مقاله پژوهشی

تشخیص انجمن در شبکه‌های پیچیده پویا مبتنی بر تعبیه گراف و خوشه‌بندی جمعی

مجید محمدپور، سیداکبر مصطفوی و وحید رنجبر

چکیده: امروزه شبکه‌های پیچیده پویا به یکی از ارکان مهم زندگی بشر تبدیل شده‌اند و تشخیص انجمن در این شبکه‌ها یکی از مهم‌ترین مسائل در تحلیل آنها محسوب می‌شود. در این مقاله یک روش تشخیص انجمن مبتنی بر تعبیه گراف و روش یادگیری جمعی ارائه شده که می‌تواند درجه پیمانه‌ای‌بودن هر انجمن را حداکثر نماید. روش‌های تعبیه گراف یا یادگیری نمایش کم‌بعد از گره‌ها در گراف به علت قابلیت کاربردی گسترده آن در عملکرد شبکه‌های پیچیده پویا مانند تشخیص انجمن در شبکه، بسیار مورد توجه قرار گرفته‌اند. در این مقاله، یک روش تعبیه گراف پویا مبتنی بر یادگیر عمیق پیشنهاد شده که گراف خروجی از مرحله تعبیه گراف را به‌عنوان ورودی به مدل یادگیر جمعی می‌دهد تا با دقت قابل قبولی، انجمن‌ها را در شبکه تشخیص دهد. همچنین یک الگوریتم حریصانه جدید به نام پیوند جمع برای بهینه‌سازی تابع هدف برای مجموعه داده‌های مقیاس بزرگ در زمان بسیار کوتاه ارائه گردیده است. نشان داده شده که پارتیشن توافقی پیشنهادی نسبت به پارتیشن‌های به‌دست‌آمده از کاربرد مستقیم روش‌های خوشه‌بندی جمعی رایج، به ساختارهای خوشه‌ای واقعی نزدیک‌تر است. روش پیشنهادی به‌دلیل استفاده از روش پیش‌پردازش مبتنی بر تعبیه گراف پیشنهادی و همچنین استفاده از روش خوشه‌بندی جمعی، توانسته کارایی مناسبی را در مقایسه با سایر روش‌های رقیب از خود نشان دهد. نتایج تجربی آزمایش‌های انجام‌شده حاکی از برتری روش پیشنهادی در مقایسه با روش‌های رقیب است.

کلیدواژه: **تعبیه گراف، تشخیص انجمن، درجه پیمانه‌ای، خوشه‌بندی جمعی، شبکه پیچیده، یادگیر عمیق.**

۱- مقدمه

اخیراً تحلیل گراف به دلیل فراگیری شبکه‌ها در دنیای واقعی، توجه فزاینده‌ای را به خود جلب کرده است. گراف‌ها (شبکه‌ها) برای نشان‌دادن اطلاعات در حوزه‌های مختلف مثل زیست‌شناسی (شبکه‌های برهم‌کنش پروتئین- پروتئین)، علوم اجتماعی (شبکه‌های دوست‌یابی) و زبان‌شناسی استفاده می‌شوند [۱] تا [۳]. مدل‌سازی تعاملات بین گراف‌ها، محققان را قادر ساخته تا سیستم‌های مختلف تحت شبکه را به‌صورت مجموعه‌ای از موجودیت‌های سیستمی درک کنند [۳]. مثلاً شبکه‌های اجتماعی برای برنامه‌های کاربردی مانند دوست‌یابی یا توصیه محتوا و نیز برای تبلیغ مورد استفاده قرار گرفته‌اند. وظایف تحلیل گراف را می‌توان به‌طور گسترده به چهار دسته تقسیم کرد: طبقه‌بندی گره [۴] و [۵]، پیش‌بینی لینک [۶]، خوشه‌بندی [۷] و مصورسازی [۸]. هدف طبقه‌بندی گره، مشخص‌کردن برچسب گره‌ها (رأس‌ها) بر پایه سایر گره‌های برچسب‌گذاری‌شده و توپولوژی شبکه است. پیش‌بینی لینک به وظیفه پیش‌بینی لینک یا لینک‌هایی که احتمالاً در آینده رخ می‌دهند، اشاره دارد. خوشه‌بندی برای پیداکردن زیرمجموعه‌هایی از گره‌های مشابه مورد استفاده قرار می‌گیرد و نهایتاً مصورسازی به فراهم‌کردن بینش‌هایی درباره ساختار شبکه کمک می‌کند. شبکه پویای پیچیده[1] به شبکه‌ای با مقیاس بالا گفته می‌شود که دارای ساختارهای توپولوژی پیچیده و رفتارهای پویاست که در گذر زمان تغییر می‌کند [۹]. محققان در رشته‌های مختلف سعی کرده‌اند تا سیستمی را که بر روی آن کار می‌کنند بر ساختار شبکه‌های پیچیده منطبق سازند. با استفاده از مشخصات و ابزارهایی که در زمینه شبکه‌های پیچیده وجود دارند به‌راحتی می‌توان سیستم را مدل نمود [۱۰]؛ مثلاً می‌توان به شبکه‌های زیستی، شبکه‌های ارتباطی، شبکه‌های اجتماعی و شبکه‌های بیماری‌های همه‌گیر اشاره کرد. در ابتدا مهم‌ترین هدف استفاده از شبکه‌های پیچیده، تحلیل رفتارهای نهفته در سیستم‌های مورد استفاده محققان در رشته‌های مختلف بود. با استفاده از امکاناتی که علم شبکه‌های پیچیده در اختیار محققان قرار می‌داد این امکان فراهم شد تا بتوانند تحلیل‌های دقیق و مفیدی را بر روی شبکه‌های مورد استفاده در زمینه‌های مختلف علمی داشته باشند. از جمله مهم‌ترین کارهایی که اخیراً محققین در حوزه شبکه‌های پیچیده پویا انجام داده‌اند، کاهش ابعاد شبکه با روش‌های تعبیه گراف[2] بوده است. تعبیه گراف می‌تواند شبکه را از فضای بزرگی به فضای کوچک‌تری تعبیه کند؛ بدون اینکه به ساختار گراف اصلی صدمه‌ای وارد شود و بنابراین این تعبیه گراف را می‌توان یکی از مهم‌ترین مراحل در پیش‌پردازش شبکه‌های پیچیده پویا در نظر گرفت. اگر هدف تشخیص انجمن‌ها[3] در شبکه پیچیده باشد، مرحله تعبیه گراف را می‌توان به‌عنوان ورودی مدل برای تشخیص انجمن‌ها در نظر گرفت. از این رو تعبیه گراف نهایتاً منجر به افزایش دقت مدل در تشخیص انجمن‌ها خواهد گردید. روش ارائه‌شده این مقاله شامل دو بخش اساسی است. بخش اول معرفی یک روش تعبیه گراف برای شبکه‌های پیچیده پویاست که با حفظ ساختار شبکه آن را به فضایی با ابعاد کوچک‌تر تعبیه

این مقاله در تاریخ ۲۸ اردیبهشت ماه ۱۴۰۱ دریافت و در تاریخ ۱۱ دی ماه ۱۴۰۱ بازنگری شد.

مجید محمدپور، دانشکده مهندسی کامپیوتر، دانشگاه یزد، یزد، ایران، (email: m.mohammadpour@stu.yazd.ac.ir).
سیداکبر مصطفوی، دانشکده مهندسی کامپیوتر، دانشگاه یزد، یزد، ایران، (email: a.mostafavi@yazd.ac.ir).
وحید رنجبر (نویسنده مسئول)، دانشکده مهندسی کامپیوتر، دانشگاه یزد، یزد، ایران، (email: vranjbar@yazd.ac.ir).

1. Complex Network
2. Graph Embedding
3. Community Detection

می‌کند که این روش تعبیه مبتنی بر یادگیر عمیق LSTM[1] است [۱۱]. بخش دوم ارائه مدلی است که از خوشه‌بندی جمعی[2] برای تشخیص انجمن‌ها استفاده می‌کند. تحقیقات پیشین نشان دادند به‌کارگیری اجماعی از خوشه‌بندها نسبت به خوشه‌بندی تکی، عملکرد بسیار چشم‌گیرتری دارد [۱۲] تا [۱۵]. ورودی مدل تشخیص انجمن‌ها همان تعبیه گراف است و چالش‌های عمده در حوزه تعبیه گراف به شرح زیر هستند [۱۶]:

- درک ناقص از نمایش تعبیه آموخته‌شده: در طول دهه گذشته، چندین روش تعبیه گراف پیشنهاد گردیده است که هدف همه آنها حفظ ویژگی‌های متغیر می‌باشد. با این حال، درک تفاوت اصلی بین آنها و توصیه‌کردن روش‌های تعبیه برای یک گراف جدید هنوز چالش‌برانگیز است.
- ساده‌سازی بیش از حد داده‌های دنیای واقعی: داده‌های دنیای واقعی غالباً چندوجهی بوده و حاوی اطلاعات بیشتری نسبت به داده‌های ثبت‌شده توسط گره‌ها و یال‌های گراف‌ها هستند. روش‌های اخیر نیز بر ثبت ویژگی‌های گره در تعبیه شبکه تمرکز دارند؛ اما ویژگی‌های یال که ممکن است تأثیر زیادی بر درک این شبکه داشته باشند، خیلی مطالعه نشده‌اند.
- فقدان مدل‌های تعبیه برای گراف‌های پویا: اکثر روش‌های جدید بر روی یک تصویر لحظه‌ای از یک گراف متمرکز هستند. شبکه‌های دنیای واقعی اغلب با پیشبرد زمان و پیوستن گره‌های جدید و یال‌های جدید در حال تکامل هستند و بنابراین ثبت پویایی غیربدیهی و چالش‌برانگیز است.

در روش تعبیه گراف این مقاله سعی خواهد شد که چالش‌های اساسی مذکور رفع و روشی ارائه گردد که پویایی شبکه را در نظر گرفته و تعبیه‌ها را برای گراف‌های جریانی به‌روزرسانی کند و بتواند الگوهای موقتی را در گراف‌های متوالی ثبت نماید.

در بخش مربوط به تشخیص انجمن‌ها روش خوشه‌بندی جمعی پیشنهادی به‌طور خلاصه به شرح زیر تشریح خواهد شد:

۱) یک چارچوب کلی برای انتخاب خوشه‌بندی جمعی پیشنهاد می‌شود. علاوه بر این، مسئله خوشه‌بندی جمعی را به مسئله تشخیص انجمن در شبکه تبدیل می‌کند و سپس از شاخص پیمانه برای حل بهینه آن استفاده می‌نماید.

۲) یک مدل جدید درجه دوم اعداد صحیح[3] برای حداکثرسازی درجه پیمانه[4] به‌عنوان تابع اجماع پیشنهاد می‌شود.

۳) یک طرح انباشته حریصانه[5] جدید به‌عنوان یک استراتژی جستجوی بسیار سریع برای حل مدل به‌ویژه در مجموعه داده‌های مقیاس بزرگ پیشنهاد می‌شود.

۴) الگوریتم پیشنهادی را با یک مثال توصیفی و بحث در مورد پیچیدگی زمان و مکان و همچنین تحلیل همگرایی غنی می‌کنیم.

نوآوری اساسی روش پیشنهادی این مقاله، ترکیب دو مدل پیشنهادی تعبیه گراف پویا با یک روش خوشه‌بندی مبتنی بر اجماع است.

در ادامه و در بخش بعد، ابتدا تعاریف مربوط به روش پیشنهادی آورده شده و سپس مقدمه‌ای بر روش‌های تعبیه گراف ارائه خواهد شد. همچنین توضیحاتی راجع به یادگیر عمیق[6] و خوشه‌بندی جمعی آمده و کارهای گذشته در این بخش مرور خواهند گردید. بخش ۳ روش پیشنهادی را ارائه می‌دهد. در بخش ۴ نتایج ارزیابی‌های انجام‌شده با مجموعه داده‌های واقعی آمده و بخش نهایی مقاله شامل نتیجه‌گیری است.

۲- ادبیات تحقیق و مرور منابع

در این بخش سعی گردیده که ابتدا تعاریف مرتبط با روش ارائه‌شده در این مقاله مختصراً بیان شوند و سپس روش‌های تعبیه گراف تشریح گردد. در ادامه، خوشه‌بندی جمعی نیز توضیح داده شده و نهایتاً سعی گردیده تا جدیدترین کارهای گذشته در این حوزه مرور شوند.

تعریف ۱: گراف

گراف $G=(V,E)$ مجموعه‌ای از $V=\{v_1,v_2,\ldots,v_n\}$ رأس و $E=\{e_{ij}\}_{i,j}^{n}$ یال است. ماتریس مجاورت S از گراف G شامل وزن‌های نامنفی همراه با هر یال است؛ به‌طوری که اگر v_i و v_j به هم متصل باشند آنگاه $S_{ij}=1$ و اگر v_i و v_j به هم متصل نباشند آنگاه $S_{ij}=0$ است. به‌طور کلی وزن یال S_{ij} به‌عنوان یک معیار شباهت بین گره‌های v_i و v_j محسوب می‌شود که هرچه وزن یال بیشتر باشد، انتظار می‌رود که دو گره شباهت بیشتری داشته باشند.

تعریف ۲: مجاورت مرتبه اول

مجاورت مرتبه اول نشان می‌دهد که آیا بین جفت‌رئوس یالی وجود دارد یا خیر. دو رأس را مجاور می‌گوییم، اگر یک یال بین آنها وجود داشته باشد. در اینجا مجاورت رئوس با یال متصل‌کننده آنها تعریف می‌شود. وزن‌های یال S_{ij} مجاورت مرتبه اول بین گره‌های v_i و v_j نامیده می‌شوند؛ چون اولین معیار شباهت بین دو گره هستند.

تعریف ۳: مجاورت مرتبه دوم

مجاورت مرتبه دوم بین یک جفت از گره‌ها، مجاورت ساختار همسایگی آنها را توصیف می‌کند. فرض کنید که $s_i=[s_{i1},\ldots,s_{in}]$ نشان‌دهنده مجاورت مرتبه اول بین v_i و سایر گره‌ها باشد. سپس مجاورت مرتبه دوم بین v_i و v_j با شباهت s_i و s_j تعیین می‌شود. مجاورت مرتبه دوم همسایگی این دو گره را با هم مقایسه می‌کند و اگر همسایه مشابهی داشته باشند، آنها را به‌عنوان همسایه لحاظ می‌نماید. تعریف مجاورت‌های درجه بالاتر با استفاده از معیارهای دیگر مثل شاخص کاتز[7]، رتبه صفحه[8]، همسایه‌های مشترک و ... امکان‌پذیر است.

تعریف ٤: تعبیه گراف

با توجه به گراف $G=(V,E)$، تعبیه گراف یک نگاشت $f:v_i\rightarrow y_i\in\mathbb{R}^d,\forall i\in[n]$ است به‌گونه‌ای که $d\ll|V|$ باشد و تابع f برخی معیارهای مجاورت تعریف‌شده در گراف G را حفظ کند. بنابراین تعبیه نگاشت هر گره به یک بردار ویژگی با ابعاد کم و تلاش برای حفظ نقاط اتصال بین رأس‌ها است. مثلاً یک تعبیه حفظ‌کننده مجاورت مرتبه اول ممکن است با به‌حداقل‌رساندن $\sum_{i,j}s_{ij}\|y_i-y_j\|_2^2$ به‌دست آید. فرض کنید که دو جفت گره (v_i,v_j) و (v_i,v_k) با نقاط اتصال مرتبط باشند؛ به‌گونه‌ای که $s_{ij}>s_{ik}$ است. در این حالت، دو جفت گره (v_i,v_j) به نقاط در فضای تعبیه نگاشت خواهند شد که از نگاشت

1. Long Short-Term Memory
2. Ensemble Clustering
3. Integer Quadratic Model
4. Modularity
5. Greedy Agglomerative Scheme
6. Deep Learning
7. Katz
8. Page Rank

(v_i, v_k) به یکدیگر نزدیک‌تر هستند. تعبیه گراف پویا، مفهوم تعبیه را به گراف‌های پویا تعمیم می‌دهد.

تعریف ۵: تشخیص انجمن

مفهوم تشخیص انجمن در علم شبکه به‌عنوان روشی برای یافتن گروه‌ها در سیستم‌های پیچیده از طریق نمایش در یک گراف پدیدار شده است. روش‌های تشخیص انجمن، زیرشبکه‌هایی را پیدا می‌کنند که از نظر آماری، پیوندهای قابل توجهی بین گره‌های یک گروه نسبت به گره‌ها در گروه‌های مختلف دارند [۱۶].

تعریف ٦: یادگیر عمیق

مدل‌های یادگیر عمیق، مدل‌های برگرفته از یادگیر ماشین در هوش مصنوعی هستند که به‌طور خودکار ویژگی‌های مربوط را از یک گراف استخراج می‌کنند.

۲-۱ تعبیه گراف

بسیاری از وظایف مهم در تجزیه و تحلیل شبکه، شامل پیش‌بینی بر روی گره‌ها و یا لبه‌های یک گراف است که به الگوریتم‌های مؤثر برای استخراج الگوهای معنادار و ساخت ویژگی‌های پیش‌بینی نیاز دارد. اخیراً در میان بسیاری از تلاش‌ها برای رسیدن به این هدف، تعبیه گراف یعنی یادگیری نمایش با ابعاد کم برای هر گره در گراف که ارتباط آن را با گره‌های دیگر به‌طور دقیق نشان می‌دهد، توجه زیادی را به خود جلب کرده است. نشان داده شده که تعبیه گراف در بسیاری از وظایف یادگیری نظارت‌شده مانند طبقه‌بندی گره[1]، پیش‌بینی لینک[2] و ساخت گراف[3] و خوشه‌بندی[4]، نسبت به جایگزین‌ها برتری دارد [۱۷] تا [۲۰]. اکثر روش‌های موجود بر روی گراف‌های ایستا تمرکز دارند [۲۱] تا [۲۳]. از جمله روش‌های ایستا می‌توان به مدل‌های مبتنی بر تجزیه مقادیر منفرد (SVD) اشاره نمود که ماتریس مجاورت لاپلاسین یا مرتبه بالا را برای ایجاد تعبیه گره تجزیه می‌کنند [۲۴]. سایر موارد شامل مدل‌های مبتنی بر قدم‌زدن تصادفی[5] [۲۵]، تعبیه‌هایی را از قدم‌زدن تصادفی محلی و بسیاری بسیاری دیگر ایجاد می‌کنند. اخیراً در [۲۶] مدلی ابتکاری به نام SDNE طراحی شده که از یک رمزگذار خودکار عمیق برای مدیریت غیرخطی‌بودن برای تولید تعبیه‌های دقیق‌تر استفاده می‌کند [۲۶]. با این حال در کاربردهای عملی، بسیاری از گراف‌ها مانند شبکه‌های اجتماعی، پویا هستند و در طول زمان تکامل می‌یابند [۲۷] و [۲۸]. معمولاً ما گراف‌های پویا را به‌عنوان مجموعه‌ای از تصاویر لحظه‌ای گراف[6] [۲۸] و [۲۹] در مراحل زمانی مختلف نشان می‌دهیم [۳۰] و [۳۱]. در ادامه برخی از مشهورترین روش‌های تعبیه گراف در جدول ۱ بررسی شده است.

