در هر گروهِ تحت تبلیغات است. مسلم است که در گروه‌های کم‌جمعیت سرانه تبلیغات به هر نفر بیشتر از گروه‌های پرجمعیت بوده (با در نظر گرفتن هزینه تبلیغاتی یکسان) و این امر می‌تواند در متقاعد شدن بیشتر افراد گروه و افزایش درصد پیشرفت موثر باشد.

علاوه بر موارد بالا، نکته دیگری نیز در بالابردن احتمال اتخاذ استراتژی در گروه‌های تحت تبلیغ موثر است و آن میزان ارتباطات اعضای گروه با یکدیگر است. تاثیرگذاری‌های درون گروهی باعث می‌شود که برخی از نودهایی که از طریق تبلیغات گروهی متقاعد به اتخاذ استراتژی نشده‌اند، از طریق تاثیرگذاری همسایه‌های خود که استراتژی را از طریق تبلیغات اتخاذ کرده‌اند، متقاعد شده و به این ترتیب درصد پیشرفت اولیه بالا رود. بنابراین بایستی توجه کنیم که یال‌های درون گروهی می‌تواند تاثیر بسزایی در تعیین درصد پیشرفت اولیه داشته باشد. به همین خاطر معیار جدیدی به نام انسجام گروهی با نماد θ_i بر اساس وزن یال‌های میان گروهی برای هر گروه باید تعریف شود.

براساس یافته‌های بالا متوجه می‌شویم که درصد پیشرفت اولیه گروه‌های تبلیغ براساس تابعی از هزینه، جمعیت گروهی و انسجام گروهی تعریف می‌شود. به عبارت دقیق‌تر، تابع درصد پیشرفت گروهی برای گروه i به صورت $P_{\%i}(C, N_i, \theta_i)$ خواهد بود که در آن C هزینه تبلیغ و N_i میزان جمعیت گروه i را نشان می‌دهد.

هر تابع $P_{\%}$ که به عنوان تابع درصد پیشرفت گروهی تعیین می‌شود به شرط قطعی بودن باید خصوصیت مهم زیرپیمانه‌ای را داشته باشد. یعنی:

$$x < y \Rightarrow P_{\%i}(x+v, N_i, \theta_i) - P_{\%i}(x, N_i, \theta_i) \geq P_{\%i}(y+v, N_i, \theta_i) - P_{\%i}(y, N_i, \theta_i) \qquad (۳-۲)$$

مثال ۲.۳ برای تعیین درصد پیشرفت گروهی، برخی از خصوصیات تابع $P_{\%i}$ و معیار θ_i را مطرح کردیم. در اینجا می‌خواهیم مثالی از این موارد در حالات خاص بیان کنیم. گفتیم که معیار انسجام گروهی براساس وزن یال‌های درون گروهی تعیین می‌شود. بنابراین یک تخمین از این معیار می‌تواند به صورت زیر بیان شود:

$$\theta_i = \frac{\sum_{l,j \in g_i} w_{lj}}{\sum_{j \in g_i} w_{kj}} \qquad (۳-۳)$$

که در آن g_i بیانگر گروه iام است. اگر شبکه به‌گونه‌ای باشد که در آن جمع یال‌های ورودی هر نود برابر ۱ باشد، آنگاه خواهیم داشت:

$$\theta_i = \frac{\sum_{l,j \in g_i} w_{lj}}{N_i} \qquad (۳-۴)$$

در ادامه قصد داریم تا یک مثال هم در مورد تابع درصد پیشرفت بیان کنیم تا مشاهده کنیم که چگونه می‌توان پارامترهای فوق را در تعریف این توابع به کار برد. پیشتر اشاره کردیم که با افزایش C تاحدی می‌توان درصد پیشرفت در یک گروه را افزود. فرض کنید که هزینه استراتژی (کالای) جدید برابر x باشد. در مساله یافتن k نود تاثیرگذار غالباً به این ترتیب تبلیغات را انجام می‌دادیم که کالا به صورت رایگان به نودهای تاثیرگذار داده می‌شد. در اینجا نیز می‌توان مطمئن بود که با خرج کردن هزینه‌ای حداکثر برابر $N_i x$ می‌توان به درصد پیشرفت ۱۰۰% رسید به این طریق که کالا را به صورت رایگان در اختیار تمام اعضای گروه قرار داد. واضح است که به علت وجود تاثیرگذاری‌های میان اعضای گروه که در معیار انسجام مطرح شده است، می‌توان به این درصد پیشرفت با هزینه کمتر از $N_i x$ دست یافت. اما در اینجا به همین تخمین خام اکتفا می‌کنیم. اولین ایده‌ای که در تعیین توابع می‌توان به کار برد، استفاده از روش‌های خطی است. بنابراین اگر به صورت خطی عمل کنیم، میزان درصد پیشرفت گروهی با هزینه C نسبت مستقیم خواهد داشت. با درنظر گرفتن معیار جمعیت در این تابع به این نتیجه می‌رسیم که تابع درصد پیشرفت گروهی به میزان $\frac{C}{N_i x}$ بستگی مستقیم خواهد داشت. به عبارت دیگر اولین تخمین از تابع درصد پیشرفت می‌تواند به صورت زیر باشد:

$$P_{\%i} = \frac{C}{N_i x} \qquad (۳-۵)$$

در گام دوم می‌خواهیم که معیار انسجام را نیز در تابع دخیل کنیم. همانطور که گفتیم با افزایش θ_i، ارزش هزینه خرج‌شده C بیشتر می‌شود و درصد پیشرفت را افزایش می‌دهد. بنابراین θ_i در تابع پیشرفت به طور مستقیم روی هزینه تاثیرگذار است. فرض می‌کنیم که احتمال فعالیت یال‌های درون گروهی از توزیع یکنواخت تصادفی تبعیت کند و انسجام گروهی برابر θ باشد. اگر گروه به اندازه $\alpha\%$ در ابتدا فعال شده باشد، انتظار می‌رود که توسط یال‌های درون گروهی پس از یک مرحله فعالیت یال‌ها، گروه به اندازه $\alpha \times \theta \times \frac{1}{2}$ فعال‌تر شود. به همین ترتیب در فاز دوم به مقدار $\alpha \times \theta \times \frac{1}{2} \times \theta \times \frac{1}{2}$ به فعالیت گروه اضافه شود و به همین ترتیب ادامه یابد. علت اینست که در هر

مرحله نودهای فعال‌شده مرحله قبل دست به فعال‌سازی همسایگان درون گروهی می‌زنند. انسجام گروهی برابر θ است که بدین معناست که به طور متوسط مجموع وزن یال‌های خروجی هر نود به نودهای همان گروه برابر θ است و بعلاوه چون احتمال فعالیت یال‌ها از توزیع یکنواخت تصادفی به دست می‌آید، بنابراین در حالت پایدار، نیمی از این یال‌ها فعال می‌شوند و بنابراین در این مرحله به مقدار حاصلضرب نودهای فعال‌شده مرحله قبل در $\frac{1}{2} \times \theta$، نود تازه فعال شده خواهیم داشت. بدین ترتیب اگر میزان تاثیرپذیری (فعالیت) اولیه برابر α باشد، میزان تاثیرپذیری نهایی برابر خواهد بود با:

$$\alpha + \alpha\theta/2 + \alpha(\theta/2)^2 + \alpha(\theta/2)^3 + ... = \alpha\frac{2}{2-\theta}$$

به همین ترتیب اگر احتمال فعالیت یال‌ها از توزیعی با میانگین $\frac{1}{\mu}$ تبعیت کند، میزان اتخاذ نهایی استراتژی در این گروه بر اساس تاثیرگذاری اولیه α برابر $\alpha\frac{\mu}{\mu-\theta}$ می‌باشد.

بنابراین تخمین دیگر و دقیق‌تری از تابع درصد پیشرفت می‌تواند به صورت زیر باشد:

$$P_{\%i} = min(1, \frac{C(\frac{2}{2-\theta_i})}{N_i x}) \qquad (۳-۶)$$

تابع درصد پیشرفت می‌تواند به صورت تصادفی نیز تعریف شود. به عنوان مثال، استفاده از توزیع‌های نرمال در این رابطه سودمند خواهد بود. مسلم است که در هنگام استفاده از توزیع‌های تصادفی، خاصیت زیرپیمانه‌ای بودن الزاماً برقرار نخواهد بود. اما به علت ماهیت تصادفی اتخاذ استراتژی جدید توسط افراد، در بسیاری از مسائل استفاده از توزیع‌های تصادفی در محاسبه درصد پیشرفت اولیه موثر بوده و انطباق خوبی با واقعیت خواهد داشت. به عنوان مثال، برای انواعی از مسائل تابع درصد پیشرفت می‌تواند به صورت زیر تعریف شود:

$$P_{\%i} = min(1, N(\frac{C(\frac{2}{2-\theta})}{N_i x}, 0.1)) \qquad (۳-۷)$$

که در آن N بیانگر توزیع نرمال است.

توضیح آنکه تابع درصد پیشرفت می‌تواند به تناسب مساله متفاوت باشد و نمونه‌های تخمینی ذکرشده در بالا تنها مثال‌هایی از موارد متعدد قابل قبول هستند.

در انتهای این بخش لازم است که با اطلاعاتی که اینک بدان دست یافتیم، تغییری در انتشار میان گروهی اعمال کنیم. گفتیم که هر گروه در هر بار فعال‌شدن به فعال‌سازی گروه‌های همسایه می‌پردازد و برحسب احتمال‌هایی به موفقیت یا عدم موفقیت می‌رسد. نکته مهمی که پس از آن بایستی بدان توجه کنیم، ارتباط نودهای درون گروه دوم پس از فعال شدن با یکدیگر است. یعنی اگر گروه همسایه توسط گروه اول فعال شد، نودهای فعال شده به تبلیغات درون گروهی مبادرت می‌کنند و میزان فعالیت گروه را افزایش می‌دهند. با محاسباتی که در بالا انجام شد، به نتیجه زیر

می‌رسیم:

$$\left\{ \begin{array}{lll} \text{به احتمال } 1-Pr & : & q_2 = q_1 \\ \text{به احتمال } Pr & : & q_2 = q_1 + w_{AB} \times (p_2 - p_1) \times \frac{\mu}{\mu - \theta} \end{array} \right. \quad (۳-۸)$$

که برای توزیع یکنواخت: $\mu = 2$.

۳-۳ پیچیدگی زمانی

تاکنون در خصوص چگونگی مدل‌سازی و حل مساله برای حاصل شدن سود بیشینه سخن گفتیم. اکنون زمان آن رسیده است که در خصوص زمان اجرای الگوریتم خود بحث کنیم.

۳-۳-۱ زمان اجرای الگوریتم انتشار گروه محور

در حالت کلی روش حل گروه‌بندی از مراحل زیر تشکیل شده است:

۱. نودها در ابتدا گروه‌بندی می‌شوند.

۲. مدل گروهی شبکه شکل داده می‌شود. یعنی وزن یال‌های میان گروهی با استفاده از یال‌های میان نودها در شبکه محاسبه می‌شود.

۳. مساله بر روی گروه‌های نهایی که اندازه‌ای از اردر لگاریتمی اندازه اصلی مساله را دارند حل می‌شود.

حال به محاسبه زمان لازم برای اجرای هریک از مراحل ۳گانه بالا می‌پردازیم. الگوریتم ROCK مساله‌ای با اندازه n را در زمان $T = \theta(n^2 + nm_a m_m + n^2 \log n)$ گروه‌بندی می‌کند. که در آن n اندازه مساله یعنی تعداد نقاط مورد گروه‌بندی است، m_a مقدار متوسط همسایه‌های هر نود در شبکه است و m_m تعداد همسایه‌های نودی در شبکه است که بیشترین تعداد همسایه‌ها را داراست. اما بایستی توجه کنیم که گروه‌بندی استفاده شده در روش ما یک گروه‌بندی سلسله مراتبی است و بعلاوه در آن در عوض استفاده از تمامی نقاط در گروه‌بندی از s نقطه به عنوان نقاط نمونه استفاده شده و سپس بعد از مشخص شدن گروه‌ها، تمامی نقاط را با گروه‌های مشخص شده بررسی کرده و بدین ترتیب کل شبکه گروه‌بندی می‌شود. زمان اجرای کار گروه‌بندی را در سطح اول محاسبه می‌نماییم.

گفتیم برای گروه‌بندی، s نود را در نظر می‌گیریم. بنابراین زمان اجرای الگوریتم در این حالت برابر $T = \theta(s^2 + sm'_a m'_m + s^2 \log s)$ خواهد بود. توجه کنید که m'_m و m'_a بیشینه و میانگین تعداد همسایه‌های نودهای نمونه در میان خودشان است که به مراتب از m_m و m_a کمتر است.

بنابراین زمان اجرای الگوریتم تاکنون برابر $T \simeq \theta(s^2 \log s)$ خواهد بود. فرض کنید که در این مرحله g گروه مشخص شده است. پس ارتباط هر یک از n نود شبکه را بایستی با هر یک از این g گروه بررسی نماییم. برای یافتن میزان ارتباط n نود با g گروه باید وزن یال میان هر یک از این n نود را با تمامی اعضای هر یک از این گروه‌ها بررسی نماییم تا تعداد همسایه‌ها و یا مجموع وزنی میان نودها با همسایه‌هایشان در هر یک از این گروه‌ها مشخص شود. بنابراین زمان لازم برای تکمیل این مرحله کاری حداکثر برابر $\theta(ns)$ خواهد بود (بدترین حالت زمانی است که تمام n نود با تمام s نود نمونه یال داشته باشند و بنابراین به ns محاسبه نیاز باشد). پس در کل زمان گروه‌بندی در سطح اول برابر $T = \theta(s^2 \log s + ns)$ خواهد بود.

نکته‌ای که باید بدان توجه کنیم اینست که زمان اجرای سطح اول به مراتب از سطوح پایین‌تر، بیشتر است. در حقیقت در انتهای هر سطح گروه‌بندی بسیاری از نودها از محاسبات حذف می‌شوند زیرا تنها بخشی از گروه‌های مشخص شده در سطح بعدی باز می‌شوند (به عنوان مثال تنها ۱۰ گروه از میان ۴۰ گروه در سطح بعد باز می‌شوند و بقیه گروه‌ها به عنوان گروه‌های نامهم از محاسبات حذف می‌شوند). بدین ترتیب تعداد نودها از اندازه n کاهش یافته و به همین ترتیب تعداد نقاط نمونه لازم نیز برای گروه‌بندی از s کوچکتر خواهد شد. با این فرض که در کل k سطح گروه‌بندی داریم، زمان کل برابر خواهد بود با $T < \theta(k * (s^2 \log s + ns))$ و چون $k = \theta(1)$، زمان برابر خواهد بود با $T_1 = \theta(s^2 \log s + ns)$. یادآور می‌شویم که در یک شبکه با حدود ۱۰٬۰۰۰٬۰۰۰ نود، طبق محاسبات قبلی صورت گرفته، $k \simeq 7$.

پس از محاسبه زمان اجرای قسمت اول الگوریتم، اینک می‌خواهیم زمان لازم برای تکمیل قسمت دوم را محاسبه کنیم. این کار نیز در سطوح متفاوتی انجام می‌گیرد. ما برای محاسبات خود از سطح اول آغاز می‌کنیم. گفتیم که g گروه داریم. برای ساختن مدل گروهی می‌بایست هر دو گروه را در نظر گرفته و وزن یال میان آن دو را محاسبه نماییم. بدین ترتیب بایستی وزن حدود $g \times (g-1) < g^2$ یال محاسبه شود. برای محاسبه وزن یال از گروه i به گروه j بایستی وزن تمام یال‌ها از اعضای گروه i به اعضای گروه j را با یکدیگر جمع کنیم. بدین ترتیب زمان محاسبه یال از گروه i به گروه j برابر $N_i \times N_j$ خواهد بود. بنابراین در مجموع زمان محاسبه کلیه یال‌ها و در نتیجه زمان لازم برای محاسبه مرحله دوم در سطح اول سلسله مراتب برابر خواهد بود با:

$$N_1 \times N_2 + N_2 \times N_1 + N_1 \times N_3 + N_3 \times N_1 + \cdots + N_g \times N_{g-1}$$

اگر میانگین جمعیت گروه‌ها را با N_g نمایش دهیم، آنگاه زمان به صورت تقریبی برابر خواهد بود با $g^2 \times N_g^2$. در سطح اول به علت بزرگ بودن گروه‌ها، نودها غالباً در بیش از یک گروه عضویت ندارند و بنابراین $N_g = n/g$. بدین ترتیب زمان اجرای این مرحله برابر n^2 خواهد بود. در سطوح پایین‌تر جمعیت گروه‌ها کاهش می‌یابد، زیرا در هر سطح با انتخاب تعدادی از گروه‌ها از میان تمامی گروه‌های موجود (به عنوان مثال ۱۰ گروه از ۴۰ گروه) تعداد نقاط تحت بررسی کاهش می‌یابد (در حدود ۱/۴ می‌شود). اما به همان اندازه، عضویت نودها در گروه‌های متفاوت افزایش نخواهند یافت. برای روشن‌تر شدن موضوع دقت کنید که طبق بررسی‌هایی که در اواسط این فصل صورت دادیم، به این نکته رسیدیم که نودها به تدریج در سطوح مختلف عضو گروه‌های بیشتری می‌شوند

تا در آخرین سطح، هر نود به طور متوسط عضو فعال حدود ۵ گروه خواهد شد. در حالیکه اگر در هر سطح حدود ۱/۴ گروه‌ها و در نتیجه ۱/۴ نودها انتخاب شوند، درنهایت در سطح ۷ حدود $n/۴^۶$ نود مهم برای گروه‌بندی باقی خواهند ماند. به عبارت دیگر اندازه شبکه برای ساختن مدل گروهی و محاسبه وزن‌ها حدود $\frac{۱}{۴۰۹۶}$ شده ولی هر نود حداکثر در ۵ گروه عضو خواهد بود. این نشان می‌دهد که زمان اجرای قسمت دوم در سطوح پایین‌تر به مراتب کمتر از سطح اول خواهد بود و بنابراین زمان کل اجرای فاز دوم کوچکتر از $k \times n^۲$ و برابر $T_۲ = \theta(n^۲)$ خواهد بود.

