

پردازش مؤثر گراف‌های ناهمگن بر اساس زیرگراف‌های غنی شده

وحید معتقد

پژوهشکده علوم کامپیوتر

پژوهشگاه دانش‌های بنیادی، تهران

(رایانامه: vahid.motaghd2020@gmail.com)

علیرضا نیک‌پوش

دانشکده ریاضی و علوم کامپیوتر

دانشگاه علم و صنعت ایران، تهران

(رایانامه: alireza.nikpoosh.cs@gmail.com)

فاطمه بهاری‌فرد

پژوهشکده علوم کامپیوتر

پژوهشگاه دانش‌های بنیادی، تهران

(رایانامه: f.baharifard@ipm.ir)

چکیده

امروزه گراف‌ها جایگاه ویژه‌ای در مدل‌سازی مسائل دنیای واقعی داشته و برحسب تنوع رئوس و یال‌های آن‌ها، به دو دسته همگن و ناهمگن تقسیم می‌شوند. از روش‌های بروز و کارا برای پردازش گراف، الگوریتم‌های یادگیری ماشین می‌باشد. در روش یادگیری نظارت‌شده حجم زیادی داده برچسب‌دار آموزشی نیاز است که تهیه آن‌ها در برخی مواقع هزینه‌بر یا غیرعملی خواهد بود. یادگیری خودناظر و یادگیری با مجموعه داده محدود روش‌هایی برای مقابله با این مشکل هستند. در رویکرد پیش‌آموزش-تنظیم پرامپت، مدل پیش‌آموزش به صورت عام‌منظوره و بردار پرامپت برای مسائل پایین‌دست آموزش داده می‌شوند. در این روش نیاز به ساخت شبکه عمیق به ازای هر مسأله پایین‌دست و بروزرسانی وزن‌های مدل پیش‌آموزش وجود ندارد. در این مقاله چارچوبی برای پردازش گراف‌های ناهمگن با رویکرد پیش‌آموزش-تنظیم پرامپت ارائه شده است. در این الگوریتم، ابتدا گراف ناهمگن ورودی به زیرگراف‌هایی همگن تجزیه شده و سپس با افزودن یال‌های غنی‌سازی، ارتباطات غیر مستقیم موجود در گراف ناهمگن ورودی به زیرگراف‌ها تزریق می‌شود. پیش‌آموزش بدون استفاده از داده‌های برچسب‌دار، به صورت خودناظر و بر اساس شبکه عصبی گرافی انجام می‌شود. برای مسائل پایین‌دست، بردار پرامپت ویژگی و لایه کانولوشنی تجمع به صورت هم‌زمان روی تعداد محدودی داده آموزشی، آموزش می‌بیند. الگوریتم پیشنهادی روی سه مجموعه داده ناهمگن ACM، DBLP و Freebase اعمال و نتایج به‌دست‌آمده با نتایج حاصل از چند الگوریتم پردازش گراف مقایسه شده است که نشانگر دقت مناسب روش ارائه‌شده می‌باشد.

کلیدواژه‌ها

یادگیری ماشین، شبکه عصبی گراف، گراف ناهمگن، جاسازی رئوس، یادگیری خودناظر، یادگیری پرامپت

۱. مقدمه

گراف‌ها ساختمان داده‌های قدرتمند برای مدل‌سازی روابط پیچیده بین موجودیت‌ها هستند. در گراف‌های ناهمگن^۱، موجودیت‌ها

(رئوس) و روابط بین آن‌ها (یال‌ها) می‌تواند انواع مختلف داشته باشد. گراف‌های ناهمگن برای نمایش سیستم‌های دنیای امروزه مانند شبکه‌های اجتماعی، گراف‌های دانش و شبکه‌های بیولوژیکی که در آن‌ها با چندین نوع رأس و چندین نوع یال مواجه هستیم، مؤثر بوده و به طور گسترده مورد استفاده قرار می‌گیرد [۱-۳].