۲-۲ خوشه‌بندی جمعی

دو گرایش جدید در رویکردهای خوشه‌بندی جمعی وجود دارد: انتخاب خوشه‌بندی جمعی و بهینه‌سازی اجماع. در این بخش به بررسی برخی از مطالعات به‌روز در این زمینه‌ها پرداخته می‌شود.

1. Node Classification
2. Link Prediction
3. Graph Generation
4. Clustering
5. Random Walk
6. Snapshots

۲-۲-۱ انتخاب خوشه‌بندی جمعی

در اکثر مطالعات قبلی، همه پارتیشن‌ها و خوشه‌های آنها در اجماع، دارای وزن یکسانی در تصمیم‌گیری نهایی هستند [۳۷] و [۳۸]. با این حال از نظر منطقی به نظر می‌رسد سنجش ایده‌های بهتر، تصمیم‌گیری نهایی مجمع را تقویت می‌کند. در رویکرد انتخاب خوشه‌بندی جمعی، ایده این است که زیرمجموعه‌ای از خوشه‌بندی‌های پایه را انتخاب کنیم تا پارتیشن توافقی حاصل از این زیرمجموعه برابر یا بهتر از مجمع کامل باشد. انتخاب خوشه‌بندی جمعی از آموزش یادگیری گروه نظارت‌شده پیروی می‌کند [۳۹]. مسئله انتخاب خوشه‌بندی جمعی برای اولین بار در [۴۰] معرفی و بیشتر در [۴۱] تا [۴۶] بررسی شد. در روش پیشنهادی [۴۷]، ابتدا مجموعه‌های خوشه‌ای چندگانه تولید می‌شوند و سپس مجمعی با تنوع میانه برای تولید خوشه‌بندی نهایی انتخاب می‌گردد. در مقابل، فرن و لین در [۴۱] به دنبال انتخاب یک زیرمجموعه کوچک از یک مجموعه داده‌شده بزرگ برای تشکیل مجمع نهایی هستند. آنها روش‌های انتخاب مجمع خود را بر اساس کیفیت و تنوع (دو عامل تأثیرگذار در عملکرد خوشه‌بندی جمعی) طراحی کردند و از معیار اطلاعات متقابل نرمال (NMI) استفاده نمودند که ابتدا توسط استرل و گاش در [۳۸] تعریف شد و در [۳۷] تکامل یافت تا پارتیشن‌های اولیه را ارزیابی کنند. آنها با نتایج تجربی خود نشان دادند که انتخاب زیرمجموعه‌ای از پارتیشن‌ها، نتایج بهتری نسبت به پارتیشن‌های کامل به همراه دارد. مشابه فرن و لین [۳۷]، علیزاده و همکاران در [۴۴] اخیراً یک روش انتخاب پیشنهاد داده‌اند که مجموعه‌ای از خوشه‌های عملکرد بهتر را ایجاد می‌کند. آنها ایده انتخاب خوشه‌بندی جمعی را از سطح پارتیشن‌ها به خوشه‌های فردی گسترش دادند و معیار جدیدی به نام MAX را برای ارزیابی پایداری خوشه‌ها معرفی کردند و به‌طور تجربی نشان دادند که انتخاب خوشه‌های واجد شرایط بالا با توجه به معیار MAX، پارتیشن اجماع بهتری را به همراه دارد. این بدان معنی است که بین ثبات خوشه‌های انتخاب‌شده و عملکرد خوشه‌بندی نهایی قیاس وجود دارد. علیزاده و همکاران در [۳۹] معیار اصلاح‌شده‌ای MAX را برای ارزیابی پایداری خوشه‌های فردی به نام معیار AAPMM[7] معرفی کردند.

۲-۲-۲ بهینه‌سازی اجماع

یکی از روندهای بسیار جدید در مسئله خوشه‌بندی جمعی، نگاشت مسئله به یک بهینه‌سازی ریاضی است. در این رویکرد، پارتیشن اجماع از حل یک روش بهینه‌سازی به‌دست می‌آید. با توجه به مجموعه‌ای از نتایج خوشه‌بندی، هدف مسئله بهینه‌سازی، یافتن پارتیشن بهینه از طریق حداکثرکردن (حداقل‌سازی) یک تابع هدف است. یک ویژگی مشترک در بیشتر مطالعات قبلی، تکیه بر مدل‌سازی نمونه‌ای از مسئله خوشه‌بندی جمعی به‌عنوان یک گراف شامل تعدادی گره و یال است. گره‌ها معمولاً نقاط داده را و یال‌ها مقداری از شباهت بین گره‌ها را نشان می‌دهند که از مجموعه داده‌شده محاسبه می‌گردند. نمایش گراف یک مجمع، صرف نظر از پیچیدگی الگوریتم برای کار، احتمالاً نتایجی کمتر از حد بهینه ایجاد می‌کند. تحقیقات جدیدتر در زمینه خوشه‌بندی جمعی تمایل به فرمول‌بندی مسئله به عنوان یک مسئله بهینه‌سازی و سپس حل آن با استفاده از حل‌کننده‌های ریاضی (یا حتی حل‌کننده‌های بهینه‌سازی هوشمند) را نشان می‌دهد [۴۸] تا [۵۲]. در [۴۹] مدل بهینه‌سازی ارائه گردیده که توافق را به حداکثر می‌رساند و عدم توافق نتیجه، توافق را با

7. Alizadeh-Parvin-Moshki-Minaei

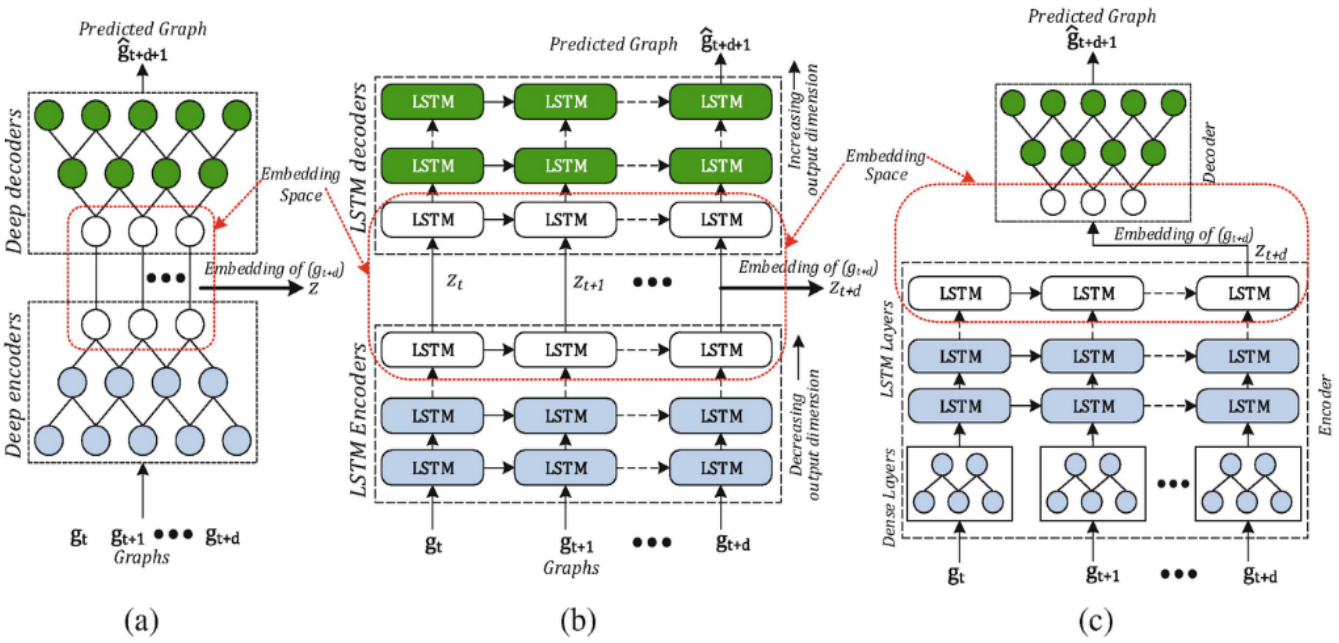
جدول ۱: برخی از مشهورترین روش‌های تعبیه گراف.

روش	توضیحات	مزایا و معایب
روش‌های مبتنی بر تجزیه مقادیر منفرد (SVD) [۲۴]	الگوریتم‌های مبتنی بر تجزیه مقادیر منفرد، ارتباط بین گره‌ها را به شکل یک ماتریس نشان می‌دهند و این ماتریس را برای به‌دست‌آوردن تعبیه، تجزیه می‌کنند.	مزایا: استفاده گسترده در کاربردهای عملی، مقیاس‌پذیربودن، اجرای آسان معایب: ایجاد نویز در راه حل، مثبت‌نبودن ماتریس مجاورت
تعبیه خطی محلی [۱] (LLE) [۳۲]	روش LLE فرض می‌کند که هر گره، یک ترکیب خطی از همسایه‌های آن در فضای تعبیه است. این روش رابطه غیرخطی بین ویژگی‌ها را با تصویرکردن یک خم خطی محلی در فضای ویژگی می‌یابد.	مزایا: پیاده‌سازی ساده، استخراج ورودی‌های با اطلاعات بیشتر، یافتن وابستگی‌های غیرخطی ویژگی‌ها، حفظ ساختار همسایگی داده‌ها معایب: عملکرد ضعیف در داده‌های مربوط به صوت
نگاشت ویژه لاپلاس [۳۳]	هدف نگاشت ویژه لاپلاس، حفظ تعبیه دو گره با تأکید بر جفت فاصله‌های بزرگ است. نگاشت ویژه‌های لاپلاس از تابع جریمه درجه دو در فاصله بین تعبیه‌ها استفاده می‌کند.	مزایا: سادگی روش معایب: عدم حفظ توپولوژی محلی در کاربردهای عملی
تعبیه کوشی [۲] [۳۳]	بر فاصله کوتاه تأکید می‌کند و تضمین می‌نماید که هرچه دو گره شبیه‌تر باشند، در فضای تعبیه نزدیک‌تر خواهند بود.	مزایا: حفظ توپولوژی محلی در فضای تعبیه معایب: پیچیدگی تابع هدف، عدم بهینگی در شبکه‌های پیچیده
تعبیه شبکه عمیق ساخت‌یافته (SDNE) [۳] [۳۴]	از خودرمزگذارهای عمیق برای حفظ همسایگی‌های مرتبه اول و دوم شبکه استفاده می‌شود. این روش از توابع غیرخطی برای به‌دست‌آوردن قابلیت تعبیه استفاده می‌کند.	مزایا: حفظ ساختار شبکه محلی و سراسری، بهینگی برای شبکه‌های با ساختار پیچیده معایب: از نظر محاسباتی هزینه‌بر و برای گراف‌های پراکنده بزرگ، غیربهینه
شبکه عصبی عمیق DNGR [۴] [۳۵]	موج‌سواری تصادفی را با خودرمزگذار عمیق ترکیب می‌کند. این مدل شامل ۳ بخش است: ۱) موج‌سواری تصادفی، ۲) محاسبه اطلاعات متقابل نقطه‌به‌نقطه مثبت و ۳) خودرمزگذارهای حذف نویز روی هم انباشته.	مزایا: استحکام مدل در حضور نویز در گراف، ثبت ساختار زیربنایی مورد نیاز برای وظایفی مانند پیش‌بینی لینک و طبقه‌بندی گره معایب: هزینه محاسباتی و غیربهینه برای گراف‌های پراکنده بزرگ
شبکه‌های کانولوشن گراف (GCN) [۵] [۳۶]	این مدل به‌طور تکراری، تعبیه‌های همسایه‌ها برای یک گره را تعریف می‌کند و از تابع به‌دست‌آمده و تعبیه آن در تکرار قبلی برای به‌دست‌آوردن تعبیه جدید استفاده می‌نماید و با تعریف یک عملگر کانولوشن بر روی گراف، با مشکل هزینه محاسباتی و غیربهینه برای گراف‌های پراکنده بزرگ، مقابله می‌کند.	مزایا: حل مشکل عدم بهینگی برای گراف‌های پراکنده و بزرگ، مقیاس‌پذیری معایب: وابستگی به فیلترهای مکانی و طیفی، یافتن مقدار مناسب برای عملگر استفاده‌شده در این روش

1. Locally Linear Embedding
2. Cauchy Graph Embedding
3. Structural Deep Network Embedding
4. Deep Neural Networks for Graph Representations
5. Embedding Based Graph Convolutional Network

توجه به اعضای مجمع به‌طور همزمان به حداقل می‌رساند.

نویسندگان در [۵۳] رابطه جدیدی از مسئله خوشه‌بندی جمعی را پیشنهاد کرده‌اند که از نمایش رشته برای رمزگذاری اطلاعات مجمع استفاده می‌کند. فرمول پیشنهادی از منطق فازی به‌منظور تعریف تابع هدف فازی استفاده می‌کند. آنها فرمول پیشنهادی خود را در یک مدل بهینه‌سازی ریاضی قرار دادند. اگرچه هر حل‌کننده ریاضی یا اکتشافی می‌تواند برای حل یک برنامه غیرخطی استفاده شود، آنها از یک الگوریتم ژنتیک تخصصی برای حل برنامه غیرخطی پیشنهادی خود استفاده کردند. نتایج تجربی آنها توانایی عملگرهای الگوریتم ژنتیک اصلاح‌شده را در مدیریت معضل اکتشاف- بهره‌برداری نشان داد. به موازات آن، تعدادی از مطالعات برای تبدیل یک وظیفه تشخیص انجمن به برنامه‌ریزی ریاضی در زمینه شبکه‌های پیچیده وجود دارد [۵۴] تا [۵۷]. محققان در این زمینه، مدل‌های ریاضی مختلفی را برای بهینه‌سازی برخی معیارهای کیفیت در انجمن‌ها تعریف کردند. از آنجایی که مجموعه خوشه اصلی و مسئله تشخیص انجمن، اصولاً به یکدیگر نزدیک هستند به نظر می‌رسد که پیوستن به آنها و استفاده از دستاوردهای یکی در دیگری پیشرفت‌های قابل توجهی را به همراه داشته باشد [۵۸]. اخیراً در [۵۸] با موفقیت از خوشه‌بندی جمعی در تشخیص انجمن استفاده شده و نویسندگان نشان دادند که خوشه‌بندی جمعی را می‌توان با هر روش موجود به روشی خودسازگار ترکیب کرد و ثبات و دقت انجمن‌های حاصل را به‌طور قابل توجهی افزایش داد. علاوه بر این نویسندگان در [۵۹] اصل خوشه‌بندی جمعی را در گراف‌ها با ساخت مجموعه‌ای از گراف‌ها مشابه گراف اولیه، اعمال روش خوشه‌بندی برای هر یک از آنها، ساختن یک مجمع از پارتیشن‌ها و محاسبه اجماع، پیاده‌سازی کردند. پارتیشن‌بندی برای این نمایه به پارتیشن‌بندی گراف نهایی منجر می‌شود. آنها ادعا کردند که روش خوشه‌بندی bootstrap امکان بهبود کیفیت پارتیشن را به‌خصوص زمانی که عدم قطعیت در لبه‌های گراف وجود دارد، فراهم می‌کند. همچنین در [۵۸] یک طرح اجماع برای تشخیص انجمن در شبکه‌های اجتماعی پیشنهاد گردید. نویسندگان در [۵۸] نشان دادند که روش آنها جوامع پایدار و قوی را به همراه دارد. در این مقاله، چارچوبی برای ترسیم یک مسئله خوشه‌بندی جمعی به یک مسئله حداکثرسازی معیار پیمانه‌ای پیشنهاد می‌شود. سپس با استفاده از تجربیات قبلی در مدل‌سازی مسئله حداکثرسازی پیمانه‌ای به‌عنوان یک برنامه ریاضی، یک مدل ریاضی جدید، ارائه و بعداً با یک الگوریتم سریع حریصانه حل خواهد شد. ایده این مقاله بدین صورت تشریح می‌گردد: ابتدا یک روش تعبیه گراف برای شبکه‌های پیچیده پویا معرفی شده که این روش به‌عنوان گراف پیش‌پردازش به مدل اجماع پیشنهادی داده خواهد شد تا بتوان با عملکرد بالایی، انجمن‌ها در شبکه‌های پیچیده پویا تشخیص داده شوند.



شکل ۱: معماری تعبیه گراف پیشنهادی با روش یادگیر عمیق LSTM.

۲-۲-۳ تابع توافقی

آخرین و نیز مهم‌ترین مرحله در چارچوب انتخاب خوشه‌بندی جمعی، تجمیع خوشه‌های انتخابی است. چند رویکرد اساسی برای ترکیب نتایج وجود دارد. یک رویکرد آن است که خوشه‌های انتخاب‌شده را به‌عنوان ورودی‌های الگوریتم هایپرگراف از جمله الگوریتم تقسیم‌بندی هایپرگراف (HGPA)، الگوریتم خوشه‌بندی (MCLA) و الگوریتم تقسیم‌بندی شباهت مبتنی بر خوشه (CSPA) در نظر بگیریم [۶۰]. با توجه به خوشه‌های انتخاب‌شده به‌عنوان ورودی این الگوریتم‌ها، خروجی پارتیشن توافقی تولید می‌شود. روش دیگر برای استخراج پارتیشن نهایی این است که خوشه‌های انتخاب‌شده را به‌عنوان یک فضای داده جدید در نظر گرفته و از یک الگوریتم خوشه‌بندی مشترک مانند k-means یا k-means فازی برای پارتیشن‌بندی آنها استفاده کنیم [۵۹]. خوشه‌بندی انباشت شواهد (EAC)، روش دیگری است که در آن اطلاعات هم‌رخدادی خوشه‌های انتخاب‌شده در ماتریس همبستگی انباشته می‌شوند. سپس یک الگوریتم پیوند برای استخراج پارتیشن اجماع از این ماتریس استفاده می‌گردد [۳۹]، [۴۴] و [۴۵].