در فاز سوم الگوریتم یک شبکه متشکل از $log(n)$ گروه داریم که میان هر دو گروه دو یال جهتدار برقرار است . هدف یافتن h گروه تاثیرگذار است که تبلیغات اولیه به آنها باعث بیشترین انتشار در شبکه شود. بزرگترین ویژگی تکنیک گروه‌بندی این است که اندازه مساله از سایز n به $log(n)$ کاهش یافته است و مساله در این سایز کوچک به راحتی قابل تجزیه تحلیل می‌باشد. بدین ترتیب می‌توان برای تمام ترکیب‌های h عضوی از گروه‌های موجود، سود حاصل را محاسبه نمود. فرض کنیم که در کل g گروه تشخیص داده شده باشند. تعداد حالات مورد بررسی برابر $\binom{g}{h}$ است. در هر حالت هر گروه پس از فعال شدن، اقدام به فعال‌سازی همسایگان خود از طریق یال‌های خروجی‌اش می‌کند و با یک احتمالی موفق می‌شود. به همین ترتیب گروه‌ها به محض فعال شدن به تبلیغات خود برای همسایگان ادامه می‌دهند. نکته‌ای که بایستی مورد توجه قرار گیرد اینست که هر گروه حداکثر می‌تواند N_i بار به فعال‌سازی بپردازد، هرچند تعداد واقعی فعال‌سازی‌ها به مراتب کمتر است. این حقیقت بدین علت اتفاق می‌افتد که گروه‌ها هنگامی به تبلیغات می‌پردازند که $new_i - old_i \geq \%\frac{۱۰۰}{N_i}$ باشد و نتیجه می‌گیریم که حداکثر $N_i = ۱۰۰ \div \frac{۱۰۰}{N_i}$ بار این امر روی می‌دهد. در هربار تبلیغ نیز هر نود به $g - ۱$ گروه دیگر تبلیغات خواهد کرد. بنابراین اگر در بدترین حالت فرض کنیم که تمامی نودها N_i بار فعال می‌شوند، آنگاه زمان اجرای این فاز برابر خواهد بود با:

$$T_۳ = \binom{g}{h} \times N_a \times g \times (g - ۱)$$

که در آن N_a متوسط تعداد اعضای گروه‌هاست.

در روش آبشاری به این علت که یال‌های میان گروهی بر اساس احتمال‌های مشخصی فعال می‌شوند، زمان اجرای این فاز کاهش می‌یابد زیرا گروه‌های کمتری در مراحل کمی فعال می‌شوند و بعلاوه به علت احتمالی عمل کردن ممکن است کل انتشار در اواسط کار متوقف شود.

بنابراین زمان اجرای الگوریتم در کل برابر است با:

$$T_{total} = O(s^۲ \log s + ns + n^۲ + \binom{g}{h} N_a g(g - ۱)) \qquad (۳-۹)$$

می‌دانیم که $g = \theta(\log(n))$ و همچنین:

$$n = ۲^{\log(n)} = \binom{\log(n)}{۰} + \binom{\log(n)}{۱} + \binom{\log(n)}{۲} + \cdots + \binom{\log(n)}{h} + \cdots + \binom{\log(n)}{\log(n)}$$

بنابراین:

$$T_۳ < n \times (\log(n))^۲ \times N_a << n^۲$$

چون به علت کم بودن نقاط نمونه، $s\sqrt{\log s} << n$ خواهیم داشت:

$$T_{total} = \theta(n^۲) \qquad (۳-۱۰)$$

واضح است که به روش فوق با افزودن کمی تقریب و گروه‌بندی شبکه اجتماعی توانستیم مساله NP-Complete مطرح شده را با الگوریتمی با زمان $\theta(n^۲)$ حل کنیم و این تکنیک موفقیت بزرگی در زمینه افزایش سود در شبکه‌های اجتماعی رقم خواهد زد.
در زیربخش ۳-۳-۳ نشان خواهیم داد که با اعمال تغییرات جزیی در الگوریتم برای گراف‌های پراکنده، که معمولاً شبکه‌های اجتماعی نیز از این نوعند، می‌توان زمان اجرا را کاهش داد و به سرعت بالاتری دست یافت.

۳-۳-۲ مقایسه زمان انتشار گروه محور با HC

همانطور که در بخش ۳-۱ اشاره کردیم، الگوریتمی که برای یافتن نودهای تاثیرگذار در شبکه‌های اجتماعی در مقالات [۶]، [۲] و [۴] برای مدل‌های آستانه و آبشاری مطرح شده است بر مبنای روش hill-climbing می‌باشد. در حقیقت همانطور که پیشتر گفتیم، روش hill-climbing تنها روشی است که در حال حاضر برای حل این مساله در مدل‌های آستانه و آبشاری استفاده می‌شود. بنابراین بسیار مفید می‌دانیم که زمان الگوریتم خود را با الگوریتم hill-climbing در شبکه‌های اجتماعی مقایسه نماییم.
همانطور که می‌دانیم، در روش hill-climbing برای یافتن k نود تاثیرگذار، در k مرحله کار جستجو را بدین ترتیب انجام می‌دهیم که در ابتدای مرحله iام یک مجموعه $(i-۱)$تایی از نودهای تاثیرگذار در اختیار داریم. در مرحله iام از میان $n-i+۱$ نود باقیمانده به دنبال نودی می‌گردیم که با مشارکت مجموعه $i-۱$تایی حاصل شده از مراحل قبلی بیشترین سود را نصیب سیستم کند. به عبارت دقیق‌تر اگر مجموعه $i-۱$تایی حاصل شده از $i-۱$ مرحله قبلی را S بنامیم و مجموعه دربرگیرنده تمامی نودهای شبکه را A بنامیم، نود u که در شرایط زیر صدق کند، به عنوان نود iام در انتهای مرحله i به مجموعه S اضافه می‌شود.

$$f(S \cup \{u\}) \geq f(S \cup \{u'\}) \qquad \forall u' \in A - S$$

اکنون می‌خواهیم زمان اجرای چنین الگوریتمی را محاسبه کنیم. طبق توضیحات اخیر در هر مرحله از الگوریتم برای هر یک از نودها بایستی میزان انتشار را در شبکه محاسبه کنیم. در مرحله اول n نود داریم که بایستی تحت بررسی قرار گیرند. زمانی که برای بررسی انتشار هر نود لازم است برابر تعداد یال‌های خروجی نودهای فعال است. فرض کنیم زمان متوسط برای محاسبه سود حاصل از انتشار برای هر نود در مرحله اول برابر $t_۱$ باشد، بنابراین زمان مرحله اول برابر $n * t_۱$ خواهد بود.

در ابتدای مرحله دوم مجموعه S یک عضو دربردارد که نود بهینه مرحله اول است. در این مرحله ما به دنبال یافتن عضو دوم مجموعه S هستیم. بنابراین برای هر نود u از این $n-1$ نود باقیمانده بایستی میزان انتشار مجموعه $S \cup \{u\}$ را محاسبه نماییم. فرض کنید متوسط زمان محاسبه میزان انتشار برابر t_2 باشد. که t_2 برابر تعداد یال‌های خروجی نودهای فعال خواهد بود. به همین ترتیب در مرحله kام بایستی نود بهینه از میان $n-k+1$ نود موجود را بیابیم. فرض می‌کنیم که t_k نیز زمان متوسط محاسبه میزان انتشار این مجموعه kتایی باشد. بنابراین زمان اجرای الگوریتم برابر خواهد بود با:

$$T = n \times t_1 + (n-1) \times t_2 + \cdots + (n-k+1) \times t_k \qquad (۳-۱۱)$$

اما می‌دانیم که $t_k \to E$. به این معنا که در مراحل پایانی توقع داریم که موج محبوبیت استراتژی نوین بخش بزرگی از یال‌های شبکه را دربرگیرد. اگر $E = O(n^2)$ باشد، زمان اجرای الگوریتم برابر $T = O(n^3)$ خواهد شد. بدین ترتیب نتیجه می‌گیریم که از نظر زمانی نیز الگوریتم گروه‌بندی کاراتر از الگوریتم hill-climbing عمل خواهد کرد.

شایان ذکر است که در بسیاری از شبکه‌ها، گراف پراکنده[1] است. به عبارت دیگر تعداد آشنایان هر نود یک مقدار ثابت است. به عنوان مثال در بسیاری از شبکه‌های اجتماعی این حقیقت وجود دارد که هر نود با تعداد محدودی از نودها، مثلاً ۱۰۰ نود به طور متوسط، آشنایی دارد. فرض می‌کنیم که تعداد یال‌های خروجی هر نود به طور متوسط برابر α باشد. بنابراین تعداد یال‌های کل گراف برابر خواهد بود با $E = \theta(n\alpha)$. در این صورت واضح است که الگوریتم hill-climbing زمان اجرایی برابر $T_{HC} = \theta(n \times n\alpha) = \theta(n^2\alpha)$ خواهد داشت. که اگر $\alpha = O(1)$ باشد،آنگاه زمان اجرای الگوریتم hill-climbing برابر $\theta(n^2)$ خواهد شد.

برای کارایی بیشتر الگوریتم گروه‌بندی ارائه‌شده در حضور گراف‌های پراکنده، آن را به صورت زیر سازگار می‌کنیم.

۳-۳-۳ انتشار گروه محور و شبکه‌های اجتماعی پراکنده

همانطور که گفتیم در شبکه‌های اجتماعی پراکنده فرض می‌کنیم که تعداد یال‌های خروجی هر نود در گراف به طور متوسط برابر α باشد که α می‌تواند $O(1)$ باشد. هنگامیکه گراف متناظر شبکه اجتماعی تاحدی کامل است یعنی $E = \theta(n^2)$، برای محاسبه وزن‌ها عاقلانه‌ترین کار اینست که برای هر عضو گروه اول و هر عضو گروه دوم، وزن یال میانی را درنظر گرفته (در صورتیکه یالی وجود نداشته باشد، وزن میان این دو نود برابر صفر خواهد بود) و آنها را با هم جمع کنیم. این روش عملکرد برای محاسبه تمام یال‌های میان‌گروهی زمانی برابر $\theta(n^2)$ نیاز داشت. اما زمانیکه $E = \theta(n\alpha)$ است، ما از ماتریس مجاورت نودها برای محاسبات یال‌های میان‌گروهی استفاده می‌کنیم.

فرض کنید که ما می‌خواهیم وزن یال میان دو گروه شناسایی شده A و B را محاسبه نماییم. در فاز اول مشخص می‌شود که هر نود در چه گروه‌هایی عضویت دارد. بنابراینکه گراف ما پر است

[1] sparse

یا پراکنده، چگونگی ذخیره خروجی این فاز متفاوت است. یعنی در حالت اول که گراف پر است ما نیاز داریم که هر یک از گروه‌های شناسایی شده یک لیست پیوندی[۲] برای خود ذخیره کنند و هر نودی که عضو این گروه‌ها تشخیص داده می‌شود، در این لیست‌ها اضافه شود. نکته‌ای که توضیح آن در این قسمت شاید ضروری به نظر برسد، اینست که در مواردی که محدودیت حافظه‌ای داریم و اندازه شبکه بزرگ است، ممکن است نتوانیم لیست پیوندی کامل تمامی گروه‌ها را در حافظه اصلی تا انتهای فاز نگاه داریم. در چنین مواردی می‌توان هر از چند گاهی لیست‌های پیوندی گروه‌ها را در حافظه کمکی ذخیره کرده و با تشکیل لیست‌های پیوندی جدید و خالی برای همه گروه‌ها، سایر نودها را گروه‌بندی کرده و آنها را نیز در زمان‌های بعدی به لیست‌های ذخیره‌شده در حافظه کمکی بیفزاییم. اما در حالت دوم که گراف تقریباً خالی و پراکنده است، برای کارایی الگوریتم به نحو زیر عمل می‌کنیم.

در این حالت در حین اجرای فاز دوم هر نود تحت بررسی برای خود یک لیست پیوندی نگاه می‌دارد و گروه‌های میزبان خود را در آن لیست پیوندی ذخیره می‌کند. بدیهی است که چون تعداد گروه‌هایی که هر نود در آن عضویت دارد محدود و ثابت است، اندازه هر یک از این لیست‌های پیوندی $O(۱)$ خواهد بود. بنابراین خروجی فاز اول در حالت دوم علاوه بر لیست‌های پیوندی گروه‌ها، لیست‌های پیوندی هر یک از نودها خواهد بود که مشخص می‌کند هر نود عضو چه گروه‌های است.

در فاز دوم برای محاسبه وزن یال AB، هر نود u عضو گروه A را گرفته و تمام یال‌های خروجی‌این نود را که به طور متوسط برابر α یال است، با استفاده از لیست مجاورت u در نظر می‌گیریم. سپس در لیست پیوندی نودهای سر دیگر هر یک از این یال‌ها بررسی می‌کنیم که نود عضو گروه B هست یا نه؟ اگر نود همسایه عضو B بود، وزن میان این نود با نود u را با حاصلی که تابحال به دست آمده جمع می‌کنیم. بدین ترتیب زمان لازم برای محاسبه وزن یال AB برابر خواهد بود با $\theta(n_A \times \alpha)$ و زمان کل اجرای فاز دوم برابر است با:

$$T_{grouping} = \theta(\alpha(n_A + n_B + ... + n_Z))$$

که در آن $A, B, \cdots, Z$ گروه‌های تشخیص داده شده در فاز گروه‌بندی هستند. می‌دانیم که $n_A + n_B + \cdots + n_Z = \theta(n)$ است چه نودها فقط عضو یک گروه باشند چه در گروه‌های متفاوتی عضویت داشته باشند. مثلًا در حالت نهایی نیز که فرض کردیم نودها به صورت متوسط عضو ۵ گروه هستند، مقدار $n_A + n_B + \cdots + n_C$ برابر $۵n = \theta(n)$ خواهد بود. بدین ترتیب زمان لازم برای اجرای فاز ۲ در این حالت برابر $\theta(n\alpha)$ است که اگر α را $O(۱)$ در نظر بگیریم، زمان فاز دوم برابر $\theta(n)$ می‌شود. بنابراین زمان کل اجرای الگوریتم گروه‌بندی در این حالت برابر خواهد بود با:

$$T_{Grouping} = O(s^۲ \log s + ns + \binom{\log(n)}{h}\alpha + n\log^۲(n)N_a)$$

[2] linklist

که با کوچک گرفتن s و این فرض که $\alpha = \theta(1)$ داریم:

$$T_{Grouping} = O(ns + \binom{\log(n)}{h} \log^2(n) N_a) << O(n^2) \quad (۳-۱۲)$$

روشن است که زمان لازم برای اجرای الگوریتم به روشی که در این رساله ارائه شد کمتر از زمان لازم برای الگوریتم رایج hill-climbing است.
به عنوان خلاصه و نتیجه‌گیری می‌توان به قضیه زیر رسید.

قضیه ۱.۳ زمان اجرای الگوریتم گروه‌بندی از زمان اجرای الگوریتم متداول hill-climbing کمتر است و به ترتیب زیر می‌باشد:

- در گراف‌های پراکنده که $E = O(n)$ است، $T_{HC} = \theta(n^2)$، در حالیکه $T_{Grouping} = \theta(ns + \binom{\log(n)}{h} \log^2(n) N_a)$
- در گراف‌های پر که $E = O(n^2)$ می‌باشد داریم $T_{HC} = \theta(n^3)$، در حالیکه $T_{Grouping} = \theta(n^2)$

۳-۴ حافظه لازم برای ذخیره مجموعه داده

در این بخش می‌خواهیم حجم حافظه‌ای را که برای ذخیره مجموعه داده نیاز داریم، محاسبه کنیم. با یک تخمین ساده نشان می‌دهیم که حجم این حافظه حتی برای مجموعه داده‌های بسیار بزرگ، چالش برانگیز نیست.

فرض کنید که شبکه اجتماعی مورد تحلیل دارای ۱۰۰۰۰۰۰۰۰ نود مستقل باشد. بعلاوه طبق تخمین‌های صورت گرفته، فرض می‌کنیم که به طور متوسط هر فرد دارای ۱۰۰ دوست باشد. چون $۱۰۰۰۰۰۰۰۰ < ۲^{۳۲}$ است، پس ۴ بایت برای نمایش هر نود کافیست. بعلاوه فرض می‌کنیم که وزن هر یال را با ۲ بایت نمایش دهیم. بنابراین با در نظر گرفتن هر نود، می‌توان هر یال متصل و وزن آن را با ۶ بایت نمایش داد.

با محاسبات بالا روشن است که برای هر نود نیاز به حافظه‌ای برابر ۶۰۰ بایت داریم و بنابراین در کل به حافظه‌ای به میزان حدود $۱۰۰۰۰۰۰۰۰ * ۶۰۰ = ۶۴GB$ نیاز است. با در نظر گرفتن این نکته که حافظه رایانه‌های شخصی نیز در حال حاضر در حدود چند صد گیگابایت می‌باشد، مشخص است که برای ذخیره مجموعه داده مشکلی نخواهیم داشت.

۳-۵ نتایج تجربی

نگرشی دوباره به گفته‌های این فصل مشخص می‌کند که ایده تبلیغات گروهی برای افزایش سود برای اولین بار در این رساله مطرح شده است. و در حقیقت این رساله آغازگر حوزه جدیدی در

استفاده از شبکه‌ها برای افزایش سود بوده است. بنابراین الگوریتم ارائه شده در این رساله تنها روشی است که در این حوزه مطرح شده و این باعث می‌شود که روش دیگری موجود نباشد که با بررسی نتایج دو الگوریتم، کارایی روش ما اثبات شود. از طرف دیگر ارزیابی روش گروه‌بندی در برابر روش‌های مطرح شده برای k نود تاثیرگذار از جمله روش hill-climbing خود با مشکلات و نواقصی روبروست زیرا معیارهای مساله و مدل‌ها در دو روش با هم متفاوت است. روشن است که بهترین روش برای ارزیابی الگوریتم ما اینست که یک آزمایش واقعی در یک شبکه اجتماعی صورت گیرد که در آن یک کالای جدید به ۲ روش در شبکه تبلیغ شود و سود نهایی در ۲ حالت با یکدیگر مقایسه شود. به عبارت روشن‌تر k نود تاثیرگذار شبکه توسط الگوریتم hill-climbing و m گروه تاثیرگذار توسط روش گروه‌بندی در این رساله، تشخیص داده شوند و با استفاده از این اطلاعات، کالا یا استراتژی جدید در شبکه تبلیغ شود. سود نهایی در هر حالت میزان کارایی روش‌ها را نسبت به هم مشخص خواهد کرد. اما در حال حاضر چون امکان همکاری با یک موسسه تجاری آنهم در چنین سطح وسیعی برای این تحقیق علمی فراهم نیست، ما به آزمایشات زیر بسنده می‌کنیم.

همانطور که بارها در این رساله و مقالات علمی در این زمینه گفته شده است، روش hill-climbing بهترین روش موجود برای یافتن k نود تاثیرگذار است. بنابراین در این بخش ما روش خود را با این روش مقایسه می‌نماییم. مقایسه بایستی به این صورت انجام شود که با هزینه کردن مبلغ یکسانی برای تبلیغات در ۲ روش، مشخص کنیم کدامیک سود بیشتری نصیب خواهند کرد. در روش k نود تاثیرگذار، ما به k نود کالا را به رایگان می‌دهیم. برای سادگی فرض می‌کنیم قیمت کالا ۱ واحد باشد. بدین ترتیب هزینه خرج شده برابر k خواهد بود. از طرف دیگر در روش m گروه تاثیرگذار، می‌خواهیم با صرف هزینه‌ای برای m گروه مهم شبکه، کالای خود را در شبکه محبوب کنیم. در این رساله فرض می‌کنیم که برای تمام گروه‌ها هزینه یکسان c صرف شود. یکی از کارهایی که می‌توان در آینده انجام داد، تخصیص هزینه‌های متفاوت به گروه‌ها براساس میزان اهمیت آنهاست. بدین ترتیب هزینه خرج شده در این حالت برابر mc خواهد بود. چون هزینه تبلیغ در دو روش بایستی برابر باشد، داریم: $k = mc$

بدین ترتیب روش آزمایش بدین صورت است که در hill-climbing به دنبال k نود تاثیرگذار گشته و به آنها کالا را رایگان می‌دهیم و میزان انتشار را در شبکه محاسبه می‌کنیم و در گروه‌بندی m گروه تاثیرگذار را یافته و برای آنها هزینه‌ای برابر k/m صرف تبلیغات می‌کنیم و درنهایت میزان سود حاصل از این روش را نیز به دست می‌آوریم. مقایسه سود حاصل از دو روش، معیاری برای ارزیابی کارایی روش ارائه شده این رساله خواهد بود.

مثال ۳.۳ فرض کنید که شبکه‌ای داریم با ۱۰۰ نود که می‌خواهیم ۱۰ نود تاثیرگذار و یا ۳ گروه تاثیرگذار را انتخاب کنیم. بدین ترتیب در هر گروه با خرج هزینه‌ای برابر $\frac{۱۰}{۳}$، میزان انتشار را محاسبه می‌کنیم. بدین صورت بدون درنظرگرفتن انسجام گروهی حداقل ۳۳٪ از هر گروه استراتژی جدید را در همان شروع کار می‌پذیرند و انتشار را آغاز می‌کنند. در طرف مقابل با در نظر گرفتن انسجام گروهی برابر ۱، در ۳ گروه اولیه به اندازه ۶۶٪ اعضا در ابتدای کار استراتژی را اخذ

می‌کنند. این نتایج با استفاده از تخمین $P_{\%i} = \min(۱, \frac{C(۱+\theta_i)}{N_i x})$ به دست آمده است.

مثال ۴.۳ در یک شبکه با اندازه ۱۰٬۰۰۰٬۰۰۰ نود، اگر فرض کنیم که $K = ۱۰۰۰$ و $m = ۱۰$، بایستی به هر گروه ۱۰۰ واحد تبلیغ شود. با در نظر گرفتن این موضوع که طبق تخمین‌های ما در هر گروه حدود ۵۰۰ نود عضو خواهند بود، میزان اتخاذ استراتژی در ابتدای کار در هر یک از این گروه‌ها مقداری بین ۲۰% تا ۴۰% خواهد بود.

برای آزمایش دو روش از مجموعه داده GAGNON & MACRAE PRISON استفاده کرده‌ایم که یکی از مجموعه داده‌های استاندارد UCINET IV [۳] است. این مجموعه داده در سال ۱۹۵۰ توسط John Gagnon تهیه شده است که در آن از ۶۷ زندانی خواسته شده که صمیمی ترین دوستان خود را از میان سایر افراد تعیین کنند. هر زندانی مجاز بوده که به تعداد دلخواه دوست صمیمی تعیین کند. این مجموعه داده یک ماتریس غیرمتقارن دودویی تشکیل می‌دهد. در این آزمایش میزان تاثیرگذاری دوستان هر فرد بر او را یکسان فرض کرده‌ایم. به عبارت دیگر اگر فردی ۸ دوست داشته باشد، تصور می‌شود که هر کدام از دوستان بر او به میزان برابر ۰.۱۲۵ تاثیرگذار است. هر دو روش بر روی مجموعه داده فوق آزمایش شده است و نتایج آنها در جداول ۳-۱، ۳-۲ و ۳-۳ قابل مشاهده است. کد برنامه به زبان جاوا نوشته شده است و این کدها روی کامپیوتری با پردازنده $۲GHz$ و دارای حافظه اصلی به ظرفیت $۲GB$ اجرا شده است. در ادامه جزییات بیشتر هر روش را توضیح داده‌ایم.

در روش Hill-Climbing، هدف یافتن $k = \{۴, ۵, ۶, ۷\}$ نود تاثیرگذار است که سود بیشینه را به ارمغان بیاورد. در روش Grouping، $\log(۶۷) = ۶$ گروه تشکیل داده می‌شوند و آزمایش بر روی این گروه‌ها انجام می‌شود. می‌خواهیم m گروه تاثیرگذارتر را در این میان بیابیم. مشخص است که میزان تبلیغات روی هر گروه برابر $c = k/m$ واحد خواهد بود. در این آزمایش به علت حجم قابل قبول داده نیازی به استفاده از روش سلسله‌مراتبی و نمونه‌برداری احساس نمی‌شود و بنابراین کل مجموعه داده بر اساس الگوریتم بهینه‌سازی شده ROCK در این رساله، گروه‌بندی شده‌اند. سپس در نهایت برای چند عضویتی کردن نودها، هر نود به گروه‌هایی که ارتباط بیش از ۳۰% دارد، اختصاص می‌یابد. همانطور که پیشتر گفته بودیم، منظور از ارتباط نود با گروه، نسبت مجموع وزنی نود با همسایه‌هایش در گروه به کل همسایه‌هایش می‌باشد. شایان ذکر است که چون یافتن سود بهینه یک مساله $NP - Complete$ است، نمی‌توانیم جواب خود را با حالت بهینه مقایسه نماییم و در عوض با مقایسه میان جواب الگوریتم خود با الگوریتم متداول Hill-Climbing، کارایی الگوریتم خود را نمایش می‌دهیم.

بنابراین در یک جمع‌بندی آزمایش به این صورت طراحی شده است. در روش Hill-Climbing

---

[۳] برای جزییات بیشتر و بارگذاری فایل مربوطه می‌توانید به آدرس مقابل مراجعه کنید.

http://vlado.fmf.uni-lj.si/pub/networks/data/UciNet/UciData.htm

جدولِ ۳–۱: مقایسه میزان انتشار (سود نهایی) حاصل از دو الگوریتم HC و انتشار گروه محور. (m تعداد گروه‌هایی است که در ابتدا مورد تبلیغ قرار می‌گیرند و c هزینه‌ای است که به هر کدام از این گروه‌ها تخصیص می‌یابد.)

نام الگوریتم	HC	انتشار گروه محور			
		θ			
k		(m,c)	۰.۳	۰.۱	۰.۰۵
۷	%۸۵	(۲,۳.۵)	%۸۰.۹	%۸۵.۷	%۸۵.۸
۶	%۸۰	(۳,۲)	%۷۶.۷	%۸۴	%۸۴
۶	%۸۰	(۲,۳)	%۷۸.۵	%۸۴	%۸۴
۵	%۷۴.۷	(۲,۲.۵)	%۷۶	%۸۲	%۸۲
۴	%۶۷	(۲,۲)	%۷۳	%۸۰	%۸۰

هدف یافتن سود بهینه با تبلیغ کالا به k افرد است. در انتشار گروه محور، هدف تبلیغ کالا به m گروه از کل $\log(n)$ گروه موجود با هزینه تبلیغی برابر روش Hill-Climbing است. می‌دانیم که در هر دو روش نیاز به یافتن δ هر نود یا احتمال فعال‌سازی هر یال داریم. همانطور که در [۲] آمده است، شبیه‌سازی رفتار تصادفی فرایند، تخمین بسیار خوبی برای این مقادیر به دست می‌دهد. بعلاوه در همان مقاله نشان داده شده است که نتایج اجرای الگوریتم پس از ۱۰۰۰۰ بار اجرا، تفاوت قابل ملاحظه‌ای با اجرای ۳۰۰۰۰۰ باری یا حتی بیش از آن نمی‌کند. براین اساس ما هر دو روش را به اندازه ۱۰۰۰۰ بار اجرا می‌کنیم که در هر بار اجرا مقادیر δ و خروجی یال‌ها به صورت تصادفی از بازه $[۰,۱]$ استخراج می‌شود. جدول ۳–۱ درصد پیشرفت استراتژی نوین در شبکه (در حقیقت نسبت نودهای راضی شده به کل نودها) را در هر یک از دو روش نشان می‌دهد.

اشاره کردیم که پارامتر θ به مجموعه داده بستگی دارد و مشخص می‌کند که نودهای مجاور تا چه حد بایستی بر یکدیگر تاثیرگذار باشند، تا به عنوان همسایه معرفی شوند. به روشنی می‌توان مشاهده کرد که تعیین میزان معقولی برای θ تا چه میزان می‌تواند بر دقت و سرعت و میزان انتشار در الگوریتم بیفزاید، در حالیکه مقادیر غیرمعقول چه کوچک باشند و چه بزرگ، بر پاسخ الگوریتم سایه می‌افکنند. مقادیر کوچک θ می‌تواند بسیاری از نودها را که ارتباط کافی با هم ندارند، به عنوان همسایه معرفی کند و بدین روش، اطلاعات نادرست را وارد مدل کند. در طرف مقابل، مقادیر بزرگ θ نیز می‌تواند بسیاری از همسایه‌ها را از محاسبات حذف کرده و از انتشار تا حد زیادی جلوگیری کند. جدول ۳–۱ تایید می‌کند که همانقدر که تعیین ۰.۱ و ۰.۰۵ می‌تواند پاسخ الگوریتم انتشار گروه محور را نسبت به Hill-Climbing بهتر کند، مقادیر دورتر مانند ۰.۳ می‌تواند پاسخ الگوریتم را تضعیف کند. روشن است که مقادیر نامعقول‌تر (مانند ۰.۸) می‌تواند انتشار را به شدت کاهش دهد و زمان اجرای الگوریتم را نیز به شدت افزایش دهد.

جدول ۳–۲ زمان اجرای دو روش را در کل با هم مقایسه می‌کند. این جدول زمان اجرای الگوریتم به روش HC و انتشار گروه محور را به شرط اجرای ۱۰۰۰۰ باره برای یافتن تخمین

جدولِ ۳-۲: مقایسه زمان کل اجرای دو الگوریتم انتشار گروه محور و HC برحسب میلی ثانیه برای ۱۰۰۰۰ بار تکرار. (m تعداد گروه‌ها و c هزینه را مشخص می‌کند.)

نام الگوریتم	HC	انتشار گروه محور			
		θ			
k		(m,c)	۰.۳	۰.۱	۰.۰۵
۷	۵۸۹s	(۲,۳.۵)	۷۰s	۷۰s	۷۲s
۶	۴۶۵s	(۳,۲)	۱۶۸s	۱۵۵s	۱۵۶s
۶	۴۶۵s	(۲,۳)	۷۴s	۷۴s	۷۵s
۵	۳۴۶s	(۲,۲.۵)	۷۹s	۸۲s	۸۱s
۴	۲۳۸s	(۲,۲)	۸۷s	۸۶s	۸۶s

جدولِ ۳-۳: مقایسه زمان اجرای بخش‌های مختلف الگوریتم انتشار گروه محور برای $\delta = ۰.۱$ (برحسب میلی ثانیه)

فاز الگوریتم	$(m,c) = (۲,۳)$	$(m,c) = (۳,۲)$	$(m,c) = (۲,۲)$
گروه‌بندی	۴۷ ms	۴۷ ms	۴۶ ms
مدلسازی	۰ ms	۰ ms	۰ ms
محاسبه انتشار (۱۰۰۰۰ بار)	۱۵۵۸۵۹ ms	۷۴۶۷۷ ms	۸۶۷۹۹ ms

درستی از پارامترها نشان می‌دهد. به روشنی می‌توان کاهش زمان اجرا را توسط الگوریتم انتشار گروه محور نسبت به HC مشاهده کرد. در جدول ۳-۳ نیز زمان اجرای هر بخش الگوریتم انتشار گروه محور به تفکیک برای بخش‌های گروه‌بندی، مدلسازی و محاسبه انتشار گروه محور آمده است. لازم می‌دانیم که بار دیگر اشاره کنیم که به علت حجم معقول مجموعه داده Prison از نمونه‌برداری استفاده نکرده‌ایم و بنابراین زمان آن بخش از پیچیدگی $O(n^۲ \log(n))$ برخوردار است. بدیهی است که با افزایش حجم مجموعه داده و استفاده از نمونه‌برداری زمان گروه‌بندی به شدت نسبت به الگوریتم Hill-Climbing کاهش می‌یابد.

به روشنی می‌توان دید که الگوریتم پیشنهادی ما در عین کاهش ملموس زمان اجرا، میزان انتشار را نیز افزایش داده است.