۳- روش پیشنهادی

در این بخش ابتدا روش تعبیه گراف که مبتنی بر روش یادگیر عمیق است مفصلاً تشریح می‌شود. از این مدل به‌عنوان یک مرحله پیش‌پردازش برای ورودی مدل پیشنهادی تشخیص انجمن‌ها که در ادامه آمده، استفاده شده است. در ادامه روش پیشنهادی مبتنی بر خوشه‌بندی جمعی برای تشخیص انجمن‌ها در شبکه‌های پیچیده پویا تشریح می‌گردد.

۱-۳ مسئله تعبیه گراف پویا

گراف وزن‌دار $G=(V,E)$ را در نظر بگیرید که V و E به‌ترتیب مجموعه‌ای از رئوس و یال‌ها هستند. ماتریس مجاورت G را با $\mathcal{A}$ نشان می‌دهیم؛ یعنی برای یک یال $(i,j)\in E$، $\mathcal{W}_{i,j}$ وزن آن یال را در ماتریس مجاورت $\mathcal{A}$ نشان می‌دهد. در ماتریس مجاورت اگر یالی از i به j باشد، وزن آن یعنی $\mathcal{W}_{i,j}$ در ماتریس مجاورت $\mathcal{A}$ ثبت می‌گردد؛ در غیر این صورت اگر یالی وجود نداشته باشد آنگاه $\mathcal{W}_{i,j}=0$ است.

تکامل گراف G به‌صورت $G=\{G_1,G_2,\ldots,G_T\}$ نشان داده می‌شود که در آن G_T وضعیت نمودار را در لحظه t نشان می‌دهد. مسئله تعبیه گراف به این شکل تعریف می‌شود: با درنظرگرفتن تکامل گراف G، هدف نمایش هر گره v در فضای برداری با ابعاد پایین $\mathcal{D}_{v_1},\ldots,\mathcal{D}_{v_t}$ است که در آن $\mathcal{D}_{v_t}$ تعبیه گره v در زمان t با یادگیری نگاشت‌های $\mathcal{D}_{v_t}=f_i(V_1,\ldots,V_t,E_1,\ldots,E_t)$ و $\mathcal{D}_{v_t}=f_i(V_1,\ldots,V_t,E_1,\ldots,E_t)\rightarrow\mathbb{R}^d$ است به‌گونه‌ای که $\mathcal{D}_{v_t}$ می‌تواند الگوهای زمانی مورد نیاز برای پیش‌بینی $\mathcal{D}_{v_{t+1}}$ را ثبت کند. به عبارت دیگر، تابع تعبیه در هر مرحله زمانی از اطلاعات، از تکامل گراف برای ثبت پویایی‌های شبکه استفاده می‌کند و در نتیجه می‌تواند انجمن‌ها را با دقت بالاتری تشخیص دهد.

۱-۱-۳ مدل یادگیری

روش تعبیه گراف پیشنهادی، یک مدل یادگیر عمیق است که به‌عنوان ورودی، یک مجموعه از گراف‌های قبلی را می‌گیرد و به‌عنوان خروجی، گراف در گام زمانی بعدی را تولید می‌کند؛ بنابراین تعاملات غیرخطی بین رئوس در هر گام زمانی را به‌دست می‌آورد.

پیشرفت‌های اخیر در یادگیر عمیق نظارت‌نشده، نشان داده که رمزگذارهای خودکار[1] می‌توانند با موفقیت بازنمایی‌های کم‌بعدی بسیار پیچیده به داده‌ها برای وظایف مختلف بیاموزند. مدل تعبیه گراف پویا از یک رمزگذار خودکار عمیق برای نگاشت داده‌های ورودی به یک فضای نهفته غیرخطی بالا استفاده می‌کند تا روندهای اتصال را در یک تصویر لحظه‌ای گراف در هر مرحله زمانی ثبت کند. مدل پیشنهادی برای تعبیه گراف پویا، یک مدل نیمه‌نظارت‌شده است و ترکیبی از دو تابع هدف مربوط به مجاورت مرتبه اول و مجاورت مرتبه دوم را به حداقل می‌رساند. مدل رمزگذار خودکار در شکل ۱ نشان داده شده و نمادهای مورد استفاده در این مدل همانند روش [۶۱] هستند.

چون تعبیه گراف، تکامل زمانی لینک‌ها را ثبت می‌کند، این امکان را می‌دهد تا لینک گراف را در گام زمانی بعدی پیش‌بینی کند [۶۱]. این مدل، تعبیه شبکه در مرحله زمانی t را با بهینه‌سازی تابع زیان زیر یاد می‌گیرد

1. Autoencoders

$$\ell_{t+1} = \left\|(\hat{\mathcal{W}}_{t+\ell+1} - \mathcal{W}_{t+\ell+1})\right\| \otimes \mathcal{B}\|_{\mathcal{F}}^{2} = \left\|f(\mathcal{W}_{t},\ldots,\mathcal{W}_{t+l}) - (\mathcal{W}_{t+\ell+1})\right\| \otimes \mathcal{B}\|_{\mathcal{F}}^{2} \quad (۱)$$

در اینجا بازسازی نادرست یال‌ها در زمان $t+\ell+1$ با استفاده از تعبیه در مرحله زمانی $t+\ell$ جریمه می‌شود. به‌حداقل‌رساندن این تابع زیان، پارامترها را به‌گونه‌ای تنظیم می‌کند که بتواند روابط الگوهای تکاملی بین گره‌ها را ثبت کند تا یال‌ها را در گام زمانی آینده برای یک تعبیه مناسب در گراف پیچیده پویا پیش‌بینی کند. تعبیه در مرحله زمانی $t+d$ تابعی از گراف در مراحل زمانی $t, t+1, \ldots, t+\ell$ است. از ماتریس وزن‌دهی $\mathcal{B}$ مطابق [۶۲] برای وزن‌دادن و بازسازی یال‌های مشاهده‌شده، بالاتر از یال‌های مشاهده‌نشده استفاده می‌گردد.

در اینجا برای $(i,j) \in E_{t+\ell+1}$ مقدار $\mathcal{B}_{ij} = \mathcal{B}$ خواهد بود که $\mathcal{B}$ یک ابرپارامتر کنترل‌کننده وزن جریمه یال‌های مشاهده‌شده است. توجه داشته باشید که $\otimes$ یک ضرب عنصر به عنصر را نشان می‌دهد. یک راه ساده برای گسترش خودرمزگذارها که به‌طور سنتی برای تعبیه گراف‌های ایستا [۶۱] به گراف‌های زمانی استفاده می‌شوند، این است که اطلاعات مربوط به گراف‌های قبلی را به‌عنوان ورودی به خودرمزگذار اضافه کنیم. این مدل از چندین لایه تمام متصل برای مدل‌سازی اتصال گره‌ها در طول زمان استفاده می‌کند. برای یک گره u با مجموعه بردار همسایگی $u_{1,t} = [a_{u_t}, \ldots, a_{u_{t+1}}]$، نمایش پنهان اولین لایه به‌صورت زیر آموخته می‌شود [۶۲]

$$\mathcal{D}_{u_t}^{(1)} = f_a(\mathcal{W}_{AE}^{(1)} u_{1,\ldots,t} + b^{(1)}) \quad (۲)$$

که در آن f_a تابع فعال‌سازی و $\mathcal{W}_{AE}^{(1)} u_{1,\ldots,t} \in \mathbb{R}^{d^{(1)} \times nl}$ و $d^{(1)}$، n و l به‌ترتیب ابعاد بازنمایی آموخته‌شده توسط لایه اول، تعداد گره‌های گراف و پارامتر بازگشت هستند. نمایش kامین لایه به‌صورت رابطه زیر تعریف می‌شود [۶۲]

$$\mathcal{D}_{u_t}^{(k)} = f_a(\mathcal{W}_{AE}^{(k)} \mathcal{D}_{u_t}^{(k-1)} + b^{(k)}) \quad (۳)$$

توجه داشته باشید که مدل ارائه‌شده در [۶۲] دارای $O(nld)^{(1)}$ پارامتر است. چون اغلب گراف‌های دنیای واقعی پراکنده‌اند، یادگیری پارامترها می‌تواند چالش‌برانگیز باشد. برای کاهش تعداد پارامترهای مدل و رسیدن به یک یادگیری زمانی مؤثرتر، مدل تعبیه گراف پویا پیشنهاد شده است. در این مدل از شبکه‌های حافظه کوتاه‌مدت (LSTM) برای یادگیری تعبیه استفاده می‌شود. LSTM نوعی از شبکه‌های عصبی تکرارشونده (RNN) و قادر به رسیدگی به مسئله‌های وابستگی بلندمدت است. در گراف‌های پویا، وابستگی‌های طولانی‌مدت وجود دارند که ممکن است با خودرمزگذارهای تمام‌متصل ثبت نشوند. بازنمایی حالت پنهان یک شبکه LSTM به‌صورت زیر تعریف می‌شود [۶۲]

$$\begin{aligned}
\mathcal{D}_{ut}^{(1)} &= o_{ut}^{(1)} \times \tanh C_{ut}^{(1)} \\
o_{ut}^{(1)} &= \sigma_{ut}(\mathcal{W}_{RNN}^{(1)}[\mathcal{D}_{ut-1}^{(1)} u_{1,\ldots,t}] + b_o^{(1)}) \\
C_{ut}^{(1)} &= f_{ut}^{(1)} \times C_{ut-1}^{(1)} + i_{ut}^{(1)} \times \tilde{C}_{ut}^{(1)} \\
\tilde{C}_{ut}^{(1)} &= \tanh(\mathcal{W}_C^{(1)}.[\mathcal{D}_{ut-1}^{(1)}, u_{1,\ldots,t} + b_c^{(1)}]) \\
i_{ut}^{(1)} &= \sigma(\mathcal{W}_i^{(1)}.[\mathcal{D}_{ut-1}^{(1)}, u_{1,\ldots,t}] + b_i^{(1)}) \\
f_{ut}^{(1)} &= \sigma(\mathcal{W}_f^{(1)}.[\mathcal{D}_{ut-1}^{(1)}, u_{1,\ldots,t} + b_f^{(1)}])
\end{aligned} \quad (۴)$$

که در آن C_{ut} حالات سلول LSTM را نشان می‌دهد، $\mathcal{W}$ ماتریس وزنی است که رفتار دروازه‌های به‌روزرسانی، فراموشی و خروجی را کنترل می‌کند، f_{ut} مقدار راه‌اندازی دروازه فراموشی است که وظیفه کنترل جریان اطلاعات از گام زمانی قبلی را دارد، i_{ut} نشان‌دهنده مقدار راه‌اندازی دروازه به‌روزرسانی است که وظیفه کنترل جریان اطلاعات جدید را بر عهده دارد، o_{ut} مقدار راه‌اندازی دروازه خروجی را نشان می‌دهد و مشخص می‌کند چه میزان از اطلاعات گام زمانی قبل به همراه اطلاعات گام زمانی فعلی به گام زمانی بعد منتقل شود، σ تابع فعال‌سازی سیگموئید، $\tilde{C}_{ut}$ حالت کاندیدای تخمینی جدید و b نشان‌دهنده بایاس است. تابع tanh کمک می‌کند تا مقادیری که در طول شبکه در جریان هستند تعدیل[1] شوند. l شبکه سلولی متصل در لایه اول وجود دارد که در آن حالات، سلول و نمایش پنهان در یک زنجیره از $t-\ell$ به t شبکه‌های LSTM منتقل می‌شوند. سپس بازنمایی لایه kام به‌صورت زیر مشخص می‌شود

$$\begin{aligned}
\mathcal{D}_{ut}^{(k)} &= o_{ut}^{(k)} \times \tanh C_{ut}^{(k)} \\
o_{ut}^{(k)} &= \sigma_{ut}(\mathcal{W}_{RNN}^{(k)}[\mathcal{D}_{ut-1}^{(k)}, \mathcal{D}_{ut}^{(k-1)}] + b_{\cdot}^{(k)})
\end{aligned} \quad (۵)$$

مشکل عبور بردار همسایگی پراکنده $u_{1,\ldots,t} = [a_{ut}, \ldots, a_{ut+i}]$ گره u به شبکه LSTM این است که پارامترهای مدل LSTM (مانند تعداد سلول‌های حافظه، تعداد واحدهای ورودی، واحدهای خروجی و ...) که باید آموخته شوند بسیار زیاد هستند. در عوض، شبکه LSTM ممکن است زمانی قادر به یادگیری بهتر نمایش باشد که بردار همسایگی پراکنده به یک نمایش با ابعاد پایین کاهش یابد. برای رسیدن به این هدف در [۶۲] مدلی مناسب پیشنهاد شده که در آن به‌جای انتقال بردار همسایگی پراکنده از یک کدگذار تمام‌متصل استفاده می‌شود تا نمایش پنهان با ابعاد پایین را بتوان به‌صورت زیر به‌دست آورد [۶۲]

$$\mathcal{D}_{ut}^{(p)} = f_a(\mathcal{W}_{AERNN}^{(p)} \mathcal{D}_{ut}^{(p-1)} + b^{(p)}) \quad (۶)$$

که در آن p لایه خروجی کدگذار تمام‌متصل را نشان می‌دهد و این نمایش سپس به شبکه‌های LSTM منتقل می‌شود [۶۲]

$$\begin{aligned}
\mathcal{D}_{ut}^{(p+1)} &= o_{ut}^{(p+1)} \times \tanh C_{ut}^{(p+1)} \\
o_{ut}^{(p+1)} &= \sigma_{ut}(\mathcal{W}_{AERNN}^{(p+1)}[\mathcal{D}_{ut-1}^{(p+1)}, y_{ut}^{(p)}] + b_o^{(p+1)})
\end{aligned} \quad (۷)$$

سپس بازنمایی پنهان تولیدشده توسط شبکه LSTM به یک کدگشای تمام‌متصل منتقل می‌شود. شکل ۱ معماری تعبیه گراف را برای شبکه‌های پیچیده پویا نشان می‌دهد.

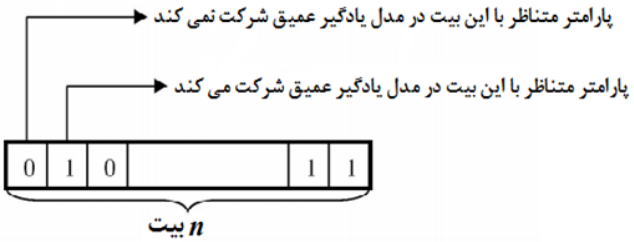
۳-۱-۲ بهینه‌سازی

در تابع استفاده‌شده در [۶۲] برای بهینه‌سازی پارامترهای تابع زیان تعریف‌شده در بالا از روش گرادیان با توجه به مقادیر کدگشا به‌صورت (۸) استفاده شده است

$$\frac{\partial \ell_t}{\partial \mathcal{W}_*^{(k)}} = [2(\hat{\mathcal{W}}_{t+1} - \mathcal{W}_{t+1}) \otimes \mathcal{B}] \frac{\partial f_a(Y^{(k-1)} \mathcal{W}_*^{(k)} + b^{(k)})}{\partial \mathcal{W}_*^{(k)}} \quad (۸)$$

که در آن $\mathcal{W}_*^{(k)}$ یک ماتریس وزن برای مدل می‌باشد. برای این مدل، گرادیان‌ها بر اساس واحدهای عصبی تکثیر می‌شوند تا مشتقات برای همه لایه‌های قبلی به‌دست آورده شوند. بعد از به‌دست‌آوردن مشتقات، مدل با استفاده از نزول گرادیان تصادفی (SGD) با روش تخمین لحظه انطباقی بهینه‌سازی خواهد شد [۶۲]. در این روش علاوه بر پیچیدگی محاسباتی بالا، سرعت بهینه‌نمودن پارامترها به کندی انجام خواهد شد. بنابراین در این بخش برای بهینه‌نمودن پارامترهای مدل LSTM عمیق از الگوریتم ژنتیک استفاده خواهد شد. الگوریتم ژنتیک در عین سادگی، قادر است با

1. Regulate



شکل ۲: بردار دودویی n بعدی شامل یک کروموزوم از جمعیت برای انتخاب یا عدم انتخاب پارامترهای مدل.

سرعت قابل توجهی، پارامترهای LSTM عمیق را بهینه نماید. مراحل الگوریتم ژنتیک برای بهینه‌نمودن پارامترهای مدل در ادامه آمده است. شکل ۲ یک کروموزوم را نشان می‌دهد که بیت‌های ۱ نشان‌دهنده شرکت پارامتر در مدل و بیت‌های ۰ نشان‌دهنده عدم شرکت پارامتر در مدل است.

کدکردن کروموزوم: هر کروموزوم شامل مجموعه‌ای از پارامترهای مربوط به یک شبکه عصبی عمیق شامل تعداد دوره‌ها، تعداد نورون‌های هر لایه و وزن‌های لایه توجه است.

مقداردهی اولیه جمعیت: جمعیت اولیه کروموزوم‌ها به‌طور تصادفی در بازه [۰,۱] ایجاد می‌شود.

لایه توجه: وزن‌های لایه توجه در هر کروموزوم روی ورودی‌های مسئله اعمال می‌شوند.

ایجاد مدل: در این مرحله به‌ازای هر کدام از اعضای جمعیت با توجه به تعداد دوره‌ها و تعداد نورون‌های لایه پنهان، یک مدل ایجاد می‌شود و با استفاده از مجموعه آموزش و با روش پس‌انتشار خطا آموزش می‌بینند که مدل به‌کاررفته در معماری پیشنهادی شبکه LSTM عمیق با سه لایه است.