فصلِ ۴

طراحی مکانیزم در شبکه‌های اجتماعی

۴-۱ مختصری بر طراحی مکانیزم

طراحی مکانیزم بخشی از بحث نظریه بازی‌هاست که راه‌حل‌های ممکن برای بازی‌هایی با اطلاعات خصوصی را بررسی می‌کند. منظور از اطلاعات خصوصی، اطلاعاتی است که برخی از نودها آن را دانسته و تمام نودها از آن آگاه نیستند. در این موارد، طراح بازی قواعد و ساختار بازی را برای ارضای هدفی مشخص تعیین می‌کند. در این بازی‌ها طراح بازی سعی می‌کند تا با تضمین خواسته‌های بازیکنان، آنها را راضی کند تا اطلاعات خصوصی خود را به درستی آشکار نمایند. در حقیقت تضمین راستگویی در آشکارسازی اطلاعات خصوصی مهمترین هدفی است که در طراحی مکانیزم به آن توجه می‌شود. به روشنی می‌توان اهداف مکانیزم‌های طراحی شده را در واژه‌ای مانند انتخاب اجتماعی[1] مشاهده کرد. یک انتخاب اجتماعی در حقیقت برآیند نظرات شرکت‌کنندگان در یک طرح برای یک تصمیم مشترک است. مثالهایی از چنین تصمیم‌گیری‌هایی را می‌توان در ابعاد گسترده در دنیای امروز مشاهد کرد. مسائلی چون انتخابات، بازارها، حراج‌ها و تصمیم‌گیریهای کلان دولتی که برپایه خواست‌های مردم استوار است، نمونه‌هایی از چنین تصمیم‌گیری‌هایی است که در [۱۵] به آنها اشاره شده است.

بازی طراحی مکانیزم در حقیقت یک بازی با اطلاعات مخفی و خصوصی است که یک عامل[2] که نقش اصلی را در آن برعهده دارد، ساختار سود بازی را تعیین می‌کند. بازیکنان دارای یک مجموعه از اطلاعات خصوصی هستند. به عنوان مثال اطلاعات خصوصی می‌تواند شامل ترجیحات آنها یا کیفیت کالاها باشد. این قبیل اطلاعات را نوع[3] بازیکنان گویند که با θ نمایش داده می‌شود. بازیکنان سپس مقادیری که ممکن است دروغی تدبیرشده برای افزایش سود باشد، به عنوان نوع

[1] Social Choice

[2] agent

[3] type

خصوصی خود به عامل اصلی ارسال می‌کنند. این بسته‌های ارسالی حاوی نوع را با $\hat{\theta}$ نشان می‌دهیم. گفتیم که $\hat{\theta}$ ممکن است به قصد افزایش سود با θ واقعی برابر نباشد. پس از گزارش مقادیر توسط بازیکنان، براساس ساختاری که توسط عامل اصلی تعیین شده، هزینه و سود بازیکنان و عامل اصلی تعیین می‌شود. در حقیقت خروجی مکانیزم را می‌توان به ۲ بخش تقسیم کرد. بخش تخصیص کالاها، $x(\theta)$، و بخش پرداخت پول، $t(\theta)$، که هر دو توابعی از نوع‌های گزارش شده‌اند. به این ترتیب مشخص می‌شود که کالا یا استراتژی نوین به کدامین بازیکنان خواهد رسید و بازیکنان چه میزان هزینه می‌پردازند یا چه میزان سود می‌برند.

در یک جمع‌بندی می‌توان مطالب بالا را به صورت دقیق‌تری بیان کرد. در یک طراحی مکانیزم براساس نوع خصوصی گزارش شده افراد، نه تنها یک انتخاب اجتماعی مشخص می‌شود، بلکه هزینه‌های مالی نیز که توسط افراد مختلف بایستی پرداخت شود، تعیین می‌گردد.

تعریف ۱.۴ به صورت صوری، مجموعه‌ای از انتخاب‌های ممکن متصور است که با A نمایش داده می‌شود. برای هر بازیکن i یک تابع ارزش $v_i : A \to R$ متصور است که اولویت این بازیکن را برای انتخاب‌های متفاوت مدل می‌کند. بنابراین یک مکانیزم، یک تابع انتخاب اجتماعی $f : V_1 \times V_2 \times \cdots \times V_n \to A$ و یک بردار از توابع هزینه $p_1, \cdots, p_n$ است که $p_i : V_1 \times \cdots \times V_n \to R$ مقداری است که بازیکن i پرداخت می‌کند.

تعریف ۲.۴ یک مکانیزم $(f, p_1, \cdots, p_n)$ راستگوست اگر برای هر بازیکن i، هر v_1 تا v_n و هر v_i'، اگر $a = f(v_i, v_{-i})$ و $a' = f(v_i', v_{-i})$، آنگاه $v_i(a) - p_i(v_i, v_{-i}) \geq v_i(a') - p_i(v_i', v_{-i})$. این بدین معناست که بازیکن i با تابع اولویت v_i ترجیح می‌دهد که بجای گزارش هر دروغ ممکن v_i'، تابع راست v_i را به مکانیزم بگوید، زیرا بدین ترتیب سود بیشتری نصیب او خواهد شد.

در میان مباحثی که در طراحی مکانیزم‌ها مطرح شده است، مکانیزم‌های VCG[۴][؟]، [۱۷] و [۱۸] از اهمیت بالایی برخوردارند و ثابت شده است که راستگو هستند. در این مکانیزم‌ها، اساس کار، بیشینه کردن تابع رفاه اجتماعی[۵] است. رفاه اجتماعیِ یک کاندید $a \in A$ برابر مجموع ارزش تمامی بازیکنان برای a است که برابر $\sum_i v_i(a)$ می‌باشد.

تعریف ۳.۴ مکانیزم $(f, p_1, \cdots, p_n)$ یک مکانیزم VCG است اگر:

- تابع f تابع رفاه اجتماعی را بیشینه کند، یعنی

$$f(v_1, \cdots, v_n) \in argmax_{a \in A} \sum_i v_i(a)$$

- برای برخی توابع $h_i : v_{-i} \to R$ که $h_n, \cdots, h_1$ (h_i به v_i بستگی ندارد)، برای $v_n, \cdots, v_1$ داریم:

$$P_i(v_1, \cdots, v_n) = h_i(v_{-i}) - \sum_{j \neq i} v_j(f(v_1, \cdots, v_n))$$

[4] Vickery-Clarke-Groves

[5] Social Welfare

۴-۲ نقص اطلاعات شبکه‌های اجتماعی و طراحی مکانیزم

همانطور که می‌دانیم مساله طراحی مکانیزم[6] به این صورت است که یک کالا و تعدادی مشتری داریم. هر مشتری میزان علاقه‌مندی خود را به کالای یادشده به درستی یا دروغ اعلام می‌کند. مکانیزم تصمیم می‌گیرد که کالا را به چه کسی بدهد و افراد چقدر پول پرداخت نمایند. بعلاوه سعی بر آنست که مکانیزم طوری طراحی شود که مشتریان با دروغ گفتن سود نکنند و در اصطلاح مکانیزم راستگو[7] باشد. می‌توان از همین ایده برای شبکه‌های اجتماعی نیز استفاده کرد. این روش مخصوصاً در شرایطی که اطلاعات و دانش ما ناقص است و جمع‌کردن اطلاعات هزینه دارد که معمولاً در بیشتر موارد واقعی چنین است، بسیار مفید به نظر می‌رسد. در حقیقت به دست آوردن این اطلاعات هم وقت‌گیر است و هم پرهزینه و در بسیاری از مواقع غیرممکن خواهد بود.

طبق این روش هر نود میزان تاثیرگذاری خود را بر سایرین اعلام می‌کند. توجه کنید که ممکن است به دروغ یا راست گفته باشد. سپس مکانیزم بر اساس این اطلاعات تصمیم می‌گیرد که به چه کسانی و به چه قیمتی کالای خود را بفروشد. بدین ترتیب ممکن است به افرادی که تاثیرگذاری بالایی دارند کالا به رایگان یا به قیمت کمی فروخته شود که این همان نقش تبلیغاتی را خواهد داشت.

برای اینکه تضمین کنیم که مکانیزم راستگوست از روش VCG استفاده می‌نماییم. در میان ارزش‌هایی که نودها به عنوان میزان ارزش‌گذاری خود تعریف می‌کنند، مقدار بیشینه را در نظر می‌گیریم. متغیر α را به صورت $\alpha = \frac{\max val}{price}$ تعریف می‌کنیم. فرض کنید که شرکت تصمیم گرفته به k نفر تبلیغ کند. در این صورت به kنفری که بیشترین ارزش را اعلام کرده‌اند کالا را با تخفیف می‌فروشیم. هزینه‌ای که هر یک از این افراد باید برای خرید کالا بپردازند برابر $\frac{\max_{k+1} val}{\alpha}$ است. $\max_{k+1} val$ مقدار بیشینه $k+1$امین را نشان می‌دهد. با این روش به افراد برنده مقدار $price - \frac{\max_{k+1} val}{\alpha}$ تخفیف داده شده و مقدار هزینه‌ای که شرکت برای تبلیغات خرج کرده است برابر $K * (price - \frac{\max_{k+1} val}{\alpha})$ است.

اما نودهای شبکه در ابتدای کار خود از میزان تاثیرگذاریشان بر شبکه اطلاعی ندارند. در حقیقت نودهای شبکه تنها از همسایگان خود و وزن یالهای میانی که تاثیرگذاری آنها را برهم نشان می‌دهد، مطلعند. بنابراین قبل از اجرای الگوریتم طراحی مکانیزم براساس میزان تاثیرگذاری نودها، لازم است تا به روشی اهمیت نودها در شبکه مشخص شود. در ادامه فصل هدف ما طراحی الگوریتمی است که توسط آن، نودها از میزان تاثیرگذاری خود در شبکه مطلع شوند.

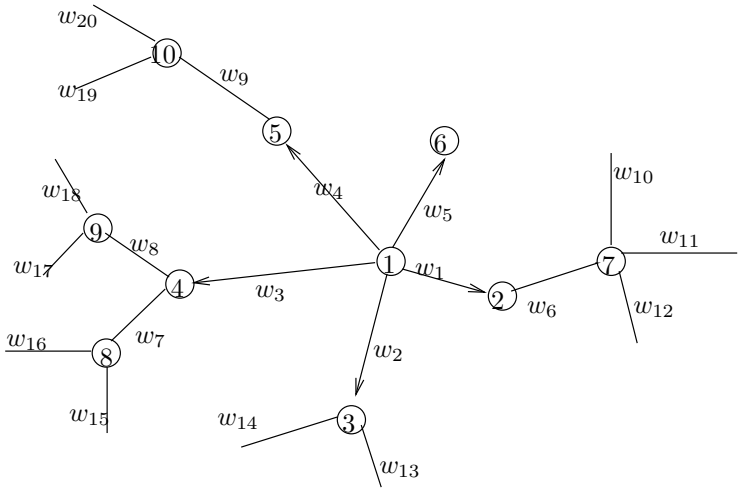
---

[6] Mechanism Design

[7] Truthful

۴-۳ تعیین میزان تاثیرگذاری نودها در شبکه اجتماعی

در بخش‌های قبل گفتیم که برای تعیین نودهای تاثیرگذار در شبکه و آغاز تبلیغات از آنها، نیاز داریم تا میزان تاثیرگذاری هر نود شبکه را داشته باشیم. بدین منظور لازم می‌دانیم تا تغییراتی در مدل انتشار اعمال کنیم. به منظور محاسبه میزان تاثیرگذاری نودها در کل شبکه، مدل انتشار طوری تغییر می‌کند که عدم قطعیت در آن از بین رفته و مدل پیوسته‌سازی می‌شود. بدین معنا که آستانه پذیرش و احتمال فعال‌سازی یالی از مدل حذف شده و میزان تاثیرگذاری نودها بر یکدیگر چه مستقیم و چه غیرمستقیم مبنای تصمیم گیری قرار خواهد گرفت. این کار بدین دلیل است که هدف در این روش یافتن ارزش نودها در شبکه است و نه شبیه‌سازی انتشار نوآوری در دنیای مجازی احتمالاتی. مثال زیر مبانی محاسبه میزان تاثیرگذاری را نشان می‌دهد.



شکلِ ۴-۱: تاثیر همسایه‌ها بر میزان تاثیرگذاری نود ۱

مثال ۱.۴ شبکه شکل ۴-۱ را در نظر بگیرید. میزان تاثیرگذاری نود i را با val_i نمایش می‌دهیم. فرض می‌کنیم که میزان تاثیرگذاری نودهای ۱۰ – ۲ در شبکه‌ای که از حذف نود ۱ به دست آمده، مشخص شده و برابر $val_۲$ تا $val_{۱۰}$ باشند. میزان تاثیرگذاری نود ۱ به روش زیر محاسبه می‌شود.

$$val_۱ = w_۱ \times val_۲ + w_۲ \times val_۳ + w_۳ \times val_۴ + w_۴ \times val_۵ + w_۵ \times val_۶ + ۱$$

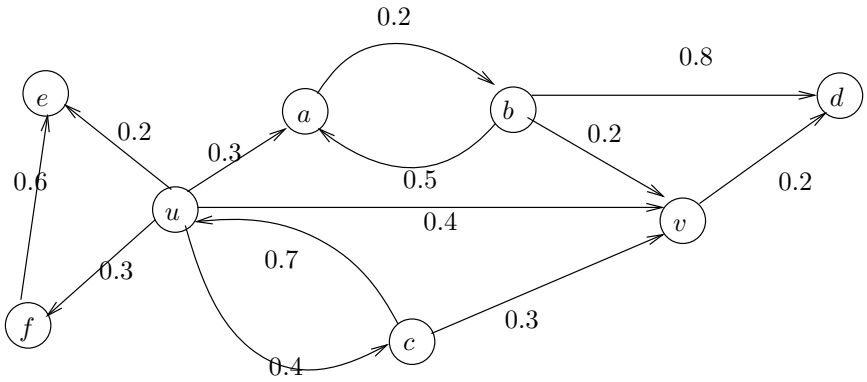
که در آن به عنوان مثال مقدار $w_۱ \times val_۲$ مشخص می‌کند که نود ۱ از طریق همسایه ۲ چقدر بر کل شبکه تاثیرگذار است. جمع عدد ۱ در عبارت بالا، برای درنظرگرفتن ارزش راضی شدن خود نود ۱ می‌باشد. بدین ترتیب ما می‌خواهیم دریابیم هر نود به صورت با واسطه (غیر مستقیم) یا بی‌واسطه (مستقیم) چقدر بر سایر نودها تاثیرگذار است. در حقیقت تعریف زیر روشن می‌کند که منظور از

میزان تاثیرگذاری نودها در شبکه چیست.

تعریف ۴.۴ میزان تاثیرگذاری نود u در یک شبکه اجتماعی برابر مجموع تاثیرگذاری u بر هر یک از نودهای شبکه خواهد بود. (توجه آنکه میزان تاثیرگذاری نود u بر خودش برابر ۱ در نظرگرفته می‌شود و مقصود از تاثیرگذاری نود u بر نود v، مجموع تاثیرگذاری باواسطه و بی‌واسطه u بر v می‌باشد.)

واضح است که تاثیرگذاری بی واسطه نود u به نود v برابر وزن یال $(u \to v)$ است (طبق تعریف وزن یال). تاثیرگذاری باواسطه نود u به v نیز به این معناست که نود u از طریق سایر نودهای شبکه تا چه اندازه می‌تواند روی نود v تاثیر گذار باشد. بدیهی است که این مقدار برابر خواهد بود با مجموع تاثیرگذاری نود u بر v از طریق مسیرهای غیر مستقیم مختلف موجود از u به v در گراف شبکه . میزان تاثیرگذاری نود u بر v از طریق یک مسیر معین نیز به ترتیب زیر محاسبه می‌شود.

تعریف ۵.۴ میزان تاثیرگذاری باواسطه نود u بر نود v از طریق مسیر p که از u آغاز شده و به v منتهی می‌شود، برابر است با حاصلضرب وزن یال‌های جهتدار مسیر p. مثال زیر برای روشن‌شدن بیشتر موضوع مفید خواهد بود.



شکلِ ۴-۲: محاسبه میزان تاثیرگذاری با واسطه نود u به نود v

مثال ۲.۴ می‌خواهیم میزان تاثیرگذاری باواسطه نود u به نود v را در شکل ۴-۲ محاسبه نماییم. براساس توضیحات ذکرشده برای محاسبه به صورت زیر عمل می‌کنیم.

میان نود u و v دو مسیر غیرمستقیم وجود دارد. مسیر اول به صورت $(u \to a \to b \to v)$ است و مسیر دوم به صورت $(u \to c \to v)$ می‌باشد. بدین ترتیب میزان تاثیرگذاری با واسطه u بر v برابر است با:

$$IndInf(u \to v) = (۰.۳ \times ۰.۲ \times ۰.۲) + (۰.۴ \times ۰.۳) = ۰.۰۱۲ + ۰.۱۲ = ۰.۱۳۲$$

همانطور که بالاتر اشاره کردیم، میزان تاثیرگذاری هر نود بر نود دیگر برابر است با مجموع تاثیرگذاری بی‌واسطه و باواسطه نود اول بر روی نود دوم. بنابراین میزان تاثیرگذاری نود u بر نود v برابر است با:

$$Inf(u \to v) = ۰.۱۳۲ + ۰.۴ = ۰.۵۳۲$$

همچنین ارزش تاثیرگذاری هر نود در شبکه برابر خواهد بود با مجموع تاثیرگذاری آن نود بر کل نودهای شبکه. بنابراین برای نود u داریم:

$$Inf(u) = Inf(u \to u) + Inf(u \to v) + Inf(u \to a) + Inf(u \to b) + \cdots + Inf(u \to f)$$

اما برای به دست آوردن این مقادیر به صورت متمرکز با ۲ مشکل برخورد می‌کنیم:

- اول اینکه نود مرکز از اطلاعات شبکه به صورت کامل مطلع نیست و همانطور که گفتیم نقص اطلاعات داریم و در حقیقت اطلاعات کامل را خود نودها در اختیار دارند. بنابراین برپایه‌ی این دانش ناقص نمی‌توان به مقادیر درست دست یافت.
- دوم اینکه به فرض اینکه با استفاده از هزینه و وقت کافی توانستیم اطلاعات کامل را جمع آوری کرده و در اختیار یک مرکز قرار دهیم، برای محاسبه ارزش تاثیرگذاری نودها به صورت متمرکز زمان زیادی لازم است.

این دو مشکل ما را قانع می‌کند که کار محاسباتی را به نوعی توزیع شده و با استفاده از خود نودها انجام دهیم. پیش از این گفتیم که حل مساله نودهای تاثیرگذار در این روش، یک مسابقه میان نودهاست و نودها مایلند برای تخفیف گرفتن و سود بیشتر در این مسابقه شرکت کنند. بنابراین ما قصد داریم تا این مسابقه را طوری طراحی کنیم که جواب مساله ما به دست بیاید با این تضمین که نودها راستگو و راست کردار عمل کرده‌اند. راستگویی بدین معناست که نودها مقادیر خصوصی خود را به درستی بیان کنند و راست کرداری بدین مفهوم است که نودها در محاسبات و انتقال پیام‌ها به درستی عمل کنند. برای حل توزیع شده مساله، ایده‌های الگوریتم BGP مفید به نظر می‌رسد. در ادامه ما توضیح مختصری در خصوص الگوریتم BGP می‌دهیم و سپس الگوریتم توزیع شده محاسبه ارزش‌ها را ارائه می‌دهیم.

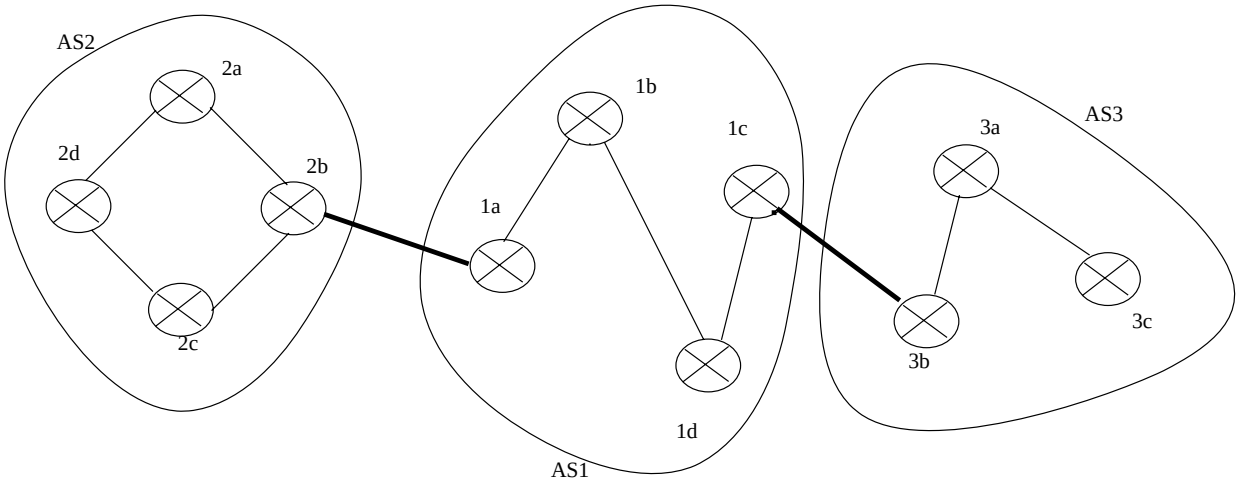
۴-۳-۱ مختصری بر الگوریتم BGP

پروتکل Border Gateway Protocol نسخه ۴ که در RFC 1771 مطرح شده است، به عنوان استاندارد مسیریابی میان سیستم‌های مستقل[8] در اینترنت به کار می‌رود. یک سیستم مستقل به مجموعه‌ای

[8] Autonomous System

از نودهای کامپیوتری اتلاق می‌شود که تحت مدیریت یک سازمان با یک استراتژی یکسان اداره می‌شوند. در این زیربخش برآنیم تا نگاهی اجمالی به چگونگی عملکرد این الگوریتم براساس مطالب [۱۹] بیندازیم. BGP برای هر سیستم مستقل امکانات زیر را فراهم می‌آورد.

- اطلاعات دسترسی به زیرشبکه‌ها را از سیستم‌های مستقل همسایه دریافت می‌کند.
- اطلاعات دسترسی را به همه مسیریاب[9]های درونی ارسال می‌کند.



شکلِ ۴-۳: یک شبکه کوچک متشکل از ۳ سیستم مستقل

در BGP، جفت مسیریاب‌ها اطلاعات مسیریابی را روی اتصالات TCP به یکدیگر ارسال می‌کنند. در BGP هر سیستم مستقل دریافت می‌کند که چه مقصدهایی از طریق همسایگانش قابل دسترسی‌اند. چگونگی کار BGP را توسط شکل ۴-۳ توضیح می‌دهیم. در ابتدای کار سیستم مستقل AS۱ از طریق اتصال میان ۱a با ۲b لیستی از مقاصدی که از طریق آن قابل دسترسی است به AS۲ می‌فرستد. در این لیست مجموعه‌ای از پیشوندها وجود دارد که بازه‌ای از IPهای قابل دسترسی را مشخص می‌کند. به عنوان مثال پیشوند ۸۱.۳۱.۱۶۴/۲۴ بیانگر آدرس‌های IP از ۸۱.۳۱.۱۶۴.۰ تا ۸۱.۳۱.۱۶۴.۲۵۵ می‌باشد. بعلاوه AS۲ نیز لیستی از پیشوندهایی که از طریق آن قابل دسترسی است به AS۱ ارسال می‌کند. اتفاق مشابهی برای AS۱ و AS۳ از طریق اتصال میان ۱c و ۳b می‌افتد. سپس هر سیستم مستقل پس از دریافت اطلاعات از سیستم‌های مستقل همسایه، این اطلاعات را به همه مسیریاب‌های درونی خود ارسال می‌کند. بنابراین مسیریاب ۱c نیز از اطلاعات سیستم مستقل AS۲ مطلع می‌شود و بنابراین بار دیگر AS۳ را از آنها باخبر می‌کند. همین عمل را مسیریاب ۱a برای سیستم مستقل AS۲ انجام می‌دهد.

هرگاه یک مسیریاب از مسیر جدیدی برای یک پیشوند مطلع می‌شود، یک قلم داده پیشوندی به جدول مسیریابی‌اش اضافه می‌کند. در ارسال اطلاعات پیشوندها، موارد دیگری نیز به همراه آنها

[9] router

ارسال می‌شوند که دو داده مهم در این میان مسیر–سیستم مستقل[10] و گام–بعدی[11] است.

- مسیر–سیستم مستقل. این داده شامل سیستم‌های مستقلی است که پیام پیشوندی از آنها تاکنون عبور کرده است. هرگاه یک پیام پیشوندی از یک سیستم مستقل عبور کند، آن سیستم مستقل، شماره خود را به داده مسیر–سیستم مستقل اضافه می‌کند. مسیریاب‌ها از این داده استفاده می‌کنند تا از حلقه‌ها اجتناب شود، بدین روش که هر مسیریاب هنگامیکه متوجه شد که شماره سیستم مستقلش در داده مسیر–سیستم مستقل وجود دارد، آن پیام را حذف کرده و به همسایه‌هایش ارسال نمی‌کند.
- گام–بعدی. دو سیستم مستقل می‌توانند از طریق لینک‌های فیزیکی متعددی به طور بی‌واسطه با هم در ارتباط باشند. بنابراین هنگامی که یک بسته بخواهد از $AS۱$ به $AS۲$ منتقل شود، باید تعیین شود که از طریق کدام دروازه خروجی[12] این انتقال انجام گیرد. بنابراین هرگاه دروازه خروجی یک سیستم مستقل پیام مسیریابی حاوی لیستی از پیشوندها به سیستم مستقل همسایه ارسال می‌کند، IP خود را بایستی به این اطلاعات اضافه نماید.

۴–۳–۲ الگوریتم محاسبه تاثیرگذاری‌ها

مشابه الگوریتم BGP به این صورت عمل می‌کنیم که کل کار الگوریتم در چند مرحله انجام می‌شود. در هر مرحله نودها اطلاعاتی را به همسایگان خود می‌فرستند و هر نود در مرحله بعدی پس از دریافت اطلاعات از همسایه‌های خود به انجام محاسباتی می‌پردازد و نتیجه آن محاسبات را در اختیار همسایگانش قرار می‌دهد. بدین ترتیب در هر مرحله هر نود u ارزش تاثیرگذاری خود را بر اساس اطلاعاتی که تابحال از همسایگانش دریافت کرده محاسبه می‌نماید و به هر نود همسایه v تاثیری را که از طریق u می‌تواند بر شبکه بگذارد، اعلام می‌کند. برای روشن‌شدن و دقیق‌تر شدن مطلب، مراحل کار را در قالب فرمول‌های زیر آورده‌ایم.

فرض کنید که میزان تاثیرگذاری نود u را در مرحله t با $val^t(u)$ نمایش دهیم. بعلاوه فرض کنید که مقدار ارسالی نود u به نود v برابر $val^t(u \to v)$ باشد (این مقدار را از این پس ارزش ارسالی ناقص می‌نامیم) . آنگاه خواهیم داشت:

$$val^t(u) = \sum_{v \in neighbors(u)} w_{uv} \times val^{t-۱}(v \to u) + ۱ \qquad (۴–۱)$$

و

$$val^t(u \to v) = val^t(u) - w_{uv} \times val^{t-۱}(v \to u) \qquad (۴–۲)$$

[10] AS–PATH

[11] NEXT–HOP

[12] Gateway

بدین ترتیب الگوریتم محاسبه تاثیرگذاری‌ها به صورت زیر خواهد بود:

۱. در ابتدا برای تمام نودهای i، قرار می‌دهیم: $val^{\circ}(i) = ۱$ و $val^{\circ}(i \to j) = ۱$ که ۱ میزان تاثیرگذاری نود i بر خودش را نشان می‌دهد و j یک همسایه i است.

۲. در مرحله $t > ۰$ام هر نود i، میزان تاثیرگذاری $val^{t-۱}(i \to j)$ را به هر همسایه j اعلام می‌کند و سپس هر نود j براساس مقادیری که از همسایگان گرفته مقدار ارزش تاثیرگذاری خود را، $val^{t}(i \to j)$، محاسبه می‌نماید و برای اعلام در مرحله بعد آماده می‌شود.

قضیه ۱.۴ نتیجه نهایی اجرای الگوریتم در مرحله t برای هر نود برابر مجموع تاثیرگذاری‌های مستقیم و غیرمستقیم با $t-۱$ واسطه به سایر نودهاست (یا به عبارت دیگر برابر مجموع مسیرهای تاثیرگذاری با حداکثر t یال).

اثبات. (اثبات با استقرا).

پایه: در پایان مرحله ۰، ارزش هر نود برابر ۱ است و از طرف دیگر مجموع مسیرهای تاثیرگذاری با حداکثر ۰ یال متناظر است با تاثیرگذاری نود بر خودش که برابر ۱ می‌باشد و بنابراین پایه استقرا برقرار است.

فرض: فرض می‌کنیم که ارزش تاثیرگذاری نودها در پایان مرحله t برابر مجموع تاثیرگذاری مسیرهای t یالی باشد.

حکم: ارزش تاثیرگذاری نودها در پایان مرحله $t+۱$ برابر مجموع تاثیرگذاری مسیرهای $t+۱$ یالی است.

می‌دانیم که در پایان مرحله $t+۱$ام الگوریتم، داریم:

$$Inf^{t+۱}(v_i) = \sum_{j \in Neighbor(i)} w_{ij} \times Inf^{t}(v_j \to v_i) + ۱$$

می خواهیم ثابت کنیم که این مقدار برابر است با مجموع تاثیرگذاری مسیرهای حداکثر $t+۱$ یالی از v_i. بنابراین خواهیم داشت:

جمع مسیرهای حداکثر $t+۱$ یالی از v_i = مسیرهای حداکثر $t+۱$ یالی $(v_i \to v_{j_۱} \to \cdots) + \cdots +$ مسیرهای حداکثر $t+۱$ یالی $(v_i \to v_{j_l})$+ مسیر $v_i \to v_i$

که در آن $l = |Neighbor(v_i)|$ و $v_{j_۱}$ تا v_{j_l} همسایه‌های نود v_i می‌باشند. اما می‌دانیم که:

مجموع تاثیرگذاری مسیرهای حداکثر $t+۱$ یالی $(v_i \to v_{j_k} \to \cdots)$ = مجموع تاثیرگذاری مسیرهای حداکثر t یالی از v_{j_k} که از v_i نمی‌گذرند $\times w_{ij_k} = w_{ij_k} \times Inf^{t}(v_{j_k} \to v_i)$

بنابراین :

جمع مسیرهای حداکثر $t+1$ یالی از v_i =
$Inf^{t+1}(v_i) = w_{ij_1} \times Inf^t(v_{j_1} \to v_i) + \cdots + w_{ij_l} \times Inf^t(v_{j_l} \to v_i) + 1$

بنابراین نتیجه می‌گیریم که ارزش تاثیرگذاری نودها در مرحله $t+1$ام برابر مجموع تاثیرگذاری از مسیرهای حداکثر $t+1$ یالی است.
بهمین ترتیب برای ارزش ارسالی نودها به همسایه‌هایشان داریم:

مجموع تاثیرگذاری مسیرهای حداکثر $t+1$ یالی که از v_i می‌گذرند ولی از v_{j_k} نمی‌گذرند = مجموع مسیرهای حداکثر $t+1$ یالی $(v_i \to v_{j_1} \to \cdots)$ + $\cdots$ + مسیرهای حداکثر $t+1$ یالی $(v_i \to v_{j_{k-1}} \to \cdots)$ + مسیرهای حداکثر $t+1$ یالی $(v_i \to v_{j_{k+1}} \to \cdots)$ + مسیرهای حداکثر $t+1$ یالی $(v_i \to v_{j_l} \to \cdots)$ + ۱

که طبق قسمت بالا برابر است با:
$Inf^{t+1}(v_i) - w_{ij_k} \times Inf^t(v_{j_k} \to v_i) = Inf^{t+1}(v_i \to v_{j_k})$

بنابراین طبق موارد اشاره‌شده ثابت می‌شود که $Inf^{t+1}(v_i \to v_j)$ نیز برابر مجموع تاثیرگذاری مسیرهای حداکثر $t+1$ یالی است که از v_i آغاز می‌شوند ولی در دومین گام به v_j نمی‌رسند.
بدین ترتیب حکم ثابت است.
□

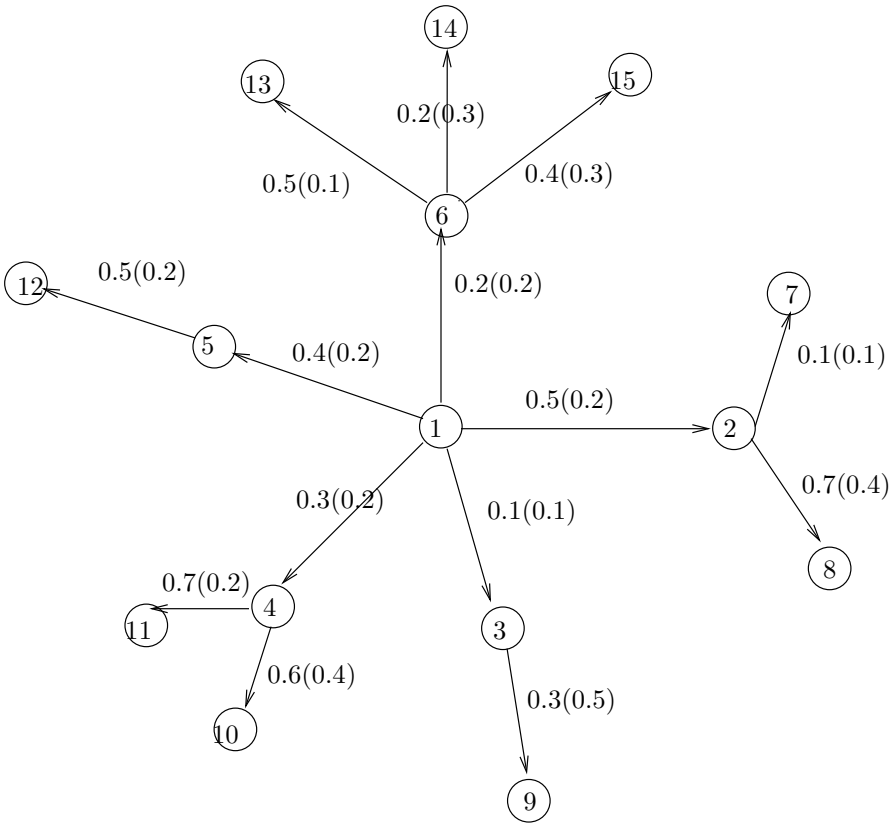
قضیه ۲.۴ تعداد مراحل لازم برای اجرای الگوریتم برابر است با قطر شبکه اجتماعی. (همانطور که می‌دانیم قطر شبکه اجتماعی به معنای بزرگترین فاصله جهتدار میان هر دو نود دلخواه در شبکه است و فاصله میان دو نود برابر کوتاهترین مسیر جهتدار از اولین نود به دومی است.)
اثبات. قضیه پیشین نشان داد که در پایان مرحله tام الگوریتم، ارزش تاثیرگذاری هر نود برابر مجموع تاثیرگذاری آن نود بر سایر نودهاست به وسیله مسیرهایی که حداکثر $t+1$ یال دارد. بنابراین واضح است که ارزش حقیقی هر نود به طول بزرگترین مسیر درون شبکه بستگی دارد که همان قطر شبکه است. پس اگر الگوریتم به تعداد یال‌های قطر شبکه اجرا شود، ارزش حقیقی نودها به دست می‌آید و پس از آن تغییر نخواهد کرد.
□

در تحقیقاتی که در علوم جامعه‌شناسی انجام شده است، محققین به این حقیقت رسیده‌اند که هر دو فرد در دنیا حداکثر با ۶ واسطه همدیگر را می‌شناسند. بنابراین در شبکه‌های اجتماعی بزرگ و کاملی که موردنظر ماست، می‌توان با تکرار الگوریتم در حدود ۶ مرحله به نتیجه ثابت و درست ارزش‌های نودها دست یافت.