انتخاب: به‌منظور انتخاب والدین برای انجام عمل ترکیب از مکانیزم انتخاب بر اساس نخبگی استفاده می‌شود.

اعمال عملگر تقاطع و جهش: بعد از انتخاب کروموزوم‌های والد، عملگرهای تقاطع و جهش برای ایجاد جمعیت جدید اعمال می‌شود.

معیار توقف: معیار توقف در الگوریتم، رسیدن به تعداد معینی از تکرارها است. در صورت اتمام الگوریتم، سه تا از بهترین کروموزوم‌های خروجی در نظر گرفته می‌شوند.

۳-۲ خوشه‌بندی جمعی پیشنهادی

در این مرحله، گراف تعبیه‌شده از مرحله قبل به یک مدل اجماع مبتنی بر خرد جمعی داده می‌شود. خوشه‌بندی جمعی فرایندی است که در آن چندین خوشه‌بندی پایه با هم ترکیب می‌شود.

روند اجرای الگوریتم بدین گونه است که ابتدا اولین الگوریتم پایه اجرا شده و نتیجه آن پس از ارزیابی پراکندگی (باید توجه داشت چون اولین الگوریتم است هیچ الگوریتمی جهت ارزیابی درجه استقلال در بخش الگوریتم‌های انتخاب‌شده وجود ندارد و نتیجه ارزیابی استقلال کاملاً مستقل خواهد بود) به بخش الگوریتم‌های انتخاب‌شده که ما آن را با عنوان جامعه خردمند می‌شناسیم اضافه می‌گردد. سپس نوبت الگوریتم بعدی است که پس از تولید نتیجه به ارزیابی درجه استقلال و میزان پراکندگی آن می‌پردازیم و در صورتی که نتایج ارزیابی از میزان آستانه تعیین‌شده بیشتر باشد افراز تولیدشده به داخل جامعه خردمند اضافه خواهد شد. در این روش در صورت رد نتیجه به‌دست‌آمده در هر بخش، فرایند ارزیابی نتیجه الگوریتم متوقف شده و به سراغ الگوریتم پایه بعدی خواهیم رفت. در این تحقیق برخلاف روش‌های پیشین خوشه‌بندی ترکیبی، کل افراز به‌دست آمده از یک الگوریتم خوشه‌بندی پایه را در صورت داشتن شرایط لازم وارد مجمع می‌کنیم و این گونه اصالت جواب حفظ می‌شود. مهم‌ترین تفاوت این روش با روش‌های قبلی را می‌توان موارد زیر دانست:

- اولین تفاوت این روش نحوه ارزیابی الگوریتم خوشه‌بندی است که در این روش پس از اجرای هر الگوریتم پایه، استقلال و پراکندگی آن نسبت به سایر الگوریتم‌های داخل مجمع محاسبه می‌شود و در صورت داشتن شرایط، وارد مجمع می‌شود.
- دوم اینکه الگوریتم‌ها در اینجا به‌طور غیرمتمرکز عمل می‌کنند و الگوریتمی که بدون کیفیت، پراکندگی و استقلال باشد قادر به ورود در مجمع نیست؛ لذا خطاهای غیرهم‌جهت به‌وجودآمده در این روش حذف شده و آثار جواب‌های هم‌جهت در مجمع بر روی هم افزوده خواهند شد که این کاملاً منطبق بر اصول حاکم بر خرد جمعی است.
- سوم اینکه چون بعد از رسیدن جمعیت مجمع به تعداد مورد نظر ما هیچ گزینش دیگری برای تولید جواب نهایی نیاز نیست، کیفیت نتایج نهایی حفظ می‌شود.

در مورد پراکندگی آرا باید گفت چون ما در خوشه‌بندی جمعی با داده‌ها و نتایج خوشه‌بندی اولیه سروکار داریم، از واژه پراکندگی نتایج اولیه استفاده می‌کنیم و بر اساس این فرض و تعریف سورویکی از تنوع آرا آن را به این شکل بازنویسی می‌نماییم: «هر الگوریتم خوشه‌بندی پایه باید جداگانه و بدون واسطه به داده‌های مسئله دسترسی داشته و آن را تحلیل و خوشه‌بندی کند؛ حتی اگر نتایج آن غلط باشد.» در اینجا نتایج غلط موجب کشف عدم تنوع و جلوگیری از تکرار یک جواب خاص خواهد شد. در این تحقیق برای محاسبه مقدار پراکندگی یک خوشه از معیار AAPMM [۳۹] استفاده می‌شود؛ چون این معیار هم از لحاظ پیچیدگی زمانی سریع‌تر از NMI است و هم مشکل تقارن آن را ندارد. معیاری که این تحقیق جهت سنجش پراکندگی نتیجه افراز یک خوشه‌بندی پایه استفاده کرده است A۳ نام دارد که میانگین وزن‌دار AAPMM می‌باشد که به شرح زیر است [۳۹]

$$A۳(p)=\frac{۱}{n}\sum_{i=۱}^{k} n_i \times AAPMM(C_i) \qquad (۹)$$

که n_i تعداد اعضای خوشه C_i، n تعداد اعضای کل خوشه‌ها و k تعداد افرازهای الگوریتم پایه می‌باشد. در این تحقیق مطابق [۳۹] مقدار آستانه T برای سنجش میزان پراکندگی الگوریتم خوشه‌بندی استفاده می‌شود که همواره بین صفر و یک است. پراکندگی از (۱۰) محاسبه خواهد شد

$$Diversity(P_i)=۱-A۳(P_i) \qquad (۱۰)$$

بنابراین مطابق با تعاریف بالا یکی از شرایط ورود، نتیجه یک خوشه‌بندی به مجمع (۱۱) است که شرط پراکندگی نام دارد

$$Diversity(C)\geq T \qquad (۱۱)$$

نویسندگان در [۳۹] نشان داده‌اند که هرچه میزان پراکندگی در میان خوشه‌بندهای اولیه بیشتر باشد، نتایج تولیدشده نیز از دقت بیشتری برخوردار هستند.

در این روش، نتیجه‌های خروجی خوشه‌بندهای پایه با استفاده‌کردن از یک تابع توافقی با هم ترکیب می‌شوند و به تولید نتیجه‌ای به مراتب بهتر از روش‌های خوشه‌بندی منفرد منجر می‌گردد. با توجه به یک

مجموعه داده $D=\{x_1^d, x_2^d, \ldots, x_n^d\}$ که در آن x_n^i، iامین نقطه در یک فضای ویژگی dبعدی می‌باشد، مجموعه‌ای از راه حل‌های خوشه‌بندی $C=\{C_1, C_2, \ldots, C_m\}$ که از m الگوریتم خوشه‌بندی ضعیف به‌دست می‌آیند، یک اجماع نامیده می‌شوند. هر راه حل $C_i=\{C_j^i, C_j^2, \ldots, C_j^{\kappa(j)}\}$، پارتیشنی از داده‌ها درون $\kappa(j)$ خوشه و C_j^i خوشه‌های نشان‌دهنده خوشه i از پارتیشن jام است [۱۲].

ایده اصلی زیربنای چارچوب پیشنهادی، استفاده از دو اصل جدید به‌منظور افزایش کارایی خوشه‌بندی جمعی است. اولین مورد، انتخاب بهترین زیرمجموعه از خوشه‌ها به‌جای استفاده از همه نتایج اولیه است. همان طور که در بخش قبل بیان گردید، نشان داده شده که استفاده از زیرمجموعه‌ای از خوشه‌های با کیفیت بالا نتایج بهتری نسبت به خوشه‌های کامل در اجماع دارد. در چارچوب ارائه‌شده، ابتدا یک ماتریس شباهت[1] ایجاد می‌گردد. اطلاعات خوشه‌های انتخاب‌شده در یک ماتریس شباهت $n\times n$ انباشته شده و شباهت دو نقطه داده با تعداد دفعاتی که هر دو نمونه داده در یک خوشه گروه‌بندی می‌شوند اندازه‌گیری می‌گردد. از آنجا که این معیار، هم‌زمانی نمونه‌ها را محاسبه می‌کند به آن ماتریس هم‌همبستگی[2] نیز می‌گویند. خوشه‌ها بر اساس یک تابع توافقی [۳۹] با هم ترکیب خواهند شد. دوم استفاده از مفهوم پیمانه‌ای‌بودن در شبکه‌های پیچیده می‌باشد که توسط محققین در [۶۳] معرفی شده است. به حداکثر رساندن پیمانه‌ای‌بودن از طریق شبکه در طیف وسیعی از مطالعات [۶۳] تا [۶۵] نشان داده شده که منجر به یافتن انجمن‌های با پیمانه بالا می‌شود. این واقعیت ما را برانگیخت تا مسئله انتخاب خوشه‌بندی جمعی را به مسئله حداکثرسازی پیمانه‌ای تبدیل کنیم تا از این معیار به‌خوبی استفاده نماییم. با درنظرگرفتن شکاف بین این مسائل، یک رابطه جدید برای به حداکثر رساندن معیار پیمانه‌ای، معرفی و سپس یک الگوریتم جستجوی حریصانه جدید برای بهینه‌سازی تابع هدف در زمان بسیار سریع ارائه می‌گردد. ابتدا مجموعه‌ای از m پارتیشن اولیه با استفاده از الگوریتم‌های خوشه‌بندی ضعیف ارائه شده است. به‌دلیل ساده‌بودن روش c-means فازی که نتایج اولیه متفاوتی را به‌همراه دارد، از الگوریتم c-means برای این کار استفاده کردیم. سپس کیفیت کل خوشه‌های فردی موجود در اجماع با استفاده از معیار Rank محاسبه می‌شود. در مرحله بعد، جفت (خوشه؛ کیفیت) داریم که کیفیت هر خوشه مقدار Rank آن است. ما به سادگی یک آستانه را بر روی ارزش کیفیت هر خوشه اعمال می‌کنیم و آنهایی را که بالاتر از آستانه هستند برای شرکت در مجمع تصمیم‌گیری نهایی برمی‌گزینیم.

۳-۲-۱ معیار شباهت

در خوشه‌بندی جمعی، هر خوشه‌بندی پایه از تعداد معینی خوشه پایه تشکیل شده است. برای به‌دست‌آوردن اطلاعات خوشه پایه، یک راهکار معمول این است که برچسب‌های خوشه پایه را به سطح شیء (یا سطح قطعه[3] [۶۶]) با ساخت یک ماتریس همبستگی، نگاشت کرد. ماتریس همبستگی، چگونگی هم‌رخدادی دو شیء را که در یک خوشه مشابه در میان چندین خوشه‌بند پایه گروه‌بندی می‌شوند، منعکس می‌کند. نگاشت مستقیم از برچسب‌های خوشه پایه به ماتریس همبستگی شیء به‌طور ضمنی فرض می‌کند که خوشه‌های مختلف مستقل از یکدیگر هستند؛ اما نگاشت مستقیم، اطلاعات غنی پنهان در رابطه بین خوشه‌های مختلف را در نظر نمی‌گیرد. برای این منظور، ابتدا باید دو مشکل فرعی در اینجا حل شود: ۱) چگونگی تعریف شباهت اولیه بین خوشه‌ها و ۲) نحوه ترکیب اطلاعات برای ایجاد شباهت خوشه‌ای قوی‌تر. از آنجایی که یک خوشه مجموعه‌ای از اشیای داده است، رابطه اولیه بین خوشه‌ها را می‌توان با ضریب جاکارد [۶۶] بررسی نمود که شباهت بین دو مجموعه را با درنظرگرفتن اندازه اشتراک و اندازه اجتماع آنها اندازه‌گیری می‌کند. به‌طور رسمی، ضریب جاکارد بین دو خوشه $\mathcal{C}_i$ و $\mathcal{C}_j$ به‌صورت (۱۲) محاسبه می‌شود [۶۶]

$$Jaccard(\mathcal{C}_i,\mathcal{C}_j)=\frac{\left|\mathcal{C}_i\cap\mathcal{C}_j\right|}{\left|\mathcal{C}_i\cup\mathcal{C}_j\right|} \qquad (۱۲)$$

در (۱۲)، علامت $\cap$، اشتراک دو خوشه و علامت $\cup$، اجتماع دو خوشه را نشان می‌دهد. اگر یک گراف $\mathcal{G}=(\mathcal{V},\mathcal{E})$ را در نظر بگیریم، آنگاه $\mathcal{V}=\mathcal{C}$ را مجموعه گره‌های گراف و $\mathcal{E}$ مجموعه یال‌های گراف را تشکیل می‌دهند. وزن یک یال مابین دو گره $\mathcal{C}_i$ و $\mathcal{C}_j$ $(\mathcal{C}_i,\mathcal{C}_j\in\mathcal{V})$، در معیار تشابه جاکارد از (۱۳) محاسبه می‌شود

$$\mathcal{W}_{ij}=\frac{\mathcal{C}_i\mathcal{C}_j}{\left\|\mathcal{C}_i\right\|_2^2+\left\|\mathcal{C}_j\right\|_2^2-\mathcal{C}_i\mathcal{C}_j} \qquad (۱۳)$$

با ساخت گراف تشابه اولیه، گام بعدی این است که اطلاعات چندمقیاسی[4] در گراف گنجانده شود تا شباهت خوشه‌ای افزایش یابد. فرایند قدم‌زنی تصادفی، فرایندی پویاست که روی گراف انجام می‌شود. اکثر رویکردهای تعبیه‌سازی بر نمایش همسایگی محلی گره‌ها تمرکز می‌کنند و نمی‌توانند ساختار گراف سراسری را به تصویر بکشند؛ یعنی روابط را با گره‌های دور حفظ کنند. قدم‌زنی تصادفی، یکی از روش‌هایی است که توانسته بر این مشکل فایق آید. فرایند قدم‌زنی تصادفی در هر مرحله با احتمال معینی از یک گره به یکی از همسایگان خود منتقل می‌شود. در روش قدم‌زنی تصادفی باید ماتریس احتمال انتقال ساخته شود. ماتریس احتمال انتقال، احتمال قدم‌زنی تصادفی بین دو گره را مشخص می‌کند. در این مقاله، ماتریس احتمال انتقال $\mathcal{P}=\{p_{ij}\}_{\mathcal{N}\times\mathcal{N}}$ بر روی گراف به‌صورت (۱۴) محاسبه می‌شود

$$p_{ij}=\begin{cases}\dfrac{\mathcal{W}_{ij}}{\sum_{\mathcal{C}_i\neq\mathcal{C}_j}\mathcal{W}_{ij}} & \text{if } i\neq j\\ \cdot & \text{if } i=j\end{cases} \qquad (۱۴)$$

که در آن p_{ij} احتمال انتقال یک قدم‌زنی تصادفی از گره $\mathcal{C}_i$ به گره $\mathcal{C}_j$ در یک مرحله است که متناسب با وزن یال بین آنهاست. بر اساس ماتریس احتمال انتقال تک‌مرحله‌ای می‌توان ماتریس احتمال انتقال چندمرحله‌ای $\mathcal{P}^{(t)}=\{p_{ij}^{(t)}\}_{\mathcal{N}\times\mathcal{N}}$ را برای قدم‌زنی‌های تصادفی روی گراف به‌دست آورد

$$\mathcal{P}^{(t)}=\begin{cases}\mathcal{P} & \text{if } t=1\\ \mathcal{P}^{(t+1)} & \text{if } t>1\end{cases} \qquad (۱۵)$$

iامین و jامین مدخل در $\mathcal{P}^{(t)}$ با $\mathcal{P}_{ij}^{(t)}$ (احتمال انتقال یک قدم‌زنی تصادفی از گره $\mathcal{C}_i$ و $\mathcal{C}_j$ در مرحله t) نشان داده شده است. از آنجایی که طول گام‌های مختلف قدم‌زنی‌های تصادفی می‌توانند اطلاعات ساختار گراف را در مقیاس‌های مختلف منعکس کنند، برای ثبت اطلاعات

1. Similarity Matrix
2. Co-association Matrix
3. Fragment
4. Multi-Scale Information

چندمقیاسی در گراف $\mathcal{G}$ از مسیرهای قدم‌زنی تصادفی در مراحل مختلف برای اصلاح شباهت خوشه‌ای استفاده می‌شود. به‌طور رسمی برای قدم‌زنی تصادفی که از یک گره $\mathcal{C}_i$ شروع می‌شود، مسیر حرکت تصادفی آن از مرحله ۱ تا مرحله t به‌صورت $\mathcal{P}_{i:}^{1:t}=\{\mathcal{P}_{i:}^{(1)},\mathcal{P}_{i:}^{(2)},\dots,\mathcal{P}_{i:}^{(t)}\}$ نشان داده می‌شود. بدیهی است که مسیر قدم‌زنی تصادفی tمرحله‌ای (یعنی $\mathcal{P}_{i:}^{1:t}$) که از گره $\mathcal{C}_i$ شروع می‌شود و دارای طول گام t است، یک تاپل $\mathcal{N}.t$ با مقیاس چندگانه (یا اطلاعات ساختاری چندمرحله‌ای) در همسایگی $\mathcal{C}_i$ است. با مسیر قدم‌زنی تصادفی هر گره به‌دست‌آمده و درنظرگرفتن شباهت مسیرهای قدم‌زنی تصادفی آنها می‌توان یک معیار مشابهت برای هر دو گره به‌دست آورد. به‌طور خاص، ماتریس شباهت جدید بین همه خوشه‌های Π موجود به‌صورت (۱۶) نمایش داده می‌شود

$$Z=\{z_{ij}\}_{\mathcal{C}_n\times\mathcal{C}_n}$$
$$z_{ij}=Sim(\mathcal{C}_i,\mathcal{C}_j) \qquad (۱۶)$$

که z_{ij} نشان‌دهنده شباهت جدید بین دو خوشه $\mathcal{C}_i$ و $\mathcal{C}_j$ است. پس از به‌دست‌آوردن ماتریس تشابه خوشه‌ای Z، ماتریس شباهت جدید از سطح خوشه به سطح شیء نگاشت می‌شود. برای به‌دست‌آوردن ماتریس تشابه خوشه‌ای Z، ماتریس شباهت از سطح خوشه به سطح شیء طبق (۱۷) و (۱۸) نگاشت می‌شود

$$z_{ij}=Sim(\mathcal{P}_{i:}^{1:t},\mathcal{P}_{j:}^{1:t}) \qquad (۱۷)$$

$$Sim(\mathcal{P}_{i:}^{1:t},\mathcal{P}_{j:}^{1:t})=\frac{\langle\mathcal{P}_{i:}^{1:t},\mathcal{P}_{j:}^{1:t}\rangle}{\sqrt{\langle\mathcal{P}_{i:}^{1:t},\mathcal{P}_{j:}^{1:t}\rangle.\langle\mathcal{P}_{i:}^{1:t},\mathcal{P}_{j:}^{1:t}\rangle}} \qquad (۱۸)$$

که در آن $\langle\mathcal{P}_{i:}^{1:t},\mathcal{P}_{j:}^{1:t}\rangle$ حاصل‌ضرب نقطه‌ای دو بردار $\mathcal{P}_{i:}^{1:t}$ و $\mathcal{P}_{j:}^{1:t}$ را در خروجی می‌دهد. از آنجایی که ورودی‌های ماتریس احتمال انتقال همیشه غیرمنفی هستند، به‌ازای هر دو خوشه $\mathcal{C}_i$ و $\mathcal{C}_j$ در همه خوشه‌های Π موجود، $z_{ij}\in[۰,۱]$ است.

۳-۲-۲ ایجاد ماتریس همبستگی و تابع توافقی

در این مقاله از یکی از جدیدترین ایده‌های موجود برای تابع توافقی استفاده شده که در آن اطلاعات خوشه‌های انتخاب‌شده در یک ماتریس شباهت $n\times n$ تجمیع می‌شوند [۴۳]. با الهام از مسئله مربوط در زمینه شبکه اجتماعی (برای تشخیص انجمن در شبکه‌های پیچیده) می‌توان این ماتریس شباهت را به‌عنوان یک شبکه وزن‌دار بیان کرد. با این دیدگاه که شبکه‌های پیچیده دنیای واقعی در حال تغییر هستند، الگوریتم‌های تشخیص انجمن شبکه را می‌توان برای حل مسئله تابع توافقی به کار گرفت. تابع توافقی پیشنهادی، فرایند تقسیم‌بندی را در سطح خوشه انجام می‌دهد که از ماتریس شباهت خوشه‌ای تقویت‌شده Z بهره می‌برد و همه خوشه‌های مجمع را به چندین زیرمجموعه گروه‌بندی می‌کند. به هر زیرمجموعه از خوشه‌ها، متاخوشه گفته می‌شود. سپس هر شیء داده با رأی اکثریت به یکی از متاخوشه‌ها اختصاص داده می‌شود تا اجماع نهایی ساخته شود. به‌طور خاص با درنظرگرفتن خوشه‌ها در مجمع به‌عنوان گره‌های گراف و استفاده از ماتریس تشابه خوشه‌ای Z برای تعریف وزن‌های یال بین آنها می‌توان یک گراف شباهت خوشه جدید ساخت؛ به این معنا که $\hat{\mathcal{G}}=\{\mathcal{V},\hat{\mathcal{E}}\}$ که در آن $\mathcal{V}=\mathcal{C}$ مجموعه گره‌ها و $\hat{\mathcal{E}}$ مجموعه یال‌هاست. وزن یال در گراف $\mathcal{G}$ توسط ماتریس شباهت خوشه‌ای تقویت‌شده $\mathcal{B}$ تعیین می‌شود. با توجه به دو خوشه $\mathcal{C}_i$ و $\mathcal{C}_j$، وزن بین آنها به‌صورت (۱۹) تعریف می‌گردد

$$\hat{w}_{ij}=l_{ij} \qquad (۱۹)$$

سپس الگوریتم برش نرمال‌شده[1] می‌تواند برای تقسیم گراف جدید به تعداد معینی از متاخوشه‌ها استفاده شود؛ یعنی

$$\mathcal{MetaCl}=\{\mathcal{MetaCl}_۱,\mathcal{MetaCl}_۲,\dots,\mathcal{MetaCl}_\kappa\} \qquad (۲۰)$$

که در آن $\mathcal{MetaCl}_i$ متاخوشه و κ تعداد متاخوشه است. توجه داشته باشید که یک متاخوشه از تعداد مشخصی خوشه تشکیل شده و با توجه به یک شیء x_i و یک متاخوشه $\mathcal{MetaCl}_j$، شیء x_i ممکن است که به‌صورت صفر یا چند خوشه در داخل $\mathcal{MetaCl}_j$ ظاهر شود. به‌طور مشخص، امتیاز رأی x_i متاخوشه $\mathcal{MetaCl}_j$ را می‌توان به‌عنوان نسبت خوشه‌های موجود در $\mathcal{MetaCl}_j$ که حاوی شیء x_i هستند تعریف کرد. به این معنا که

$$Rank(x_i,\mathcal{MetaCl}_j)=$$
$$\frac{۱}{\left|\mathcal{MetaCl}_j\right|}\sum_{\mathcal{C}_i\in\mathcal{MetaCl}_j}\mathfrak{T}(x_i\in\mathcal{MetaCl}_j) \qquad (۲۱)$$
$$\mathfrak{T}(x_i\in\mathcal{MetaCl}_j)=\begin{cases}۱ & \text{, if } x_i\in\mathcal{MetaCl}_j \text{ is true}\\ ۰ & \text{, otherwise}\end{cases}$$

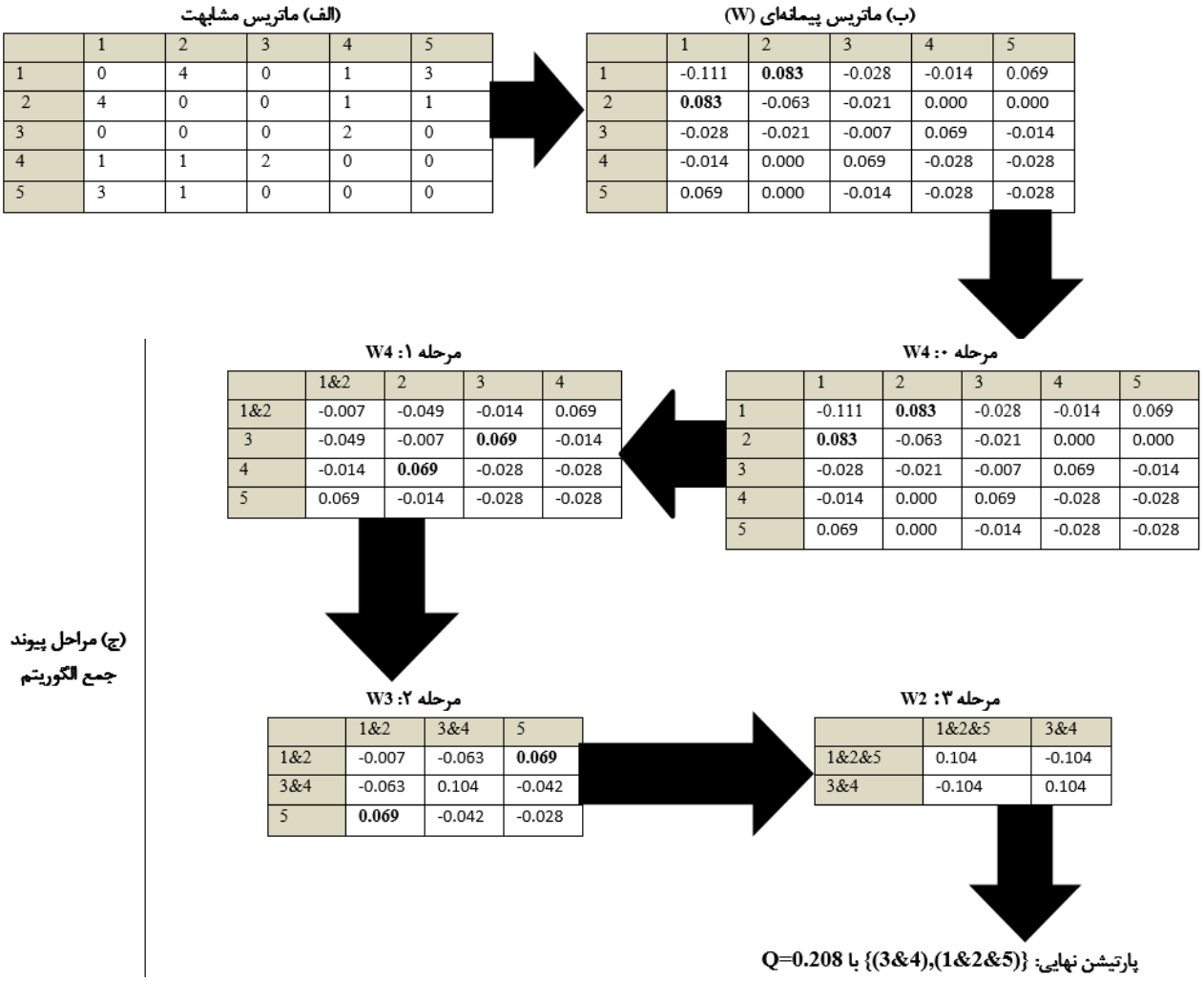
که در آن $\left|\mathcal{MetaCl}_j\right|$ تعداد خوشه‌ها را در $\mathcal{MetaCl}_j$ نشان می‌دهد. سپس با رأی اکثریت، هر شیء به متاخوشه‌ای که در آن بیشتر ظاهر می‌شود (یعنی با بالاترین امتیاز رأی) اختصاص داده می‌شود؛ به این معنا که

$$(Meta_\varphi)_{x_i}=$$
$$\arg\max_{\mathcal{MetaCl}_j\in\mathcal{MetaCl}}Rank(x_i,\mathcal{MetaCl}_j) \qquad (۲۲)$$

اگر یک شیء x_i همان بالاترین امتیاز رأی را از دو یا چند متاخوشه مختلف به‌دست آورد (که در عمل به‌ندرت اتفاق می‌افتد)، آنگاه شیء به‌طور تصادفی به یکی از متاخوشه‌های برنده اختصاص داده می‌شود. با اختصاص هر شیء به یک متاخوشه از طریق رأی اکثریت و درنظرگرفتن اشیا در همان متاخوشه به‌عنوان یک تابع توافقی، نتیجه نهایی خوشه‌بندی جمعی را می‌توان به‌دست آورد.

فرض کنید که ماتریس تشابه اندازه همان طور که در شکل ۳- الف آمده، از تشابه جاکارد مشتق شده است. هر ورودی $a(i,j)$ در این ماتریس شباهت نشان‌دهنده یک یال متصل گره i و j در گراف مربوط است که در شکل ۳-الف نشان داده شده است. از آنجایی که بدیهی است که هر نمونه داده با خودش در یک خوشه قرار دارد، بدون ازدست‌دادن کلیت مسئله، قطر اصلی ماتریس شباهت را می‌توان روی صفر تنظیم کرد؛ یعنی گره‌ها در گراف شباهت به خودشان حلقه ندارند. با توجه به یک ماتریس شباهت، ماتریس پیمانه‌ای به سادگی با کم‌کردن وزن مورد انتظار یال بین گره‌های i و j از وزن واقعی آن $a(i,j)$ محاسبه می‌شود و بنابراین یک مقدار مثبت در ماتریس پیمانه‌ای نشان می‌دهد که یال مربوط، وزنی بیش از حد انتظار دارد. شکل ۳- ب ماتریس پیمانه‌ای را نشان می‌دهد. ماتریس پیمانه‌ای به‌عنوان ورودی به الگوریتم پیوند مجموع (مرحله ۰) وارد می‌شود. در هر مرحله، حداکثر مقدار در ماتریس پیدا می‌شود و شاخص‌های سطر و ستون آن به‌عنوان خوشه‌ای تجمیع‌شده ادغام می‌شوند. اینجا ورودی $W_۵(۱,۲)=W_۵(۲,۱)=۰/۰۸۳$ بزرگ‌ترین مقدار در مرحله ۰ است (شکل ۳- ج)؛ به آن معنا که خوشه‌های نمایه‌شده با ۱ و ۲ باید ادغام شوند. حال سؤال این است که با ادغام این دو خوشه،

1. Normalized Cut



شکل ۳: مثالی برای نشان‌دادن مراحل روش پیشنهادی.

ماتریس چگونه باید به‌روز شود. به‌عنوان مثال، مقدار جدید $W_۴(۱\&۲,۴)$ چگونه محاسبه می‌شود. دو ورودی $W_۵(۱,۴)$ و $W_۵(۲,۴)$ را به‌عنوان مقدار جدید $W_۴(۱\&۲,۴)$ اضافه می‌کنیم و بنابراین طبق (۲۳) داریم

$$W_۴(۱\&۲,۴) = W_۵(۱,۴) + W_۵(۲,۴) = \\ -۰/۰۱۴ + ۰/۰۰۰ = -۰/۰۱۴ \quad (۲۳)$$

ورودی‌هایی که باید در این مرحله به‌روز شوند با رنگ خاکستری در مرحله ۰ شکل ۳- ج مشخص گردیده‌اند و همین رویه برای ماتریس به‌روزشده $W_۴$ تکرار می‌شود و بعد از چهار مرحله به پارتیشن دوخوشه‌ای می‌رسد.

پس از اینکه ماتریس همبستگی ساخته شد، در مرحله بعد از یک تابع توافقی برای استخراج خوشه‌های نهایی از این ماتریس استفاده می‌شود. ماتریس همبستگی، ماتریس شباهت $n \times n$ است که هم‌رخدادی جفت الگوها را در یک خوشه به‌عنوان رأی برای ارتباط آنها می‌گیرد. پس از آن، ماتریس همبستگی به ماتریس پیمانه‌ای تبدیل می‌شود تا بتوان روش‌های حداکثرسازی پیمانه‌ای را بر روی آن پیاده‌سازی کرد. با دانستن تعداد خوشه‌ها در اکثر وظایف خوشه‌بندی، یک برنامه عدد صحیح درجه دوم جدید را برای حل مسئله معرفی می‌کنیم. نهایتاً یک رویکرد حریصانه جدید برای حل مؤثر مدل و کشف خوشه‌های نهایی ارائه می‌شود.

۳-۲-۳ به حداکثر رساندن معیار پیمانه‌ای

یادآور می‌شود که ماتریس همبستگی مشتق‌شده، یک ماتریس شباهت و هدف در این مرحله، استخراج خوشه‌های نهایی از آن است. با توجه به ماتریس شباهت نقاط داده می‌توان آن را به‌عنوان یک گراف یا شبکه وزن‌دار مشاهده کرد. این تغییر دیدگاه به ما امکان می‌دهد الگوریتم‌های تشخیص انجمن شبکه را به‌عنوان تابع توافقی در مسئله خوشه‌بندی جمعی استفاده کنیم. در زمینه تشخیص انجمن شبکه، تعدادی الگوریتم برای استخراج مؤثر انجمن‌ها (خوشه‌ها) از یک شبکه ورودی وجود دارد [۶۶]. یکی از محبوب‌ترین الگوریتم‌ها، روش بهینه‌سازی مبتنی بر پیمانه است [۶۴]. روش پیمانه‌ای‌بودن نیومن- گیروان [۶۴] که ابتدا در ۲۰۰۴ معرفی گردید، به‌سرعت به عنصر ضروری بسیاری از روش‌های تشخیص انجمن‌ها تبدیل شده است. پیمانه‌ای‌بودن تا حد زیادی مورد استفاده‌ترین و شناخته‌شده‌ترین تابع کیفیت به‌منظور ارزیابی خروجی الگوریتم‌های تشخیص انجمن‌هاست.

نماد استفاده‌شده برای تعریف پیمانه‌ای‌بودن به‌صورت زیر ارائه می‌شود: ماتریس $\Pi = \pi_{ij}$ فرضیه صفر تعداد یال‌های مورد انتظار (وزن مورد انتظار) بین هر دو گره i و j را نگه می‌دارد که $\pi_{ij} = \kappa_i \kappa_j / K$ است. درجه یک گره i با κ_i و مجموع درجات همه گره‌ها با $K = \sum_{i=۱}^{n} \kappa_i$ نشان داده می‌شود. پیمانه‌ای‌بودن یک پارتیشن γ، وزن کل یال‌های داخل خوشه‌ها منهای فرضیه صفر آن یال‌ها می‌باشد. به‌منظور مقایسه پیمانه‌ای‌بودن گراف‌های با اندازه‌های مختلف، بهتر است که این اختلاف را با یک ضریب $۱/K$ نرمال نمود؛ بنابراین پیمانه‌ای‌بودن یک عدد در بازه $[-۱,۱]$ نرمال می‌شود. فرض کنید γ بردار به طول n است که γ شامل مقادیر صحیح $Y = \{\gamma_۱,\ldots,\gamma_n\}$، پارتیشنی است که γ_i برچسب

خوشه iامین نقطه داده را در $\gamma_i = \{۱,\dots,\kappa\}$ نشان می‌دهد که κ نشان‌دهنده تعداد خوشه‌ها می‌باشد. مثلاً $\gamma_۵ = ۳$ نشان‌دهنده این است که نمونه داده $\chi_۵$ متعلق به خوشه ۳ است و در نتیجه، پیمانه‌ای‌بودن یک پارتیشن γ را می‌توان به‌صورت (۲۴) نوشت

$$M_\gamma = \frac{۱}{K}\sum_{i,j}(\alpha_{i,j} - \pi_{i,j})\Delta(\gamma_i,\gamma_j) = \frac{۱}{K}\sum_{i,j}(\alpha_{i,j} - \frac{\kappa_i\kappa_j}{\mathrm{K}})\Delta(\gamma_i,\gamma_j) \quad (۲۴)$$
s.t.
$$\gamma_i \in \{۱,۲,..,\kappa\} \ , \ i,j \in \{۱,۲,\dots,n\}$$

که در آن Δ، دلتای کرونکر[1] را نشان می‌دهد که اگر آرگومان‌های آن یکسان باشد، مقدار آن ۱ و در غیر این صورت ۰ است. $M=۰$ به آن معناست که پارتیشن γ، یال درون خوشه‌ای بیش از حد انتظار تصادفی نمی‌دهد. مقادیر بیشتر از ۰ نشان‌دهنده انحراف از تصادفی‌بودن است. در عمل مقادیر بیشتر از حدود ۰/۳ به نظر می‌رسد که پارتیشن‌بندی قابل توجهی را نشان می‌دهد. برای ساده‌سازی، ماتریس W را با ورودی‌های $w_{ij} = \frac{۱}{K}(\alpha_{i,j} - \frac{\kappa_i\kappa_j}{K})$ به‌عنوان ماتریس پیمانه A می‌گوییم؛ بنابراین معادله پیمانه‌ای به‌صورت (۲۵) ساده می‌شود

$$M_\gamma = \sum_{i,j} w_{i,j}\Delta(\gamma_i,\gamma_j) \quad (۲۵)$$
s.t.
$$\gamma_i \in \{۱,۲,\dots,\kappa\} \ , \ i,j \in \{۱,۲,\dots,n\}$$

توجه به این نکته ضروری است که وقتی تعداد خوشه‌ها (κ) ناشناخته است، می‌توان آن را با n جایگزین کرد. هدف مسئله در این مرحله، یافتن گروه‌های مجزا از گره‌ها است که مجموع ورودی‌های متناظر در ماتریس پیمانه‌ای $\mathcal{W}$ را به حداکثر برساند. رابطه پیمانه‌ای به‌سرعت تحقیقات علمی در زمینه تشخیص انجمن در شبکه‌های پیچیده را متحول کرد و آرایه‌ای از الگوریتم‌های اکتشافی را برای بهینه‌سازی پیمانه‌ای توسعه داد. از آنجایی که هیچ یک از الگوریتم‌های اکتشافی برای مسئله حداکثرسازی پیمانه‌ای نمی‌توانند تضمین کنند که پارتیشن بهینه را تولید می‌کند، رویکرد برنامه‌ریزی ریاضی یک مکتب فکری در حل دقیق مسئله پیدا کرده است. با این حال، اکثر این مطالعات یک فرمول برنامه‌ریزی خطی/ غیرخطی (ILP/INLP) ارائه می‌دهند که از نظر محاسباتی بسیار گران‌تر از آن است که توسط حل‌کننده‌های رایج ILP/INLP حل شود.