مثال زیر چگونگی عملکرد الگوریتم در چند مرحله اول را نشان می‌دهد.

مثال ۳.۴ گراف شکل ۴-۴ را در نظر بگیرید. برای این شبکه الگوریتم را در چند مرحله ابتدایی اجرا می‌کنیم. توجه کنید که اعدادی که روی یال‌ها نوشته شده است، وزن یال جهتدار نشان داده شده بوده و اعداد درون پرانتزها وزن یال‌های معکوس را نشان می‌دهد. به عنوان مثال:

$$w(۲,۱) = ۰.۲ \ , \ w(۱,۲) = ۰.۵$$



شکلِ ۴-۴: شبکه نودی مثال اجرای الگوریتم محاسبه ارزش‌ها

مرحله ۰:

$$val^{۰}(i) = ۱$$
$$val^{۰}(i \to j) = ۱$$

مرحله ۱:

ارزش‌های درونی نودها

$$val^{۱}(۱) = ۰.۵ \times ۱ + ۰.۱ \times ۱ + ۰.۳ \times ۱ + ۰.۴ \times ۱ + ۰.۲ \times ۱ + ۱ = ۲.۵$$
$$val^{۱}(۲) = ۰.۲ \times ۱ + ۰.۱ \times ۱ + ۰.۷ \times ۱ + ۱ = ۲$$
$$val^{۱}(۳) = ۰.۱ + ۰.۳ + ۱ = ۱.۴$$
$$val^{۱}(۴) = ۰.۲ + ۰.۶ + ۰.۷ + ۱ = ۲.۵$$
$$val^{۱}(۵) = ۰.۲ + ۰.۵ + ۱ = ۱.۷$$
$$val^{۱}(۶) = ۰.۲ + ۰.۵ + ۰.۲ + ۰.۴ + ۱ = ۲.۳$$
$$val^{۱}(۷) = ۰.۱ + ۱ = ۱.۱$$
$$val^{۱}(۸) = ۰.۴ + ۱ = ۱.۴$$
$$val^{۱}(۹) = ۰.۵ + ۱ = ۱.۵$$
$$val^{۱}(۱۰) = ۰.۴ + ۱ = ۱.۴$$
$$val^{۱}(۱۱) = ۰.۲ + ۱ = ۱.۲$$
$$val^{۱}(۱۲) = ۰.۲ + ۱ = ۱.۲$$
$$val^{۱}(۱۳) = ۰.۱ + ۱ = ۱.۱$$
$$val^{۱}(۱۴) = ۰.۳ + ۱ = ۱.۳$$
$$val^{۱}(۱۵) = ۰.۳ + ۱ = ۱.۳$$

مقادیر ارسالی به همسایگان

$$val^{۱}(۱ \to ۲) = ۲.۵ - ۰.۵ \times ۱ = ۲$$
$$val^{۱}(۱ \to ۳) = ۲.۵ - ۰.۱ \times ۱ = ۲.۴$$
$$val^{۱}(۱ \to ۴) = ۲.۵ - ۰.۳ = ۲.۲$$
$$val^{۱}(۱ \to ۵) = ۲.۵ - ۰.۴ = ۲.۱$$
$$val^{۱}(۱ \to ۶) = ۲.۵ - ۰.۲ = ۲.۳$$
$$val^{۱}(۲ \to ۱) = ۲ - ۰.۲ = ۱.۸$$
$$val^{۱}(۲ \to ۷) = ۲ - ۰.۱ = ۱.۹$$
$$\dots$$

به همین ترتیب الگوریتم اجرا می‌شود و مقادیر محاسبه می‌شوند. به عنوان مثال در مرحله ۲ چند مقدار را در زیر آورده‌ایم.

مرحله ۲:

$$val^{۲}(۱) = ۰.۵ \times ۱.۸ + ۰.۱ \times ۱.۳ + ۰.۳ \times ۲.۳ + ۰.۴ \times ۱.۵ + ۰.۲ \times ۲.۱ + ۱ = ۳.۷۴$$
$$val^{۲}(۱ \to ۲) = ۳.۷۴ - ۰.۹ = ۲.۸۴$$
$$\dots$$

۴-۳-۳ پیچیدگی محاسباتی الگوریتم

گفتیم که الگوریتم در k مرحله انجام می‌شود که در هر مرحله هر نود اطلاعات را از همسایه‌های خود می‌گیرد و ارزش تاثیرگذاری خود را با جمع این مقادیر به دست می‌آورد. سپس به هر یک از همسایه‌های خود، میزان تاثیرگذاری آن همسایه را بر کل شبکه از طریق خود، $val(i \to j)$ اعلام می‌کند. واضح است که زمان اجرای هر مرحله به تعداد همسایگان نودها بستگی دارد. اگر بیشینه تعداد همسایگان هر نود برابر α باشد، آنگاه زمان اجرای هر مرحله $O(\alpha)$ و زمان اجرای کل الگوریتم برابر $O(k\alpha)$ خواهد بود. بدیهی است که در این روش فرض کرده‌ایم که نودها خود در مسابقه شرکت می‌کنند و توان محاسباتی دارند. همین الگوریتم را می‌توان با هزینه کردن زمان بیشتر به راحتی به صورت سریال و متمرکز اجرا نمود. بدین ترتیب که در هر مرحله برای تمامی نودها یک به یک ارزش تاثیرگذاری مرحله جدید محاسبه می‌شود و بعلاوه ارزش ارسالی به همسایگان آن نودها نیز محاسبه می‌شود و این ارزش‌ها در مرحله بعد برای محاسبات نودها مورد استفاده قرار می‌گیرد. بدین ترتیب اگر میانگین تعداد همسایه‌ها برای هر نود برابر β باشد، هر مرحله به اندازه $O(n\beta)$ طول می‌کشد و زمان کل اجرای الگوریتم برابر $O(nk\beta)$ خواهد بود.

پیشتر اشاره کردیم که k قطر شبکه اجتماعی را نشان می‌دهد. با در نظر گرفتن β به عنوان میانگین تعداد همسایه‌ها داریم:

$$\beta^k = n \Rightarrow k = \log_\beta n$$

به عنوان نمونه در شبکه اجتماعی زیر داریم:

$$n \simeq ۱۰۰۰۰۰۰۰, \beta = ۵۰ \Rightarrow k = log_{۵۰}۱۰'۰۰۰'۰۰۰ = ۴.۱۲$$

بنابراین مشخص است که تخمین قبلی $k = ۶$ در این شرایط نیز کارا است. طبق روش بالا زمانیکه $\beta = ۱۵$، $k \simeq ۶$ خواهد بود.

۴-۴ طراحی مکانیزم راستگو

یکی از مهمترین ویژگی‌هایی که برای الگوریتم نودهای تاثیرگذار بایستی تضمین کنیم، راستگویی است. بدین منظور نگاهی اجمالی به الگوریتم ارائه شده فعلی می‌اندازیم.

الگوریتم تعیین نودهای تاثیرگذار در حضور اطلاعات ناقص از دو فاز تشکیل شده است که در آن فاز اول m مرحله (به صورت دقیق به اندازه قطر شبکه و در حالت آماری ۶ مرحله) داراست.

در فاز اول در هر مرحله هر نود ارزش تاثیرگذاری ارسالی ناقص همسایگانش را گرفته ($val(i \to j)$) و با ضرب کردن در وزن یال‌های میانی، مجموع مقادیر را به عنوان ارزش تاثیرگذاری خود اعلام می‌کند. این کار m مرحله ادامه می‌یابد. سپس همه نودها ارزش تاثیرگذاری خود رادر

نهایت به مرکز[۱۳] اعلام می‌کنند. در فاز دوم، مرکز از میان ارزش‌های اعلام شده، k نود را انتخاب می‌کند. استراتژی انتخاب این k نود، بدین علت که ارزش‌ها پیوسته‌سازی شده‌اند، می‌تواند k نود با ارزش بیشینه باشد. بدین ترتیب مرکز با الگوریتم VCG که پیشتر توضیح دادیم، k ارزش بیشینه را به عنوان برندگان معرفی نموده و هزینه پرداختی آنها را تعیین می‌کند.

اکنون به مکانیزمی می‌پردازیم تا راستگویی تضمین شود. در ابتدای بحث لازم است یادآوری کنیم که فرض بر اینست که نودها بدجنس نبوده و فقط خودخواه هستند یعنی فقط دنبال افزایش سود خود هستند ولی اگر سود آنها تغییر نکند به خرابکاری یا کاهش سود دیگران رغبتی ندارند.

در فاز دوم الگوریتم، همانطور که گفتیم از الگوریتم VCG استفاده شده است و بنابراین فاز دوم الگوریتم راستگو است. حال می‌خواهیم به روشی راستگویی قسمت اول را تضمین کنیم.

در کل ناسازگاری‌های زیر در حین اجرای فاز اول الگوریتم متصور است.

۱. نودها در مرحله نهایی پس از مشخص شدن ارزش تاثیرگذاری کلی خود در شبکه، به مرکز دروغ بگویند، یعنی مقدار اشتباهی را به مرکز اعلام کنند.

۲. در مراحل میانی فاز اول الگوریتم، نودها ارزش موقت خود را به دروغ به همسایگان اعلام کنند.

۳. نودها در مراحل میانی فاز اول مقادیر متفاوتی تحت عنوان ارزش تاثیرگذاری خود به همسایگان مختلف اعلام کنند. آنچه واضح است اینست که مقادیر ارسالی $val(i \to j)$ از نود i به نود j برای همسایگان مختلف متفاوت است و بستگی به ارزش تاثیرگذاری خود همسایه و وزن یال میانی w_{ij} دارد. اما می‌دانیم که هر همسایه j می‌تواند با در نظر گرفتن ارزش تاثیرگذاری خود و وزن یال میانی w_{ij} به سادگی ارزش تاثیرگذاری موقت نود i را به دست آورد. منظور این حالت اینست که مقادیر به گونه‌ای به همسایگان ارسال شود که اگر نودها بخواهند ارزش واقعی i را محاسبه کنند به نتایج متفاوتی دست یابند. در واقع قرار است که همسایگان مختلف روی ارزش تاثیرگذاری نود i متفق‌القول نبوده و تصورات متفاوتی از آن داشته باشند.

۴. نودهای میانی در پیام‌هایی که فقط وظیفه انتقال آنها را به عهده دارند، دستکاری کنند.

در مقالات [۲۰] و [۲۱] در خصوص طراحی مکانیزم در ساختارهای توزیع شده، بحث و بررسی شده است و نویسندگان تکنیک‌هایی بدین منظور ارائه کرده‌اند. اما نکته‌ای که پیش از آغاز کردن بحث چگونگی تضمین راستگویی در فاز اول الگوریتم بایستی بدان توجه کرد، اینست که الگوریتم فوق در هنگام نقص اطلاعات مطرح شده است و بنابراین نمی‌توان گفت که مرکز درستی ارزش‌های تاثیرگذاری ارسالی را بررسی کرده و در صورت وجود خطا جریمه‌ای به نود خاطی تخصیص دهد. در حقیقت مرکز تنها نودها را می‌شناسد، اما از ارتباط میان آنها و وزن یال‌های میانی و همسایگی نودها با هم اطلاعی در دست ندارد.

---

[13] center

۴-۴-۱ اعمال تغییرات در الگوریتم برای تضمین راستگویی

گفتیم که هر نود در طول اجرای الگوریتم می‌تواند برای افزایش سود خود دروغ بگوید. برای رفع این مشکل نقش جدیدی برای هر نود تعریف می‌کنیم. در طول فاز اول الگوریتم هر نود در دو نقش فعالیت می‌کند.

تعریف ۶.۴ دو نقش برای هر نود متصور است. نقش اول نقش اصلی است که در آن هر نود به محاسبه ارزش تاثیرگذاری خود می‌پردازد و ارزش ناقص خود را به همسایگانش اعلام می‌کند. در نقش دوم، هر نود نقش بررسی‌کننده را دارد. بدین معنا که هر نود مسئول بررسی ارزش تاثیرگذاری اعلام شده توسط همسایگانش است.

هر نود در نقش اصلی خود، مقدار ارزش تاثیرگذاری ارسالی ناقص همسایگانش را گرفته و ارزش تاثیرگذاری خود را محاسبه می‌نماید. سپس در انتها مقدار ارزش تاثیرگذاری ناقص خود را برای هر یک از همسایگان محاسبه نموده و ارسال می‌کند.

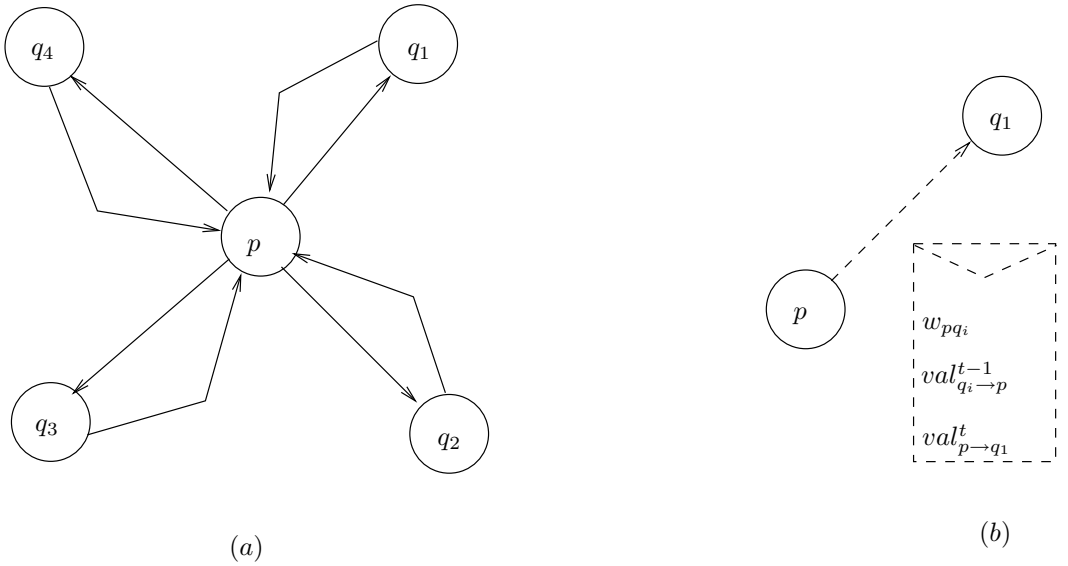
هر نود در نقش بررسی کننده، مقدار ارزش تاثیرگذاری ارسالی ناقص همسایگانش را گرفته و صحت آن‌ها را بررسی می‌کند. نودهای بررسی کننده نود اصلی p برای انجام محاسبات خود به اطلاعات همسایگان نود p نیاز دارند. یادآوری می‌کنیم که نودها و حتی نود مرکزی نیز به اطلاعات کامل شبکه دسترسی ندارند. بلکه هر نود فقط همسایه‌های خود را شناخته و وزن یال‌های متصل به خود را می‌داند. بنابراین برای اینکه بررسی کننده‌های نود اصلی p بتوانند صحت اطلاعات را تایید کنند، نیاز است که خود نود p اطلاعات همسایگانش را به آنها ارائه دهد. یکی از اطلاعات لازم، همسایه‌های نود p و وزن یال‌های متصل است. این مقادیر در طول اجرای الگوریتم ثابت است. بنابراین کافیست که هر نود در ابتدای اجرای الگوریتم یکبار این اطلاعات را به همسایگانش ارسال کند.

در مراحل بعد نودهای بررسی‌کننده نود اصلی p به مقادیر ارزش تاثیرگذاری ناقص همسایگان نود p برای انجام محاسبات نیاز دارند. بنابراین در مرحله t نود اصلی p بایستی علاوه بر ارزش تاثیرگذاری ناقص خود در مرحله t، ارزش تاثیرگذاری ناقص ارسالی همسایگانش به خود را نیز به نودهای بررسی‌کننده ارسال نماید.

بررسی‌کننده‌های نود p پس از دریافت اطلاعات لازم، ۲ مقدار را بررسی می‌کنند. گفتیم که نود p مقدار ارزش تاثیرگذاری ناقص همسایگانش در مرحله قبل را نیز در بسته اطلاعاتی ارسال می‌کند. هر نود بررسی‌کننده ابتدا صحت ارزش تاثیرگذاری ناقصی را که به عنوان ارزش خودش توسط p معرفی شده بررسی می‌کند. سپس به بررسی صحت ارزش تاثیرگذاری اعلام‌شده نود p می‌پردازد. برای روشن شدن مطلب مثال زیر را در نظر بگیرید.

مثال ۴.۴ در این مثال نحوه عملکرد نود اصلی و نودهای بررسی‌کننده در شبکه ساده شکل ۴-۵ نشان داده شده است. این مثال یک برش زمانی‌مکانی از الگوریتم را نشان می‌دهد که در آن نود p به

عنوان نود اصلی ایفای نقش می‌کند و نودهای q به عنوان نودهای بررسی‌کننده عمل می‌کنند.



(a) (b)

شکلِ ۴–۵: نحوه عملکرد نودهای بررسی‌کننده نود اصلی p. (a). شبکه اجتماعی ساده برای بررسی نحوه عملکرد نودهای اصلی و بررسی‌کننده (b). بسته اطلاعاتی ارسالی از نود p به همسایه q_1

طبق قسمت (b) شکل ۴–۵ نود p در ابتدای الگوریتم اطلاعات w_{pq_i} را به همسایگان خود می‌فرستد و در ادامه در هر مرحله t، نود p به همسایه q_k خود، اطلاعات $val^{t}_{p \to q_k}$ و $val^{t-1}_{q_i \to p}$ را که در آن $i = 1, 2, 3, 4$ است، ارسال می‌کند. همسایه q_k ابتدا مقدار $val^{t-1}_{q_k \to p}$ را که توسط p ارسال شده با مقداری که خود ذخیره دارد، مقایسه می‌کند و سپس خود به محاسبه $val^{t}_{p \to q_k} = \sum_{i \neq k} w_{pq_i} val^{t-1}(q_i \to p)$ می‌پردازد و این مقدار را با مقدار ارسالی نود p مقایسه می‌کند. در هر یک از این دو مقایسه اگر اختلافی وجود داشت، آن را به مرکز اطلاع می‌دهد. بدین روش از تخلف نود p (ناسازگاری ۲) جلوگیری می‌شود. اما همچنان می‌توان با یک نوع ناسازگاری روبرو بود که نتیجه عملکرد ناسازگاری نوع ۳ است.

این ناسازگاری بدین شکل بوجود می‌آید که نود اصلی بسته‌های اطلاعاتی متفاوتی را به همسایگان بررسی‌کننده مختلف می‌فرستد و با ایجاد این ناهماهنگی میان بسته‌ها باعث می‌شود که تخلفات کشف نشود. به عنوان نمونه در مثال بالا، فرض کنید که ارزش تاثیرگذاری نود p در مرحله t برابر $val^{t}(p)$ باشد ولی نود p می‌خواهد ارزش نادرست $val'^{t}(p)$ را به عنوان ارزش تاثیرگذاری خود در شبکه جا بیندازد. نود p در بسته‌های ارسالی مقدار $val'^{t}(p)$ را به عنوان ارزش خود اعلان می‌کند و به دستکاری ارزش ناقص ارسالی همسایگانش می‌پردازد. یعنی نود p می‌تواند به راحتی به همه همسایه‌ها به جز یکی از آنها، فرض کنید همسایه بررسی‌کننده q_k، بسته اطلاعاتی صحیح را با یک تغییر در مقدار ارزش تاثیرگذاری ناقص q_k بفرستد. به عبارت دیگر نود p مقدار صحیح $val^{t-1}(q_i \to p)$ را برای $i \neq k$ به همسایگان خود غیر از q_k می‌فرستد و بجای مقدار $val^{t-1}(q_k \to p)$،

مقدار $val'^{t-1}(q_k \to p) = val^{t-1}(q_k \to p) + \frac{val'^t(p) - val^t(p)}{w_{pq_k}}$ را به آنها ارسال می‌کند. بعلاوه به همسایه q_k بسته اطلاعاتی صحیح را با تغییری در ارزش تاثیرگذاری یکی دیگر از همسایگانش می‌فرستد. به عنوان مثال فرض کنید که q_m همسایه دیگری از p باشد. بنابراین نود p بسته اطلاعاتی شامل $val^{t-1}(q_i \to p)$ که $i \neq m$ را به عنوان ارزش تاثیرگذاری نودهای q_i به نود بررسی‌کننده q_k می‌فرستد و مقدار $\frac{val'^{t-1}(q_k \to p) \times w_{pq_k} - val^{t-1}(q_k \to p) \times wpq_k + val^{t-1}(q_m) \times w_{pq_m}}{w_{pq_m}}$ را به عنوان ارزش تاثیرگذاری ناقص نود q_m به آن اضافه می‌کند.

بدین روش هر یک از نودهای بررسی‌کننده، مشاهده می‌کند که مقدار ارزش تاثیرگذاری ناقص خودش در بسته به درستی اعلام شده و سپس نتیجه محاسباتش برای ارزش تاثیرگذاری p برابر $val'^t(p)$ خواهد شد که برابر مقدار اعلانی نود p به شبکه است. بدین ترتیب نود p می‌تواند مقدار اشتباهی را به نودهای بررسی‌کننده بقبولاند.

برای حل این مشکل و بنابراین ناسازگاری ۳، بایستی به ترتیبی عمل کنیم که نودها فقط یک بسته ارسال کنند و این بسته میان همسایه‌ها پخش شود. این کار به ۲ صورت امکان‌پذیر است. در روش اول هر نود بسته اطلاعاتی خود را به مرکز ارسال می‌کند و مرکز میان همسایه‌ها پخش می‌کند. (توجه آنکه مرکز با خواندن اطلاعات بسته، می‌تواند این همسایه‌ها را بیابد.) اشکال این روش بار زیاد روی مرکز است که الگوریتم را به شدت کند می‌کند. در روش دوم، یکی از همسایه‌ها مسئولیت پخش بسته‌های ارسالی را برعهده می‌گیرد. بدین ترتیب نود p بسته‌های اطلاعاتی خود را به آن همسایه مشخص ارسال می‌کند و آن همسایه، این بسته را میان سایر همسایه‌ها پخش می‌کند. برای اینکه همسایه‌ها خود در داده‌ها دستکاری نکنند، نیاز است که بسته توسط نود p امضا شود. اشکالاتی ممکن است در طول اجرای این الگوریتم به وجود آید. به عنوان مثال ممکن است که به یک همسایه بسته اطلاعاتی نرسد یا اینکه بسته درستی به دست آن نود نرسد. حتی ممکن است که ۲ بسته به یک نود برسد. در هر یک از این حالات نودی که خطا را تشخیص می‌دهد آن را به مرکز اطلاع می‌دهد. در حقیقت این اشکالات می‌تواند توسط نود اصلی، نود پخش کننده یا نود دریافت کننده نهایی حاصل شده باشد. یعنی در مورد اشکال اول ممکن است که نود اصلی همسایه‌هایش را به درستی معرفی نکرده باشد و یا نود پخش کننده به برخی از همسایه‌های معرفی‌شده، بسته را نرسانده باشد. در حالت دوم ممکن است نود پخش کننده در بسته دستکاری کرده باشد و در حالت سوم ممکن است که نود اصلی بسته‌های متفاوتی به نودهای مختلف ارسال کرده باشد. یعنی دو بسته با امضای p به ۲ نود پخش کننده ارسال شده باشد و آنها این ۲ بسته را میان همه همسایگان پخش کرده باشند. بعلاوه ممکن است که هیچیک از این اتفاقات رخ نداده و نود دریافت کننده نهایی به دروغ اعلام خطا کرده باشد. در اینجا فرض می‌کنیم که لینک‌های میان نودها خطا ندارند و داده را به درستی انتقال می‌دهند. می‌توان در مواردی که لینک‌ها تاحدی خطاپذیرند از تاییدیه[۱۴] دریافت کننده به ارسال‌کننده استفاده کرد. بدین صورت که اگر ارسال کننده پس از گذشت زمانی تاییدیه دریافت نکرد، دوباره به ارسال مجدد بسته روی آورد و این کار را تا حد مشخصی، مثلاً ۳ بار، ادامه دهد. بدین روش می‌توان فرض کرد که لینک‌های میانی عاری از خطا هستند و فقط نودها هستند که عمداً باعث بروز خطا می‌شوند.

[14] ack

در عمل معمولاً در الگوریتم‌های مختلف در صورت بروز چنین خطاهایی نود مرکزی جریمه‌ای نقدی برای نودهای خاطی یا کل نودهای شبکه اختصاص می‌دهد. اما ما در این الگوریتم به دلایلی که در ادامه خواهیم دید، ترجیح می‌دهیم به گونه‌ای دیگر عمل کنیم. در این روش مرکز در هر کجای فاز اول الگوریتم که پیغام خطا دریافت کرد، دستور شروع دوباره کار از ابتدا را صادر می‌کند و اینگونه الگوریتم تا هنگامیکه خطایی گزارش شود به فاز دوم که فاز اختصاص کالا یا استراتژی جدید به برندگان است، وارد نمی‌شود. مشخص است که این مکانیزم باعث می‌شود که نودها در عملیات خود خطا نکنند، زیرا با مکانیزم‌های مطرح شده، این خطاها تشخیص داده می‌شود و بنابراین الگوریتم بایستی از ابتدا آغاز شود. چون در ابتدا فرض کردیم که نودها بدجنس نیستند و فقط خودخواه بوده و می‌خواهند سود خود را افزایش دهند، این شروع‌های دوباره به نفع آنها نیست و این باعث می‌شود که از اعمال خطا اجتناب کنند.

سوالی که در این قسمت پیش می‌آید اینست که شاید نودی که در یک مرحله از الگوریتم خطایی را کشف می‌کند برای جلوگیری از ضرر از گزارش آن به مرکز اجتناب کند و راستگویی الگوریتم را زیر سوال ببرد. جواب این سوال، علت اتخاذ استراتژی آغاز دوباره مقابل اعمال جریمه را روشن می‌کند. اگر از مکانیزم جریمه استفاده می‌کردیم، این امر ممکن بود زیرا نودها پس از گزارش خطا مجبور به پرداخت جریمه بودند که از سود آنها می‌کاست. این امر خود را هنگامی بیشتر مشخص می‌کند که میزان اختلاف سودی که به ضرر نود کشف کننده خطا از خطای ایجاد شده می‌شد از میزان جریمه کمتر می‌بود. در اینحالت آن نود ترجیح می‌دهد که خطا را اعلام نکند. اما در حالت آغاز دوباره الگوریتم، مشخص است که با گزارش خطای صورت گرفته سود نود گزارش دهنده کم نمی‌شود. بلکه اگر نود گزارش ندهد باعث می‌شود که نود خاطی سود بالاتری برده و از سود نود کشف‌کننده خطا بکاهد. این باعث می‌شود که نودها در صورت کشف خطا متمایل باشند که هرچه سریعتر نسبت به گزارش خطا اقدام کنند.

در انتهای الگوریتم لازم است که مقدار ارزش تاثیرگذاری نهایی تمام نودها به نود مرکزی برای تصمیم‌گیری پایانی و اجرای الگوریتم VCG مطرح شده، ارسال شود. واضح است که اگر خود نودها مسئولیت ارسال این مقدار را داشته باشند، می‌توانند در آن دستکاری کنند. بنابراین در این قسمت نیز از همسایه پخش کننده بهره می‌گیریم. در این قسمت همسایه‌های پخش‌کننده وظیفه دارند که بسته‌های امضاشده نودهای اصلی را که حاوی مقدار نهایی ارزش تاثیرگذاری آنها در شبکه است به نود مرکزی ارسال کنند. بدیهی است که نودهای پخش‌کننده نمی‌توانند در بسته‌ها دستکاری کنند، چون بسته‌ها دارای امضا هستند. نود مرکزی بسته حاوی ارزش نود p را هنگامی قبول می‌کند که از طریق نود همسایه آن به او رسیده باشد. توضیح آنکه اگر میان نود مرکزی و سایر نودها ارتباط مستقیم برقرار نباشد، بایستی بسته‌ها توسط نودهای پخش کننده نیز در انتها امضا شده باشند تا برای مرکز درستی ارسال از این طریق اثبات شود. بنابراین در این حالت بسته‌ها دو امضا خواهند داشت. امضای نود اصلی و امضای نود همسایه آن که نقش بررسی‌کننده و پخش کننده را بر عهده دارد. نود مرکزی پس از دریافت تمام این بسته‌ها وارد فاز دوم می‌شود و همانطور که در ابتدای این فصل گفته شد نودهای برنده را تعیین می‌کنند. با تغییر الگوریتم در فاز اول برای تضمین راستگویی، عملاً از امکان اعلام ارزش نودها توسط خودشان به مرکز جلوگیری

کردیم، بنابراین راستگویی فاز دوم الگوریتم نیز پیشاپیش به همین روش تضمین می‌شود. در نتیجه، مرکز پس از تعیین نودهای تاثیرگذار، می‌تواند هر استراتژی دلخواهی را برای تعیین هزینه پرداختی هر نود انتخاب کند. به عبارت دیگر، مرکز می‌تواند به همه نودها تخفیف یکسانی بدهد، به عنوان مثال کالا را به همه رایگان بدهد و یا برای نودهای مختلف، تخفیف‌های مختلف و بنابراین هزینه‌های متفاوتی تعیین کند. صرف نظر از استراتژی اتخاذی مرکز در فاز دوم، راستگویی توسط مکانیزم اعمالی در فاز اول تضمین شده است. بدین ترتیب مباحث بالا اثباتی است برای قضیه مهم زیر.

قضیه ۳.۴ الگوریتم طراحی شده برای تعیین تاثیرگذاری‌ها و تشخیص نودهای تاثیرگذار در حضور نقص اطلاعات، یک مکانیزم راستگوست.

جمع‌بندی الگوریتم

الگوریتم یافتن k نود تاثیرگذار در یک شبکه اجتماعی در حضور اطلاعات ناقص از ۲ فاز تشکیل شده است. فاز اول از m مرحله ساخته می‌شود که m قطر شبکه را نشان می‌دهد و بصورت آماری کمتر از ۱۰ می‌باشد. در مرحله ۰ اجرای الگوریتم مقدار ارزش تاثیرگذاری کامل و ناقص هر نود برابر ۱ است. هر نود p بسته‌ای حاوی اطلاعات همسایگان و وزن یال‌های متصل به خود را به یکی از همسایه‌هایش که نقش نود پخش‌کننده p را برعهده دارد، ارسال می‌کند. نودهای پخش‌کننده همسایه‌های نود اصلی را از روی بسته شناخته و بسته را به آنها ارسال می‌کنند. در مراحل بعدی $t > 0$ هر نود اصلی p از روی مقادیر ارزش تاثیرگذاری ناقصی که از همسایگانش دریافت کرده، ارزش تاثیرگذاری کامل خود را در آن مرحله تعیین می‌کند و این مقدار تاثیرگذاری را به همراه مقادیر تاثیرگذاری ارسالی همسایگانش در مرحله قبل در یک بسته قرار داده و در اختیار نود پخش‌کننده قرار می‌دهد. نود پخش‌کننده نیز این بسته را میان سایر همسایگان پخش می‌کند. سپس هر همسایه پس از دریافت این اطلاعات ابتدا نقش بررسی‌کننده را بازی می‌کند و در گام اول مقدار ارزش تاثیرگذاری خود را که در بسته اعلام شده با مقدار واقعی که خود در اختیار دارد مقایسه می‌کند و سپس به محاسبه مقدار ارزش تاثیرگذاری نود p پرداخته و با محتوای بسته مقایسه می‌کند. در صورت بروز خطا، به مرکز اطلاع داده می‌شود. سپس نودها نقش اصلی را بازی می‌کنند و ارزش تاثیرگذاری خود را در مرحله جدید محاسبه می‌نماید. در مرحله نهایی، نودهای پخش‌کننده پس از بررسی صحت بسته‌هایی که دریافت کرده‌اند، آنها را امضا کرده و به مرکز می‌فرستند.