۳-۲-٤ جستجوی حریصانه سریع

اگرچه الگوریتم‌های تکاملی مانند الگوریتم ژنتیک یا شبیه‌سازی تبرید می‌توانند برای حل مسئله بیشینه‌سازی استفاده شوند، اما کارایی آنها به کیفیت عملگرهای ابداع‌شده‌ای بستگی دارد که حل‌کننده را با مسئله تطبیق می‌دهند. این طرح‌ها معمولاً کندتر از سایر روش‌های اکتشافی هستند. از سوی دیگر، یکی از سریع‌ترین مکانیسم‌های بهینه‌سازی، رویکرد حریصانه است. اگرچه این رویکرد، تضمینی برای رسیدن به بهینه سراسری نیست، اما در عمل معمولاً نتایج بهینه یا تقریباً بهینه را به همراه دارد. علاوه بر این، آنها معمولاً روش‌های سریعی هستند و در واقع، مهم‌ترین ویژگی یک الگوریتم حریصانه کارایی زمانی آن است. در اینجا یک رویکرد حریصانه برای حل مسئله بهینه‌سازی پیشنهاد می‌شود. فرض کنید که Z یک ماتریس $n\times\kappa$ باینری است که اگر $z_{i,\ell}=۱$ باشد، نشان می‌دهد که گره i متعلق به خوشه ℓ بوده و در غیر این صورت $z_{i,\ell}=۰$ است. ویژگی غیرهم‌پوشانی مسئله خوشه‌بندی بلافاصله محدودیت $\sum_{\ell=۱}^{\kappa} z_{i,\ell}=۱$ را ارضا می‌کند. به بیان ساده، این محدودیت به‌دلیل ویژگی باینری متغیرها در Z، هر گره را در یک خوشه مشاهده می‌نماید. در این رابطه گره i و j متعلق به ℓامین خوشه‌اند، اگر و فقط اگر $z_{i,\ell}\times z_{j,\ell}=۱$ باشد. به عبارت دیگر $z_{i,\ell}\times z_{j,\ell}=۱$ هم‌رخدادی هر جفت گره را در یک خوشه بررسی می‌کند. با توجه به تعداد خوشه‌های κ، مدل به‌صورت (۲۶) تعریف می‌شود

$$\mathcal{M}_{\mathcal{Z}} = \sum_{i=۱}^{n}\sum_{j=۱}^{n}\sum_{\ell=۱}^{\kappa} w_{i,j}\mathcal{Z}_{i,\ell}\mathcal{Z}_{j,\ell}$$
s.t.
$$\sum_{i=۱}^{n} z_{i,\ell} \geq ۱ \ , \ \sum_{j=۱}^{n} z_{j,\ell} = ۱ \quad (۲۶)$$
$$z_{j,\ell} \in [۰,۱] \ , \ i,j = \{۱,۲,\dots,n\} \ , \ \ell = \{۱,۲,\dots,\kappa\}$$

رابطه (۱۰) دو قید را برای همه i و ℓها تعریف می‌کند. از آنجایی که $i,j \in \{۱,۲,\dots,n\}$ و $\ell \in \{۱,۲,\dots,\kappa\}$ است، پیچیدگی قیود در این مدل $\mathrm{O}(MAX(n,\kappa)) = \mathrm{O}(n)$ می‌باشد. علاوه بر این، چون ماتریس متغیر Z دارای ابعاد $n\times\kappa$ است، پیچیدگی تعداد متغیرها $\mathrm{O}(n\kappa)$ می‌باشد. از آنجایی که $\kappa \ll n$ به $\mathrm{O}(n)$ بسیار نزدیک‌تر از $\mathrm{O}(n^۲)$ است و به‌طور دقیق‌تر می‌توان گفت کمی بیشتر از $\mathrm{O}(n)$ است، بنابراین این مدل همچنین در پیچیدگی متغیرها از رقبای خود بهتر عمل می‌کند. اگرچه این برنامه درجه دوم ۰-۱ یک موضوع قابل توجه در مدل‌سازی مسئله است، اما حل مسئله حداکثرسازی پیمانه‌ای از نوع NP- سخت می‌باشد. به‌طور کلی، حل‌کننده‌های برنامه درجه دوم می‌توانند برای معیارهای کوچک مورد استفاده قرار گیرند و برای مسائل مقیاس بزرگ، به‌ویژه مواردی که $n>۲۰۰$ هستند، کاربردی نیستند. در مرحله بعد الگوریتم حریصانه سریعی ارائه می‌شود که قادر است فضاهای بسیار بزرگ را در مدت زمان بسیار کوتاهی جستجو کند. الگوریتم پیشنهادی روش پایین به بالاست که همانند روش‌های پیوند تجمعی[2] کار می‌کند و با درنظرگرفتن هر نقطه داده به‌عنوان یک خوشه تکی شروع می‌شود. سپس دو خوشه را ادغام می‌کند که به نظر می‌رسد بهترین گزینه ادغام در این مرحله باشد و تا زمانی که دیگر خوشه‌های برای ترکیب وجود نداشته باشد، ادامه می‌یابد. این رویکرد به‌طور رسمی در شکل ۴ بیان می‌شود.

طرح بهینه‌سازی تجمعی پیشنهادی پس از تعدادی تکرار محدود متوقف می‌شود. در مرحله اول، $C_n^۲$ گزینه برای ادغام وجود دارد و پس از ادغام، گزینه‌های $C_{n-۱}^۲$ در مرحله دوم وجود دارد و در نتیجه، گزینه‌های $C_{\kappa+۱}^۲$ در آخرین مرحله $n-\kappa$ وجود دارد؛ بنابراین به‌طور کلی، $n-\kappa$ مرحله وجود دارد که $\sum_{i=\kappa+۱}^{n} C_i^۲$ گزینه برای انتخاب وجود دارد. تکنیک خوشه‌بندی پیوند تجمعی، یک ماتریس پیمانه‌ای $n\times n$ را دریافت می‌کند که نیاز به محاسبه و ذخیره مدخل $n^۲$ ورودی دارد. این فضا کاملاً قابل قبول و نیز از اکثر الگوریتم‌های خوشه‌بندی دیگر پایین‌تر است. زمان مورد نیاز برای پیوند تجمعی برابر با سایر الگوریتم‌های خوشه‌بندی سلسله‌مراتبی، یعنی $\mathrm{O}(n^۲\log n)$ است و بنابراین پیوند تجمعی را می‌توان به‌عنوان یکی از سریع‌ترین الگوریتم‌های خوشه‌بندی طبقه‌بندی کرد. شکل ۵ شمای کلی روش پیشنهادی را نمایش می‌دهد.

1. Kronecker Delta

2. Agglomerative Linkage Methods

Algorithm 1: Greedy Algorithm
Input: $Modularity_Matrix, \mathcal{W}, \kappa$
Output: final partition Y
1. $\mathcal{W}_n = \mathcal{W}$
2. $\varsigma = n$
3. Repeat
4. Find the maximum entry in $\mathcal{W}_\varsigma$ and set i and j to its row and column indices.
5. Merge the two clusters i and j as a new cluster $i \& j$.
6. Update the $Modularity_Matrix$ (derive $\mathcal{W}_{\varsigma-1}$ from $\mathcal{W}_\varsigma$) as follows:
7. foreach $\pi = 1$ to n
8. $\mathcal{W}_{\varsigma-1}(i \& j, \pi) + \mathcal{W}_\varsigma(j, \pi)$
9. end
10. $\varsigma - 1$
11. Until $\varsigma = \kappa$ (only κ clusters remains)
12 Return the merged indices in $\mathcal{W}_\kappa$ as the final clusters, Y.

شکل ۴: شبه‌کد رویکرد حریصانه پیشنهادی.

٤- آزمایش‌ها و نتایج تجربی

در این بخش، آزمایش‌های مختلفی بر روی روش پیشنهادی انجام گرفته و نتایج آن با سایر روش‌های رقیب مقایسه می‌گردد و همچنین مجموعه داده‌های مورد استفاده شرح داده شده است. تمامی آزمایش‌ها بر روی یک سیستم ۶۴بیتی Ubuntu ۱۶.۰۴.۱ LTS با پردازنده Intel (R) Core (TM) i۷ – ۷۹۰۰X، فرکانس ساعت پردازنده ۳٫۳۰ گیگاهرتز و ۶۴ گیگابایت حافظه انجام شده است. در این بخش برای انجام آزمایش‌های مختلف و بررسی عملکرد روش پیشنهادی، سناریوهای مختلفی در نظر گرفته شده است. در سناریوی اول، روش پیشنهادی برای پیش‌بینی لینک بر روی چندین مجموعه داده واقعی و مصنوعی، مورد آزمایش قرار گرفته است. در سناریوی دوم برای ارزیابی تأثیر اجماع در روش پیشنهادی از معیارهای مختلفی مثل معیار دقت، اطلاعات متقابل نرمال[1] و رند تنظیم‌شده[2] استفاده گردیده است. روش اجماع پیشنهادی با روش‌های رقیب در شرایط یکسان بر روی چندین مجموعه داده موجود در سایت یادگیر ماشین مقایسه شده و پارامترهای مختلف برای روش پیشنهادی در جدول ۲ آمده‌اند.

٤-١ سناریوی اول

در سناریوی اول، آزمایش‌هایی را بر روی دو مجموعه داده دنیای واقعی و یک مجموعه داده مصنوعی برای ارزیابی الگوریتم پیشنهادی خود انجام می‌دهیم. ما فرض می‌کنیم که مدل‌های پیشنهادی از همه گره‌ها آگاه هستند و هیچ گره جدیدی در مراحل زمانی بعدی معرفی نمی‌شود. در عوض، یال‌های بین گره‌های موجود با یک الگوی زمانی خاص تغییر می‌کند. خلاصه مجموعه داده‌های استفاده‌شده در جدول ۳ آمده است.

در سناریوی اول، روش پیشنهادی با روش‌های رقیب که در [۶۱] ارائه شده‌اند از نظر معیار میانگین دقت متوسط[3] و با اندازه تعبیه ۶۴ در شکل ۶ مقایسه شده‌اند. این معیار مطابق (۲۷) محاسبه گردیده است

1. Normalized Mutual Information
2. Adjusted Rand
3. Mean Average Precision

جدول ۲: پارامترهای روش پیشنهادی.

اندازه جمعیت در الگوریتم ژنتیک	۱۰۰
نرخ تقاطع	۰٫۸
نرخ جهش	۰٫۲
تعداد لایه‌های شبکه عمیق	۴
نرخ یادگیری شبکه عمیق	۱٫۰
بهینه‌ساز شبکه عمیق	آدام
تابع فعال‌ساز	سیگموئید

جدول ۳: مجموعه داده.

مجموعه داده	تعداد گره‌ها	تعداد یال‌ها	مراحل زمانی
SBM	۱۰۰۰	۵۶۰۱۶	۱۰
Hep-th	۱۴۴۴۶-۱۵۰	۴۸۲۷۴-۲۶۸	۱۳۶
AS	۷۷۱۶	۴۸۲۷۴-۲۶۸	۷۳۳

$$
\begin{aligned}
P@k &= \frac{\left|E_{pred_i}(k) \cap E_{gt_i}\right|}{k} \\
MAP &= \frac{\sum_i AP_i}{|V|} \\
AP(i) &= \frac{\sum_k precision@k(i).\{E_{pred_i}(k) \in E_{gt_i}\}}{\left|\{k : E_{pred_i}(k) \in E_{gt_i}\}\right|} \\
precision@k(i) &= \frac{E_{pred_i}(1:k) \cap E_{gt_i}}{k}
\end{aligned}
\qquad (۲۷)
$$

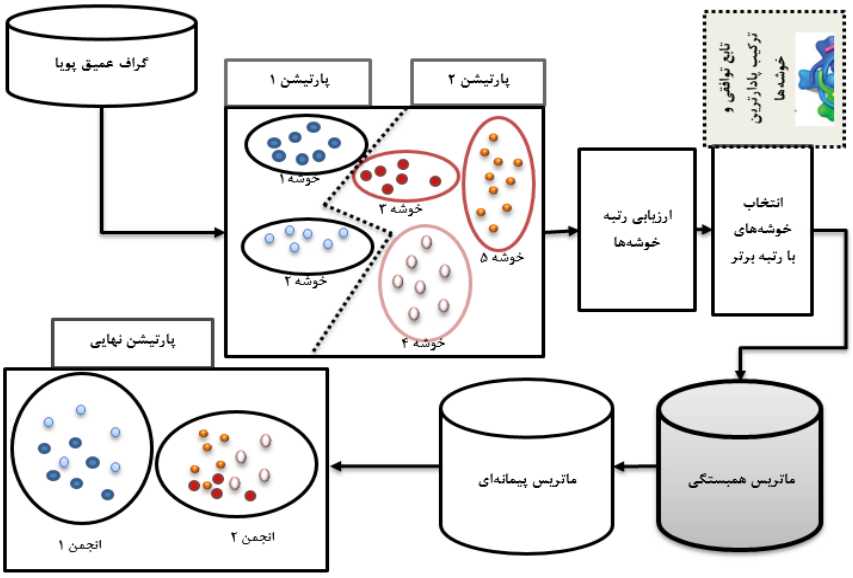
که $precision@k$ کسری از پیش‌بینی‌های درست در پیش‌بینی‌های بالای k است. در معادله $P@k$، E_{pred_i} و E_{gt_i} به‌ترتیب یال‌هایی هستند که در انجمن به درستی پیش‌بینی شده‌اند. MAP میانگین دقت را در تمام گره‌ها نشان می‌دهد و مقادیر $P@k$ برای آزمایش پیش‌بینی‌های برتر ساخته‌شده توسط مدل استفاده می‌شود. مقادیر MAP مستحکم[4] هستند و میانگین پیش‌بینی‌ها برای همه گره‌ها را شامل می‌شوند. مقادیر بالای MAP نشان می‌دهند که مدل می‌تواند پیش‌بینی‌های خوبی را برای اکثر گره‌ها انجام دهد.

در این بخش، ما نتیجه عملکرد مدل‌های مختلف را برای پیش‌بینی لینک در مجموعه داده‌های مختلف ارائه می‌کنیم. همچنین مدل را بر روی گراف‌ها از مرحله زمانی t تا $t+l$ که l نگاه به عقب مدل است آموزش می‌دهیم و لینک‌های گراف را در مرحله زمانی $t+l+۱$ پیش‌بینی می‌کنیم. l یک فراپارامتر در مدل است. برای یک نمودار در حال تکامل با مراحل T، پیش‌بینی بالا را از $T/۲$ تا T انجام می‌دهیم و میانگین MAP پیش‌بینی لینک را گزارش می‌کنیم.

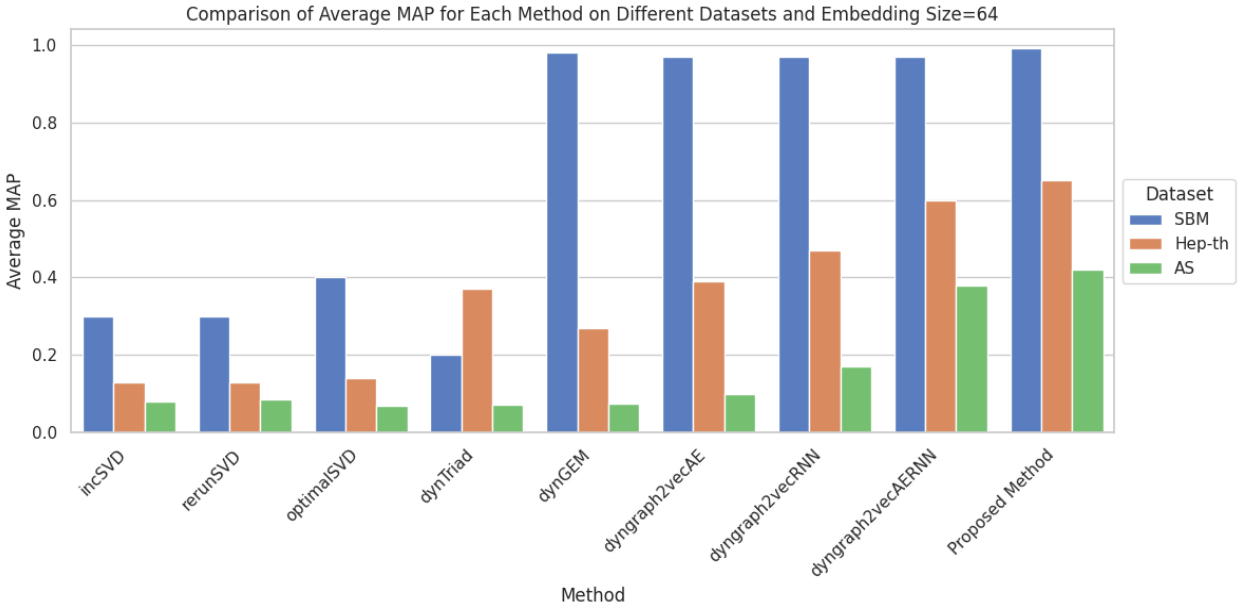
٤-٢ سناریوی دوم

در سناریوی دوم از آزمایش‌ها، روش خوشه‌بندی جمعی پیشنهادی با سایر روش‌های رقیب از نظر معیارهای NMI، AR و Accuracy مقایسه می‌شود. در این سناریو از داده‌های استاندارد سایت یادگیری ماشین استفاده شده است. شکل‌های ۷ تا ۹ به‌ترتیب روش پیشنهادی را از نظر مقدار NMI، AR و Accuracy با سایر روش‌های رقیب مقایسه نموده‌اند.