در طول اجرای فاز اول هرگاه مرکز پیغام خطایی دریافت کند، دستور آغاز این فاز را صادر می‌کند. اگر مرکز بسته‌های حاوی ارزش تاثیرگذاری نهایی تمام نودها را بدون دریافت پیغام خطا دریافت کرد، وارد فاز دوم الگوریتم می‌شود. در فاز دوم، مرکز k نودی را که ارزش بیشینه دارند، به عنوان برنده اعلام می‌کند و کالا یا استراتژی جدید را به قیمت $\frac{\max_{k+1} val}{\alpha}$ که $\alpha = \frac{maxval}{price}$ است، به برندگان می‌دهد.

فصلِ ۵

نتیجه‌گیری و ایده‌های نوین

۵-۱ خلاصه‌ی کار

همانطور که تاکنون متوجه شده‌ایم، یکی از مهمترین مسائلی که امروزه برای اقتصاددانان و تئوری پردازان علوم کامپیوتر و نظریه بازی‌ها حائز اهمیت است، مساله بازاریابی ویروسی و بکارگیری شبکه‌های اجتماعی در آن است. بارزترین مساله این حوزه، پیداکردن k نود تاثیرگذار در یک شبکه از افراد است تا توسط این عده و با هزینه ثابت بتوان از تاثیرگذاری این افراد در کل شبکه بهره گرفت و یک نوآوری را (کالا یا استراتژی یا ...) در شبکه به بیشترین حد ممکن انتشار داد.

در این رساله ما برای افزایش سرعت حل مساله و افرایش انتشار موج نوآوری در شبکه مساله را جامع‌تر کرده و تکنیک‌های نوینی مطرح نمودیم. در حقیقت مساله ما تبدیل شد به مساله پیداکردن m گروه تاثیرگذار در یک شبکه از افراد و ما تلاش کردیم تا با تکنیک‌های جدید و سازگار با مدل مساله کل شبکه را خوشه‌بندی کنیم. بعلاوه تکنیک خوشه‌بندی را به گونه‌ای ارائه دادیم تا افراد همانند دنیای واقعی بتوانند عضو گروه‌های متعددی باشند. در انتها نیز نشان دادیم که تکنیک ما نسبت به روش $Hill-Climbing$ که متداول‌ترین روش حل مساله k نود تاثیرگذار است در زمان کمتر، نتایج بهتری ارائه می‌دهد.

در فصل ۴ مساله دیگری را مطرح کردیم و آن مساله جستجوی k نود تاثیرگذار هنگام نقص اطلاعات بود. در این حالت فرض کردیم که هر نود دارای یک سری اطلاعات خصوصی شامل دوستان و میزان تاثیرگذاری میان آنهاست و مرکزی که این اطلاعات را در اختیار داشته باشد، وجود ندارد. بعلاوه اشاره کردیم که جمع‌آوری این اطلاعات برای یک مرکز بسیار دشوار، زمان‌گیر و پرهزینه است و در بسیاری از موارد ممکن است به علت عدم همکاری اعضای شبکه غیرممکن باشد و یا اطلاعات نادرستی به مرکز گزارش شود. در چنین ساختاری، روشی ارائه کردیم که بدون نیاز به جمع‌آوری این اطلاعات، مساله را حل کند و نودهای تاثیرگذار را ارائه دهد. این روش به علت دخیل کردن تمام نودها در محاسبات در زمان بسیار کوتاهی اجرا می‌شود. بعلاوه چون محاسبات درون نودها از سادگی خاصی برخوردار بود، نیاز به پیچیده‌بودن نودها نداشتیم و هر نود

با اندکی حافظه و یک پردازنده فوق‌العاده ساده می‌توانست در محاسبات شرکت کند. همچنین ۲ نقش بررسی‌کننده و اصلی برای نودها تعریف کردیم و با بهره‌گیری از آن راستگویی مکانیزم توزیع شده خود را تضمین نمودیم.

در این حوزه از علوم به علت اهمیت فوق‌العاده و جدید بودن مسائل، می‌توان کارهای درخور توجهی در آینده انجام داد. ایده‌هایی برای برخی از این کارها در زیر مطرح کرده‌ایم که امید است در تحقیقات آتی به آن‌ها بپردازیم.

۵-۲ پردازش موازی در انتشار گروه محور

پیچیدگی مساله و سختی آن از نظر زمان اجرا ما را به آن سو سوق می‌دهد که از تکنیک‌های پردازش موازی بهره گیریم، اما وجود ارتباطات نامنظم میان رئوس و تاثیرگذاری آنها بر یکدیگر ما را الزام می‌کند که در هنگام تحلیل و اجرای الگوریتم‌ها کل دانش را یکجا داشته باشیم و مساله را یکجا حل کنیم. بنابراین این محدودیت‌ها باعث می‌شود که ما نتوانیم مساله را به زیرمساله‌هایی تقسیم کرده و هرکدام را به صورت مجزا حل کنیم. اما با بهره‌گیری از تکنیک گروه‌بندی این خواسته محقق خواهد شد. همانطور که در فصل ۲ گفتیم هدف ما این بود که گروه‌هایی تشخیص دهیم که شباهت رئوس درون آنها بسیار زیاد و شباهت رئوس میان دو گروه مختلف بسیار کم باشد. این امر ما را قادر می‌کند که الگوریتم‌ها را برای رئوس درون هر گروه مستقل از سایرین اجرا کنیم. بنابراین می‌توان گروه‌های مختلف را در زیرمساله‌های متفاوتی قرار داد و هر زیرمساله را در پردازنده مجزایی حل کرد بدون آنکه نیازی به اطلاعات سایر گروه‌ها و رئوس داشته باشیم.

در حقیقت در این حالت بر این اساس عمل می‌کنیم که کل شبکه را به تعداد محدود و کم گروه تقسیم می‌کنیم و هر گروه را به یک پردازنده اختصاص می‌دهیم. در هر گروه به k نفر که ممکن است برای گروه‌های مختلف، متفاوت باشد تبلیغ می‌کنیم. در این حالت روابط بین گروه‌ها را نادیده می‌گیریم. استراتژی جدید با استفاده از روابط k نفر انتخاب شده در هر گروه گسترش می‌یابد. گروه‌ها در این روش می‌توانند مجزا[1] باشند و یا اینکه به نودهای مشترک میان گروه‌ها وزن بالاتری داده شود. بدین ترتیب وظیفه هر پردازنده یافتن نودهای تاثیرگذار در گروه تخصیص داده شده است.

۵-۳ روش ترکیبی در الگوریتم انتشار گروه محور

یک روش مفید در حل مساله تحلیل انتشار، استفاده از ایده ترکیبی است. در این روش ما از هر دو تکنیک گروه‌بندی در فصل ۲ و تکنیک پردازش موازی که در بخش ۵-۲ مطرح کردیم، بهره

[1] disjoint

می‌گیریم. آنچنانکه پیش از این گفتیم در تکنیک گروه‌بندی مجموعه‌ای از گروه‌ها را در نظر گرفته و در آنها تبلیغ می‌کنیم. در هر گروه پس از تبلیغ درصدی از افراد نوآوری را می‌پذیرند و به سایرین تبلیغ می‌کنند. گفتیم که یکی از ویژگی‌های مهم و مثبت این تکنیک، تبلیغات گروهی بود. اما با این وجود این احتمال وجود دارد که بخواهیم در مواردی خاص تبلیغات فردی داشته باشیم که خارج از ساختار و اساس الگوریتم انتشار گروه محور است.

در حالت پردازش موازی نیز فرض می‌کردیم که تعداد گروه‌ها کم است. در هر گروه به k نفر تبلیغ می‌کردیم و از یالهای میان گروهی صرف نظر می‌نمودیم. در روش ترکیبی می‌توانیم هم اشکال تکنیک موازی سازی را رفع کنیم و هم در صورت لزوم تبلیغات فردی داشته باشیم.

طبق روش گروه‌بندی، شبکه را به گروه‌هایی تقسیم کرده و با اجرای الگوریتم انتشار گروه‌های تاثیرگذار را مشخص می‌نماییم. سپس هر گروه را به یکی از پردازنده‌ها اختصاص داده و با روش موازی به دنبال k_i نود تاثیرگذار در هر کدام از این گروه‌ها می‌پردازیم. به این روش در انتها مجموعه‌ای از مهمترین نودهای تاثیرگذار مشخص خواهد شد. می‌توان مرحله انتخاب گروه‌ها را با مدل آستانه عمومی و مرحله درون گروهی برای تشخیص نودهای تاثیرگذار را با مدل آبشاری تحلیل نمود. هر چند ثابت شده که این دو مدل معادلند، اما این تفکیک مدلی در منطقی‌تر به نظر رسیدن مساله و کاهش زمان اجرا موثر است.

۵-۴ استفاده از تکنیک Hidden Markov Model در انتشار نوآوری

تکنیک Hidden Markov Model به این صورت است که در آن احتمال تولید یک زنجیره را از یک مدل خاص هنگامیکه حالت رئوس میانی بر ما پوشیده است، محاسبه می‌کنیم . می‌دانیم که در هنگام محاسبه استراتژی اخذشده نودهای مختلف، گاهی اوقات ممکن بود یک نود بارها تغییر وضعیت دهد و یا اینکه اتفاقات دیگری بیفتد که ما مجبور باشیم در عمل چندین بار گراف را پیمایش نماییم تا نتیجه نهایی را به دست آوریم. استفاده از تکنیک HMM به ما این امکان را می‌دهد که به سرعت و تنها با یکبار پیمایش گراف احتمال اتخاذ استراتژی جدید را در گراف و میزان انتشار محبوبیت آن را میان افراد به دست آوریم. بدین ترتیب در هر مرحله با یافتن احتمال اتخاذ استراتژی‌های مختلف توسط رئوس، احتمال را برای نودهای همسایه به دست می‌آوریم و در نهایت احتمال اتخاذ استراتژی جدید توسط کل گراف معین خواهد شد.

کتاب‌نامه

[1] S. Morris. *Contagion.* Review of Economic Studies, Vol. 67, 2000, pp. 57–78.

[2] D. Kempe, J. Kleinberg, and E. Tardos. *Maximizing the spread of influence in a social network.* In Proc. 9th International Conference on Knowledge Discovery and Data Mining, Washington DC, USA, 2003, pp.137–146.

[3] P. Dodds and D. Watts. *Universal behavior in a generalized model of contagion.* Physical Review Letters, Vol. 92(21): 218701, 2004.

[4] D. Kempe, J. Kleinberg, and E. Tardos. *Influential nodes in a diffusion model for social networks.* In Proc. 32th International Colloquium on Automata, Language and Programming, Lisboa, Portugal, 2005, pp. 1127–1138.

[5] P. Domingos and M. Richardson. *Mining the network value of customers.* In Proc. 7th International Conference on Knowledge Discovery and Data Mining, San Francisco, USA, 2001, pp. 57–66.

[6] G. Nemhauser, L. Wolsey, and M. Fisher. *An analysis of the approximations for maximizing submodular set functions.* Mathematical Programming, Springer Berlin, Vol. 14, pp. 265–294, 1978.

[7] M. Richardson and P. Domingos. *Mining Knowledge-Sharing Sites for Viral Merketing.* In Proc. 8th International Conference on Knowledge Discovery and Data Mining, 2002, pp. 61–70.

[8] N. Immorlica, J. Kleinberg, M. Mahdian, and T. Wexler. *The role of compatibility in the diffusion of technologies through social networks.* In Proc. ACM Conference on Electronic Commerce, San Diego, USA, 2007, pp. 75–83.

[9] J. Hartline, V. S. Mirrokni, and M. Sundararajan. *Optimal marketing strategies over social networks.* In Proc. International World Wide Web Conference, Beijing, China, 2008, pp. 189–198.

[10] *Wikipedia, the free encyclopedia.* Available at http://en.wikipedia.org

[11] S. Wasserman, and K. Faust. *Social network analysis: Methods and applications.* Cambridge University Press, USA, 1994.

[12] S. Guha, R. Rastogi, and K. Shim. *ROCK: A Robust Clustering Algorithm for Categorical Attributes.* In Proc. International Conference on Data Engineering, Sydney, Australia, 1999, pp. 512–521.

[13] J. Vitter. *Random sampling with a reservoir.* ACM Transactions on Mathematical Software, Vol 11, 1985, pp. 37–57.

[14] S. Guha, R. Rastogi, and K. Shim. *CURE: A clustering algorithm for large databases.* In Proc. ACM SIGMOD International Conference on Management of Data, Seattle, USA, 1998, pp.73–84.

[15] N. Nisan, T. Roughgarden, E. Tardos, and V. V. Vazirani. *Algorithmic Game Theory.* Cambridge University Press, USA, 2007.

[16] W. Vickrey. *Counterspeculation, auctions and competitive sealed tenders.* Journal of Finance, 1961, pp. 8–37.

[17] E. H. Clarke. *Multipart pricing of public goods.* Public Choice Journal, 1971, pp. 17–33.

[18] T. Groves. *Incentives in teams.* Econometrica, 1973, pp. 617–631.

[19] J. F. Kurose, and K. W. Ross. *Computer Networking: A Top-Down Approach Featuring the Internet.* Third Edition, Addison Wesley Publishing Company, MA, USA, 2000.

[20] D. C. Parkes, and J. Shneidman. *Distributed implementations of Vickery-Clarke-Groves Mechanisms.* In Proc. International Conference on Autonomous Agents & Multi Agent Systems, New York, USA, 2004.

[21] J. Feigenbaum, and S. Shenker. *Distributed algorithmic mechanism design: recent results and future directions.* In Proc. International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications, Atlanta, USA, 2002, pp. 1–13.

واژه‌نامه

آستانه سرایت	*Contagion Treshold*
اتصال	*Interconnectivity*
انتخاب اجتماعی	*Social Choice*
انتشار نوآوری	*Diffusion of Innovation*
بازاریابی مستقیم	*Direct Marketing*
بازاریابی ویروسی	*Viral Marketing*
بازگشت کاهشی	*Diminishing Returns*
برچسب‌گذاری	*Labeling*
برخط	*Online*
پراکنده	*Sparse*
تاثیر شبکه‌ای	*Network Effect*
تاثیرگذاری.بهره‌گیری	*Influence – exploit*
تاییدیه	*Ack*
خوبی	*Goodness*
خود اتکایی	*Self Reliant*
دروازه خروجی	*Gateway*
دور افتاده	*Outlier*
راستگو	*Truthful*
رفاه اجتماعی	*Social Welfare*
زیرپیمانه‌ای	*Submodular*
سیستم مستقل	*Autonomous System*
شبکه‌های اجتماعی	*Social Networks*
طراحی مکانیزم	*Mechanism Design*
عامل	*Agent*
گام بعدی	*NEXT – HOP*

گراف اجتماعی	*Social Graph*
لیست پیوندی	*Linkedlist*
مجزا	*Disjoint*
مدل آبشاری	*Cascade Model*
مدل آستانه خطی	*Linear Treshold Model*
مدل آستانه عمومی	*General Treshold Model*
مرکز	*Center*
مسری	*Contagious*
مسیر سیستم مستقل	$AS - PATH$
مسیریاب	*router*
مقدار شبکه‌ای	*Network Value*
نرم‌افزار پیغامی	*Messenger*
نمونه‌برداری	*Sampling*
نوع	*Type*

Diffusion of Innovations in Social Networks based on Game Theoretic Approaches

Abstract

Recently, computer scientists and economists have defined many joint problems and cooperate widely in various areas. Importance of this interconnection is clear for everybody, now. New works have been conducted, nowadays, to use the daily - increasing web-based social networks in viral marketing for improving companies profits. The main problem which is proved to be NP-Complete in this context is about discovering k most influential nodes in a network. In this dissertation, we generalize the problem to a group-based version and we we use group-based advertising to achieve our main goal. A new algorithm called Group-Based Diffusion technique is proposed in this thesis for solving this problem efficiently. Moreover, lack of information in many real networks and necessity to have a distributed algorithm, satisfied us to propose an algorithm taking advantage of mechanism design. We organized the structure of the algorithm and hereby provide the truthfulness for our new algorithm.

: Social Networks, Viral Marketing, Group-Based Diffusion, Mechanism Design, Information Absence.

Diffusion of Innovations in Social Networks based on Game Theoretic Approaches

by

Milad Eftekhar

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Master of Science
in
Computer Engineering (Software)

Under supervision of
Dr. M. Ghodsi

June 2009

Computer Engineering Department

Sharif University of Technology

Tehran