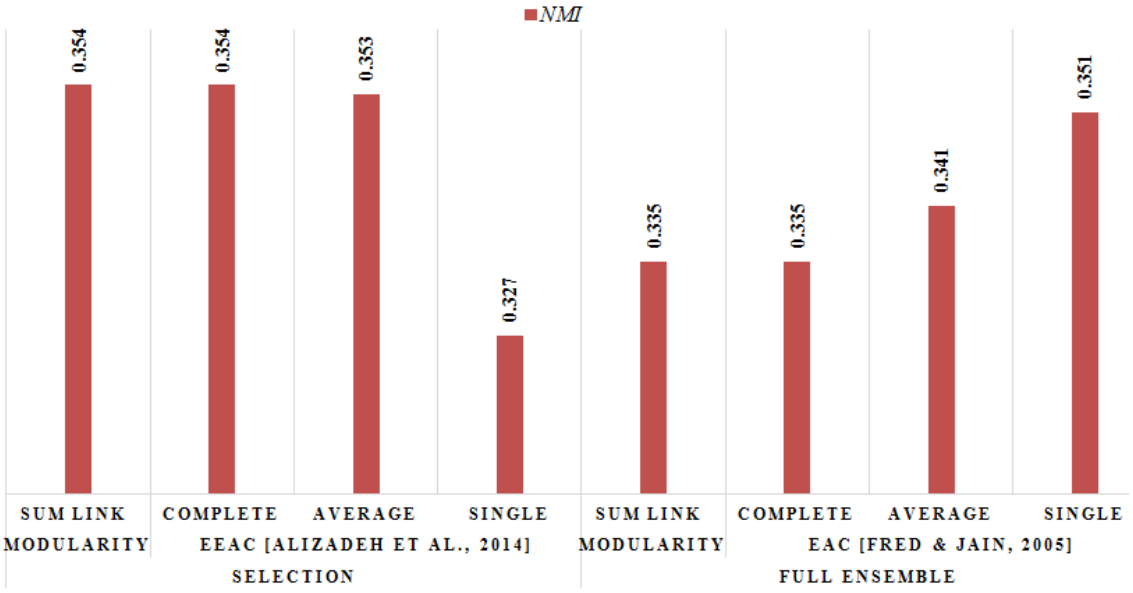
4. Robust



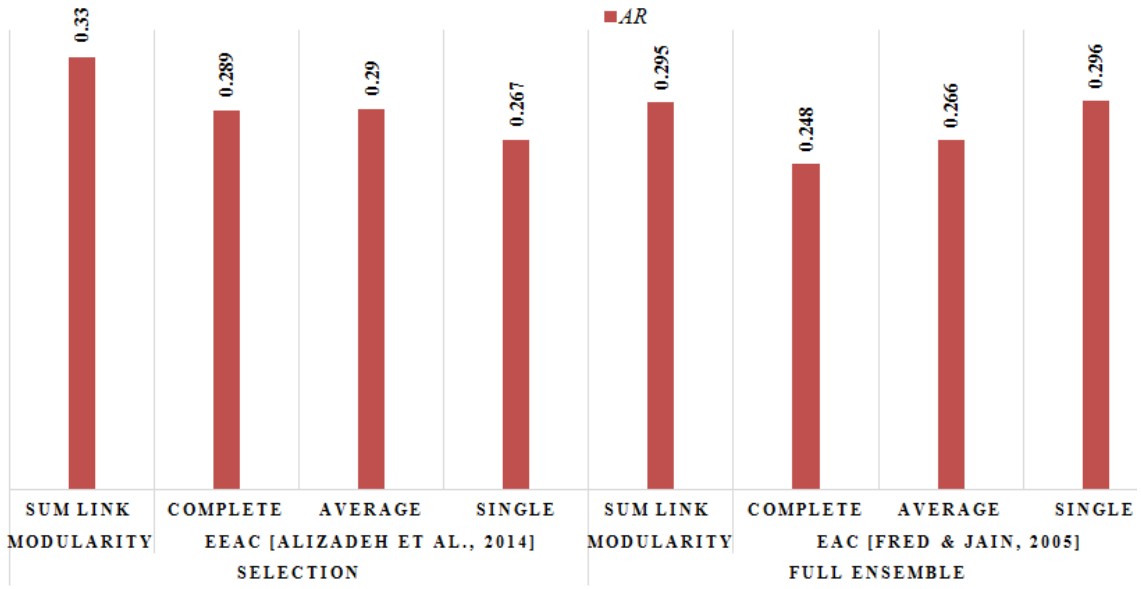
شکل ۵: شمای کلی روش پیشنهادی.



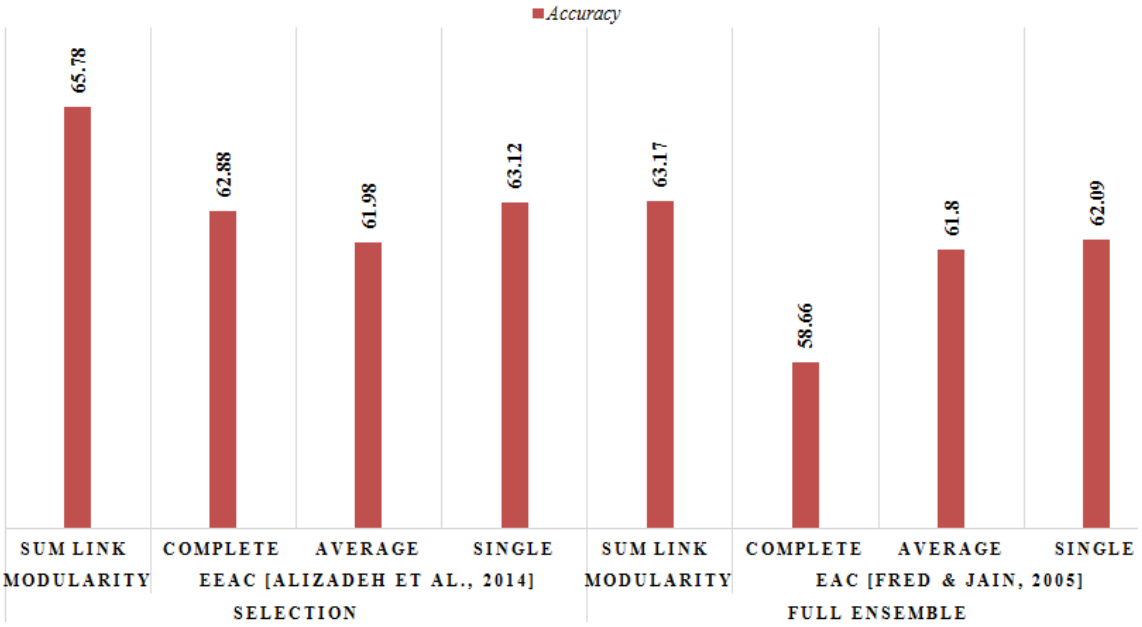
شکل ۶: مقایسه روش پیشنهادی با سایر روش‌ها از نظر معیار میانگین دقت متوسط.



شکل ۷: مقایسه روش پیشنهادی با سایر روش‌های رقیب از نظر معیار NMI.



شکل ۸: مقایسه روش پیشنهادی با سایر روش‌های رقیب از نظر معیار AR.

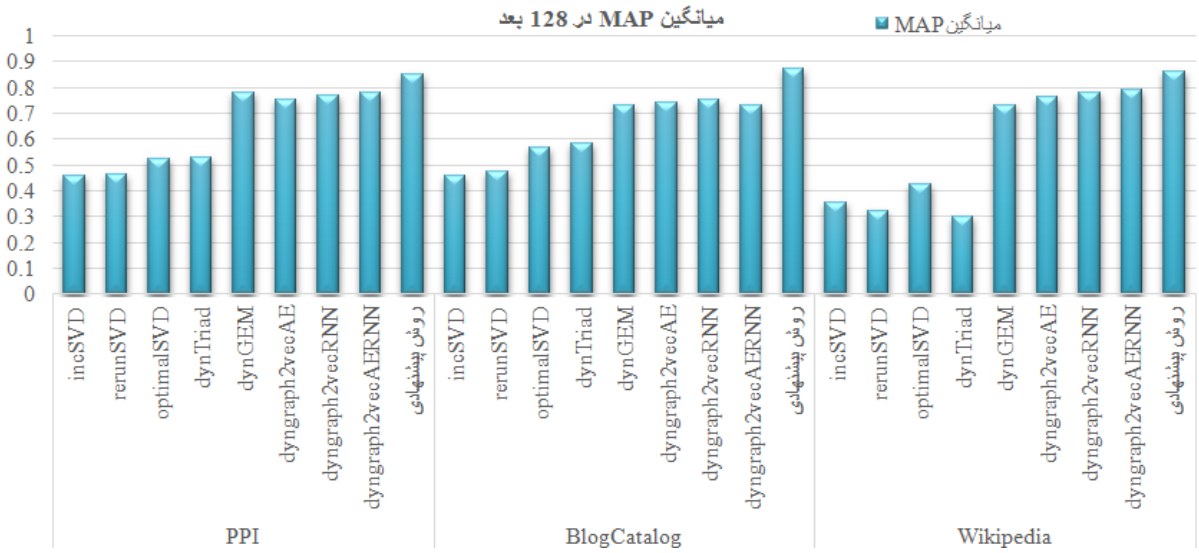


شکل ۹: مقایسه روش پیشنهادی با سایر روش‌های رقیب از نظر معیار Accuracy.

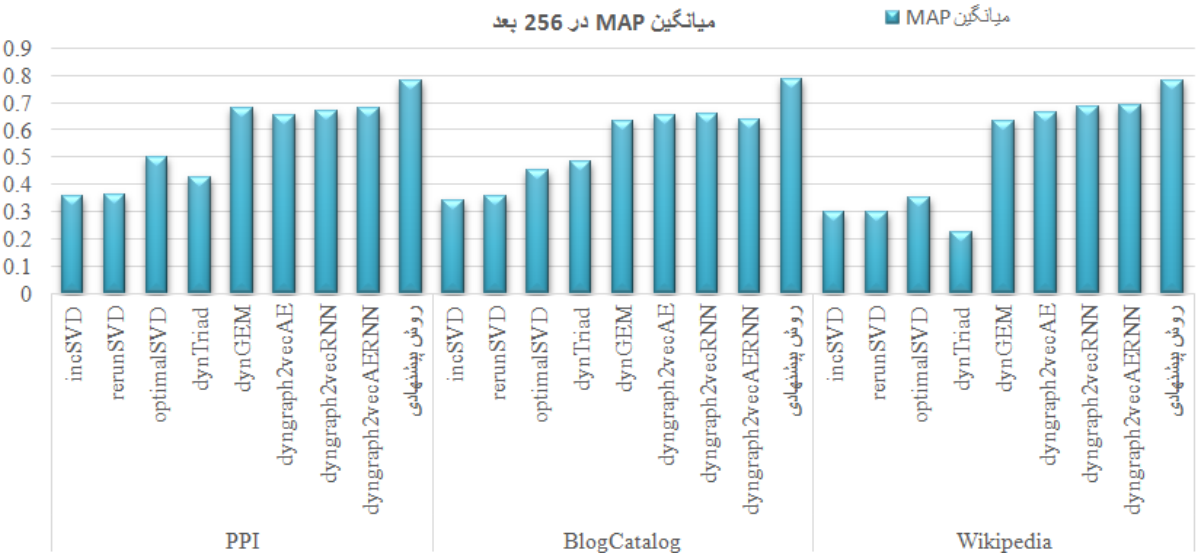
خلاصه نتایج به‌دست‌آمده از به‌کارگیری الگوریتم‌های مختلف خوشه‌بندی از نظر معیارهای NMI، AR و Accuracy در نمودارهای ترسیم‌شده در شکل‌های ۷ تا ۹ آمده‌اند. نتایج به دو بخش عمده تقسیم می‌شوند: مجمع کامل و انتخاب. مجمع کامل، رویکرد خوشه‌بندی جمعی مشترک را نشان می‌دهد که در آن از تمام نتایج اولیه برای یافتن پارتیشن توافقی استفاده می‌شود. انتخاب مخفف، رویکرد انتخاب خوشه‌بندی جمعی است که در آن تنها زیرمجموعه‌ای از نتایج اولیه بر اساس معیار AAPMM انتخاب می‌شوند. هر یک از بخش‌های اصلی مجدداً به دو روش مختلف تقسیم می‌شوند: اولی استفاده از استراتژی شناخته‌شده EAC برای جمع‌آوری نتایج در یک ماتریس همبستگی و استفاده از یک الگوریتم پیوند برای استخراج خوشه‌های نهایی است. دومین طرح جدید پیشنهادی است که اطلاعات انباشته‌شده در ماتریس همبستگی را به یک ماتریس پیمانه‌ای، تبدیل و الگوریتم پیوند جمع پیشنهادی را اعمال می‌کند. همه معیارهای NMI، AR و Accuracy در شکل‌های ۷ تا ۹ تأیید می‌کنند که الگوریتم پیشنهادی ما با به‌دست‌آوردن حداکثر عملکرد در اکثر موارد آزمایش، بهترین روش عملکردی است. این شکل صراحتاً نشان می‌دهد که چارچوب انتخاب پیشنهادی، بسیار بهتر از استراتژی اجماع کامل عمل می‌کند. همچنین تأیید می‌نماید که الگوریتم پیشنهادی به‌طور قابل توجهی از روش‌های قبلی در هر دو مجمع کامل و استراتژی انتخاب بهتر عمل می‌کند. به‌طور کلی، نتایج تجربی نشان می‌دهند که الگوریتم پیشنهادی به‌عنوان تابع اجماع مبتنی بر پیمانه‌ای انتخاب خوشه‌بندی جمعی قطعاً بهترین گزینه برای خوشه‌بندی داده‌های ورودی است.

٤-۳ سناریوی سوم

در این سناریو، روش پیشنهادی بر روی سه مجموعه داده دنیای واقعی و در ابعاد ۱۲۸ و ۲۵۶ با سایر روش‌های دیگر مقایسه شده است. همان طور که در نمودارهای ترسیم‌شده در شکل‌های ۱۰ و ۱۱ مشاهده می‌شود با افزایش تعداد ابعاد، مقدار میانگین دقت برای همه روش‌ها کاهش می‌یابد. با افزایش تعداد ابعاد، پیچیدگی فضای مسئله بیشتر شده و در



شکل ۱۰: مقایسه روش پیشنهادی با سایر روش‌ها از نظر میانگین دقت و تعداد ۱۲۸ بعد.



شکل ۱۱: مقایسه روش پیشنهادی با سایر روش‌ها از نظر میانگین دقت و تعداد ۲۵۶ بعد.

نتیجه دقت روش‌ها نیز کاهش می‌یابند. همان طور که مشاهده می‌شود روش پیشنهادی با افزایش تعداد ابعاد، نسبت به سایر روش‌ها عملکرد بسیار بهتری را از خود نشان داده است.

۵- نتیجه‌گیری و پیشنهادهای آتی

در این مقاله، ابتدا روشی برای تعبیه گراف آمده است و این تعبیه به‌عنوان ورودی به مدل خوشه‌بندی جمعی تحویل داده گردید. نشان داده شد که با کاهش ابعاد فضای مسئله می‌توان به نتایج با دقت بالاتری در تحلیل شبکه‌های پیچیده پویا رسید. در این مقاله، یک چارچوب انتخاب خوشه‌بندی جمعی پیشنهاد گردید که وظیفه بهینه‌سازی در تابع اجماع را دارد. چارچوب پیشنهادی به هر دو روند انتخاب و بهینه‌سازی کمک می‌نماید تا عملکرد مسئله خوشه‌بندی جمعی را افزایش دهد. ما همچنین مسئله انتخاب خوشه‌بندی جمعی را به یک مدل برنامه‌نویسی ریاضی تبدیل کردیم و یک الگوریتم سریع حریصانه را برای حل آن پیشنهاد نمودیم. نتایج تجربی با اجراهای منفرد مجمع کامل شناخته‌شده روش EAC و همچنین روش انتخاب EEAC مقایسه گردید. مطالعه تجربی نشان داد که چارچوب انتخاب خوشه‌بندی جمعی مبتنی بر بهینه‌سازی پیشنهادی به‌طور قابل توجهی بهتر از سایر روش‌های مجمع کامل یا انتخاب، عمل می‌کند. از آنجایی که چارچوب انتخاب خوشه‌بندی جمعی ما تنها بهترین خوشه‌ها را برای تشکیل مجموعه نهایی انتخاب می‌کند، انتظار می‌رود که استفاده از تکنیک ما در روش‌های خوشه‌بندی قوی‌تر از k-means بتواند حتی به پارتیشن‌های بهتر مجموعه داده‌های پیچیده منجر شود. ما مسئله خوشه‌بندی جمعی را به یک برنامه ریاضی تغییر دادیم و یک مدل درجه دوم ۰-۱ را برای به حداکثر رساندن پیمانه‌ای به‌عنوان تابع اجماع پیشنهاد کردیم. اگرچه که الگوریتم ارائه‌شده در بهینه‌سازی پیمانه‌ای‌بودن به خوبی عمل می‌کند، اما خروجی را تضمین نمی‌کند که راه‌حل بهینه باشد. علاوه بر این، یافتن نتیجه بهینه برای هر مدل IP یک مسئله NP-complete است. بنابراین در آینده برای حل این مسئله، برخی از پیشرفت‌ها را در مدل‌سازی کار بهینه‌سازی ابداع خواهیم کرد. به بیان دقیق‌تر، پیشنهاد مدل‌سازی ریاضی دیگری که می‌تواند در یک زمان معقول به یک راه حل دقیق برای مجموعه‌های داده در مقیاس بزرگ ارائه شود، مسئله باز ما است که قصد داریم در کار آینده خود آن را بررسی کنیم. پیشنهاد دیگر برای آینده این است که مدل و الگوریتم حل ما را برای تابع اجماع مورد بازنگری قرار دهیم تا به‌طور خودکار تعداد مناسب خوشه‌هایی را که مجموعه داده‌های آن ناشناخته است، تعیین کنیم.

مراجع

[1] Z. Gao, *et al.*, "Hierarchical graph learning for protein–protein interaction," *Nature Communications*, vol. 14, Article ID: 1093, 2023.

[2] V. Ranjbar, M. Salehi, P. Jandaghi, and M. Jalili, "Qanet: tensor decomposition approach for query-based anomaly detection in heterogeneous information networks," *IEEE Trans. on Knowledge and Data Engineering*, vol. 31, no. 11, pp. 2178-2189, Nov. 2018.

[3] H. Dai, Y. Wang, R. Trivedi, and L. Song, *Deep Coevolutionary Network: Embedding User and Item Features for Recommendation*, arXiv:1609.03675v4, 2017

[4] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, "A comprehensive survey on graph neural networks," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 32, no. 1, pp. 4-24, Jan. 2021.

[5] J. Zhou, *et al.*, "Graph neural networks: a review of methods and applications," *AI Open*, vol. 1, pp. 57-81, 2020.

[6] P. Xu, W. Hu, J. Wu, and B. Du, "Link prediction with signed latent factors in signed social networks," in *Proc. 25th ACM SIGKDD Int. Conf. on Knowledge Discovery & Data Mining*, pp. 1046-1054, Anchorage, AK, USA, 4-8 Aug. 2019.

[7] M. Khorasani, B. Minaei-Bidgoli, and C. Saedi, "Automatic synset extraction from text documents using a graph-based clustering approach via maximal cliques finding," *International J. of Information and Communication Technology Research*, vol. 11, no. 1, pp. 27-35, Winter 2019.

[8] A. Grover and J. Leskovec, "node2vec: scalable feature learning for networks," in *Proc. of the 22nd ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, pp. 855-864, San Francisco, CA, USA, 13-17 Aug. 2016.

[9] J. Skarding, B. Gabrys, and K. Musial, "Foundations and modelling of dynamic networks using dynamic graph neural networks: a survey," *IEEE Access*, vol. 9, pp. 79143-79168, 2021.

[10] M. Torricelli, M. Karsai, and L. Gauvin, "weg2vec: event embedding for temporal networks," *Scientific Reports*, vol. 10, Article ID: 7164, 11 pp., 2020.

[11] S. Cao, W. Lu, and Q. Xu, "Deep neural networks for learning graph representations," in *Proc. 13th AAAI Conf. on Artificial Intelligence*, pp. 1145-1152, Phoenix, AZ, USA, 12-17 Feb. 2016.

[12] M. T. FaghihiNezhad and B. Minaei Bidgoli, "Development of an ensemble learning-based intelligent model for stock market forecasting," *Scientia Iranica*, vol. 28, no. 1, pp. 395-411, Jan./Feb. 2021.

[13] A. Banerjee, *et al.*, "A new method for weighted ensemble clustering and coupled ensemble selection," *Connection Science*, vol. 33, no. 3, pp. 623-644, 2021.

[14] D. Huang, C. D. Wang, H. Peng, J. Lai, and C. K. Kwoh, "Enhanced ensemble clustering via fast propagation of cluster-wise similarities," *IEEE Trans. on Systems, Man, and Cybernetics: Systems*, vol. 51, no. 1, pp. 508-520, Jan. 2021.

[15] N. Ilc, "Weighted cluster ensemble based on partition relevance analysis with reduction step," *IEEE Access*, vol. 8, pp. 113720-113736, 2020.

[16] P. Goyal, A. Sapienza, and E. Ferrara, "Recommending teammates with deep neural networks," in *Proc. 29th on Hypertext and Social Media,* pp. 57-61, Baltimore, MD, USA, 9-12 Jul. 2018.

[17] J. B. Lee, R. Rossi, and X. Kong, "Graph classification using structural attention," in *Proc. 24th ACM SIGKDD Int. Conf. on Knowledge Discovery & Data Mining*, pp. 1666-1674, London, UK, 19-23 Aug. 2018.

[18] J. You, B. Liu, R. Ying, V. Pande, and J. Leskovec, "Graph convolutional policy network for goal-directed molecular graph generation," in *Proc. 32nd Int. Conf. on Neural Information Processing Systems*, pp. 6412-6422, Montréal, Canada, 3-8 Dec. 2018.

[19] R. Ying, J. You, C. Morris, X. Ren, W. L. Hamilton, and J. Leskovec, "Hierarchical graph representation learning with differentiable pooling," in *Proc. 32nd Int. Conf. on Neural Information Processing Systems*, 11 pp., Montréal, Canada, 3-8 Dec. 2018.

[20] Y. Ma, Z. Guo, Z. Ren, E. Zhao, J. Tang, and D. Yin, *Streaming Graph Nneural Networks*, arXiv preprint arXiv:1810.10627, 2018.

[21] F. Manessi, A. Rozza, and M. Manzo, *Dynamic Graph Convolutional Networks*, arXiv preprint arXiv:1704.06199, 2017.

[22] F. Monti, M. Bronstein, and X. Bresson, "Geometric matrix completion with recurrent multi-graph neural networks," in *Proc. 31st Conf. on Neural Information Processing Systems*, 11 pp., Long Beach, CA, USA, 4-9 Dec. 2017.

[23] J. You, R. Ying, X. Ren, W. Hamilton, and J. Leskovec, "GraphRNN: generating realistic graphs with deep auto-regressive models," in *Proc. 35 th Int. Conf. on Machine Learning*, 12 pp., Stockholm, Sweden, 10-15 Jul. 2018.

[24] Z. Zhang, *A Note on Spectral Clustering and SVD of Graph Data*, arXiv preprint arXiv:1809.11029, 2018.

[25] R. Ying, *et al*., "Graph convolutional neural networks for web-scale recommender systems," in *Proc. 24th ACM SIGKDD Int. Conf. on Knowledge Discovery & Data Mining*, pp. 974-983, London, UK, 19-23 Aug. 2018.

[26] D. Wang, P. Cui, and W. Zhu, "Structural deep network embedding," in *Proc. 22th ACM SIGKDD Int. Conf. on Knowledge Discovery & Data Mining*, pp. 1225-1234, San Francisco, CA, USA, 13-17 Aug. 2016.

[27] F. Manessi, A. Rozza, and M. Manzo, Dynamic *Graph Convolutional Networks*, arXiv preprint arXiv:1704.06199, 2017.

[28] Y. Ma, Z. Guo, Z. Ren, E. Zhao, J. Tang, and D. Yin, *Streaming Graph Neural Networks*, arXiv preprint arXiv:1810.10627, 2018.

[29] L. Zhu, D. Guo, J. Yin, G. Ver Steeg, and A. Galstyan, "Scalable temporal latent space inference for link prediction in dynamic social networks," *IEEE Trans. on Knowledge and Data Engineering*, vol. 28, no. 10, pp. 2765-2777, Oct. 2016.

[30] C. Hongyun, V. W. Zheng, and K. Chen-Chuan Chang, "A comprehensive survey of graph embedding: problems, techniques, and applications," *IEEE Trans. on Knowledge and Data Engineering*, vol. 3, no. 9, pp. 1616-1637, Sept. 2018.

[31] M. Ou, *et al.*, "Asymmetric transitivity preserving graph embedding," in *Proc. 22nd ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, pp. 1105-1114, San Francisco, CA, USA, 13-17 Aug. 2016.

[32] W. L. Hamilton, J. Leskovec, and D. Jurafsky, *Diachronic Word Embeddings Reveal Statistical Laws of Semantic Change*, arXiv preprint arXiv:1605.09096, 2016.

[33] S. Cao, W. Lu, and Q. Xu, "GraRep: learning graph representations with global structural information," in *Proc. 21st Intl. Conf. on Knowledge Discovery and Data Mining*, pp. 891-900, Melbourne Australia, 18-23 Oct. 2015.

[34] D. Wang, P. Cui, and W. Zhu, "Structural deep network embedding," in *Proc. 22th ACM SIGKDD Int. Conf. on Knowledge Discovery & Data Mining*, pp. 1225-1234, San Francisco, CA, USA, 13-17 Aug. 2016.

[35] A. L. Barabasi, *Network Science*, Cambridge University Press, 2016.

[36] K. Tu, P. Cui, X. Wang, P. S. Yu, and W. Zhu, "Deep recursive network embedding with regular equivalence," in *Proc. 24th ACM SIGKDD Int. Conf. on Knowledge Discovery & Data Mining*, pp. 2357-2366, London, UK, 19-23 Aug. 2018.

[37] A. Fred and A. K. Jain, "Combining multiple clusterings using evidence accumulation," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 27, no. 6, pp. 835-850, Jun. 2005.

[38] A. Strehl and J. Ghosh, "Cluster ensembles-a knowledge reuse framework for combining multiple partitions," *J. of Machine Learning Research*, vol. 3, pp. 583-617, 2002.

[39] H. Alizadeh, B. Minaei-Bidgoli, and H. Parvin, "Cluster ensemble selection based on a new cluster stability measure," *Intelligent Data Analysis*, vol. 18, no. 3, pp. 389-408, 2014.

[40] R. Caruana, *et al.*, "Ensemble selection from libraries of models," in *Proc. of the 21st Int. Conf. on Machine Learning*, 9 pp., Banff, Canada, 4-8 Jul. 2004.

[41] X. Fern and W. Lin, "Cluster ensemble selection," *Statistical Analysis and Data Mining*, vol. 1, no. 3, pp. 128-141, Nov. 2008.

[42] J. Azimi and X. Fern, "Adaptive cluster ensemble selection," in *Proc. of 21th Int. Joint Conf. on Artificial Intelligence,* pp. 992-997, Pasadena, Ca, USA, 11-17 Jul. 2009.

[43] H. Alizadeh, P. H. Parvin, and S. Parvin, "A framework for cluster ensemble based on a max metric as cluster evaluator," *IAENG International J. of Computer Science*, vol. 39, no. 1, 10 pp., Feb. 2012.

[44] H. Alizadeh, B. Minaei-Bidgoli, and H. Parvin, "To improve the quality of cluster ensembles by selecting a subset of base clusters," *J. of Experimental & Theoretical Artificial Intelligence*, vol. 26, no. 1, Article ID: 813974, 2014.

[45] W., Li, Z. Wang, W. Sun, S. Bahrami, "An ensemble clustering framework based on hierarchical clustering ensemble selection and clusters clustering," *Cybernetics and Systems,* vol. 54, no. 5, pp. 741-766, 2023.

[46] H. Khalili, M. Rabbani, E. Akbari, "Clustering ensemble selection: a systematic mapping study," *International Journal of Nonlinear Analysis and Applications*, vol. 14, no. 9, pp. 209-240, 2023.

[47] S. T. Hadjitodorov, L. I. Kuncheva, and L. P. Todorova, "Moderate diversity for better cluster ensembles," *Information Fusion*, vol. 7, no. 3, pp. 264-275, Sept. 2006.

[48] H. Parvin, B. Minaei-Bidgoli, and H. Alizadeh, "A new clustering algorithm with the convergence proof," in *Proc. of the 15th Int. Conf, on Knowledge-Based and Intelligent Information and Engineering Systems,* vol. 1, pp. 21-31, Kaiserslautern, Germany, 12-14 Sept. 2011.
[49] V. Singh, L. Mukherjee, J. Peng, and J. Xu, "Ensemble clustering using semidefinite programming with applications," *Mach. Learn.*, vol. 79, pp. 177-200, Dec. 2010.
[50] P. R. Rao and J. P. P. da Costa, "A performance study of a consensus clustering algorithm and properties of partition graph," in *Proc. of IEEE Int. Conf. on Computational Intelligence and Computing Research*, 5 pp., Coimbatore, India, 28-29 Dec. 2011.
[51] A. Gunoche, "Consensus of partitions: a constructive approach," *Advances in Data Analysis and Classification*, vol. 5, no. 3, pp. 215-229, 2011.
[52] I. T. Christou, "Coordination of cluster ensembles via exact methods," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 33, no. 2, pp. 279-293, Feb. 2011.
[53] S. Vega-Ponz and J. Ruiz-Shulcloper, "A survey of clustering ensemble algorithms," *Int. J. of Pattern Recognition and Artificial Intelligence*, vol. 25, no. 3, pp. 337-372, 2011.
[54] U. Brandes, *et al.*, "On modularity clustering," *IEEE Trans. on Knowledge and Data Engineering*, vol. 20, no. 2, pp. 172-188, Feb. 2008.
[55] G. Agrawal and D. Kempe, "Modularity-maximizing network communities via mathematical programming," *Eur. Phys. J. B*, vol. 66, pp. 409-418, 2008.
[56] X. S. Zhang, *et al.*, "Modularity optimization in community detection of complex networks," *Europhys Lett*, vol. 87, no. 3, Article ID: 38002, 2009.
[57] X. S. Zhang, *et al.*, "A combinatorial model and algorithm for globally searching community structure in complex networks," *J. Comb Optim*, vol. 23, pp. 425-442, 2012.
[58] C. Zhao, *et al.*, "Social network optimization for cluster ensemble selection," *Fundamenta Informaticae*, vol. 176, no. 1, pp. 79-102, 2020.
[59] R. Hosseinzadeh, H. Alizadeh, and E. Nazemi, "Community detection ensemble in social networks," in *Proc. the 11th Iranian Conf. on Intelligent Systems,* pp. 27-28, Tehran, Iran, Feb. 2013.
[60] D. Huang, C. D. Wang, and J. H. Lai, "Locally weighted ensemble clustering," *IEEE Trans. on Cybernetics*, vol. 48, no. 5, pp. 1460-1473, May 2018.
[61] P. Goyal, S. R. Chhetri, and A. Canedo, "dyngraph2vec: capturing network dynamics using dynamic graph representation learning," *Knowledge-Based Systems*, vol. 187, Article ID: 104816, Jan. 2020.
[62] P. Goyal, S. R. Chhetri, and A. Canedo, *dyngraph2vec: Capturing Network Dynamics using Dynamic Graph Representation Learning*, arXiv preprint arXiv:1809.02657, 2018.
[63] A. Clauset, M. E. J. Newman, and C. Moore, "Finding community structure in very large networks," *Phys. Rev. E*, vol. 70, Article ID: 066111 2004.
[64] M. E. J. Newman, "Modularity and community structure in networks," in *Proc. Natl Acad. Sci. USA*, vol. 103, no. 23, pp. 8577-8582, 6 Jun. 2006.
[65] L. Shuzhuo, C. Yinghui, H. Haifeng, and M. W. Feldman, "A genetic algorithm with local search strategy for improved detection of community structure," *Complexity*, vol. 15, no. 4, pp. 53-60, Mar./Apr. 2010.
[66] Y. Ren, C. Domeniconi, G. Zhang, and G. Yu, "Weighted-object ensemble clustering: methods and analysis," *Knowledge and Information Systems*, vol. 51, no. 2, pp. 661-689, 2017.

مجید محمدپور تحصیلات خود را در مقاطع کارشناسی و کارشناسی ارشد مهندسی کامپیوتر به‌ترتیب در سال‌های ۱۳۸۶ و ۱۳۹۳ در دانشگاه بعثت کرمان و دانشگاه علوم تحقیقات تهران به پایان رسانده است و در حال حاضر دانشجوی دکتری مهندسی کامپیوتر در دانشگاه یزد می‌باشد. نام‌برده هم‌اکنون در چندین واحد دانشگاهی در رشته کامپیوتر مشغول به تدریس می‌باشد. از جمله زمینه‌های تحقیقاتی مورد علاقه ایشان عبارتند از: بهینه‌سازی تکاملی پویا، یادگیری عمیق، یادگیری فدرال، و اینترنت اشیا.

سید اکبر مصطفوی تحصیلات خود را در مقطع کارشناسی رشته مهندسی فناوری اطلاعات در سال ۱۳۸۷ در دانشگاه صنعتی شریف و مقاطع کارشناسی ارشد و دکتری شبکه‌های کامپیوتری را به‌ترتیب در سال‌های ۱۳۸۹ و ۱۳۹۴ در دانشگاه صنعتی امیر کبیر به پایان رسانده است و هم‌اکنون دانشیار دانشکده مهندسی کامپیوتر دانشگاه یزد می‌باشد. زمینه‌های تحقیقاتی مورد علاقه ایشان عبارتند از: اینترنت اشیا، رایانش مه و ابری، امنیت شبکه، شبکه‌های بی‌سیم و سیار.

وحید رنجبر بافقی تحصیلات کارشناسی خود را در دانشگاه صنعتی شیراز در رشته مهندسی فناوری اطلاعات در سال ۱۳۹۰ گذراند و مقطع کارشناسی ارشد فناوری اطلاعات خود را در دانشگاه صنعتی شریف در سال ۱۳۹۲ به پایان رساند، سپس در سال ۱۳۹۷ دکتری فناوری اطلاعات خود را از دانشگاه تهران دریافت کرد. وی هم‌اکنون استادیار دانشکده مهندسی کامپیوتر دانشگاه یزد است و زمینه های تحقیقاتی مورد علاقه ایشان امنیت اطلاعات و شبکه، تحلیل شبکه های اطلاعاتی و اینترنت اشیا است.