یکی از کاراترین الگوریتم‌هایی که می‌تواند برای آموزش نمایش^۲ داده‌های گراف مورد استفاده قرار گیرد، شبکه‌های عصبی گرافی^۳ هستند. شبکه‌های عصبی گراف می‌تواند برای طیف وسیعی از مسائل مانند پیش‌بینی ایجاد یال^۴، دسته‌بندی رئوس^۵ و دسته‌بندی گراف‌ها^۶ مورد استفاده قرار گیرند [۴].

اساساً شبکه‌های عصبی گراف برای گراف‌های همگن طراحی شده‌اند و روی گراف‌های ناهمگن نمی‌تواند اطلاعات محتوایی و ساختاری را به خوبی استخراج کنند. این محدودیت‌ها سبب شد تا شبکه‌های عصبی گرافی ناهمگن^۷ طراحی شوند تا توانایی مدیریت پیچیدگی‌های داده‌های ناهمگن، فراهم شود. در واقع شبکه‌های عصبی گراف ناهمگن نوعی شبکه عصبی گراف هستند که برای مواجه شدن با گراف‌های ناهمگن که شامل بیش از یک نوع رأس و/یا یک نوع یال هستند ارائه شده‌اند [۵].

یکی از چالش‌های استفاده از شبکه‌های عصبی گراف ناهمگن، در اختیار داشتن حجم مناسبی از داده برچسب‌دار برای آموزش شبکه عمیق است. در بسیاری از کاربردهای دنیای واقعی، در اختیار داشتن چنین داده‌ای آموزشی می‌تواند گران‌قیمت و زمان‌بر باشد و در مواقعی شاید اصلاً نتوان به آن دست یافت. برای مواجه شدن با این محدودیت،

^۲ representation learning

^۳ graph neural network

^۴ link prediction

^۵ node classification

^۶ graph classification

^۷ heterogeneous graph neural network

^۱ heterogeneous graph

برای تنظیم تأثیر هر زیرگراف استفاده می‌شود. فرایند آموزش در این الگوریتم بر مبنای یادگیری با مجموعه داده محدود است و با توجه به این‌که در بخش قالب‌بندی گراف، حجم زیادی از دانش ساختاری گراف ناهمگن نادیده گرفته می‌شود، ممکن است یادگیری با دقت مناسبی انجام نشود.

هدف این مقاله گسترش و بهبود الگوریتم Hgprompt برای استخراج مناسب‌تر اطلاعات و افزایش دقت است. نوآوری الگوریتم پیشنهادی را می‌توان به دو بخش تقسیم کرد. بخش اول بهبود فرایند تجزیه گراف ناهمگن به چند زیرگراف و بخش دوم مربوط به تجمیع اطلاعات خروجی از زیرگراف‌ها است. برای بهبود فرایند تجزیه، ارتباطات غیرمستقیم موجود در گراف ناهمگن ورودی به زیرگراف‌ها اضافه شده و وجود لایه تجمیع باعث استخراج بهتر اطلاعات جاسازی‌های^{۱۶} رئوس زیرگراف‌ها می‌شود.

۲. پژوهش‌های پیشین

در گذشته الگوریتم‌هایی مانند DeepWalk [۱۱] برای پردازش گراف استفاده می‌شد. در این الگوریتم پس از قدم‌زنی تصادفی روی گراف، مجموعه‌ای از دنباله‌های کوتاه از رئوس تولید شده و سپس با استفاده از روش Word2Vec جاسازی رئوس در خروجی ارائه می‌شود. الگوریتم Node2Vec [۱۲] هم فرایند مشابهی دارد، با این تفاوت که در آن پارامترهای کنترلی معرفی شد تا امکان جستجوی تعمیم‌یافته‌تر فراهم آید. در این الگوریتم‌ها گراف ورودی باید همگن باشد و رویکرد مؤثری برای درک و تحلیل گراف‌های ناهمگن در آن‌ها وجود ندارد.

فرامسیرها^{۱۷} یک مفهوم تعریف شده در اکثر روش‌های جاسازی گراف ناهمگن هستند. در روش‌هایی مانند metapath2vec [۱۳] و HIN2Vec [۱۴] از این مفهوم به عنوان دانش اولیه استفاده شده است. فرامسیر الگویی از توالی چند نوع رأس است که رابطه ترکیبی بین انواع مختلف رئوس در یک گراف ناهمگن را نشان می‌دهد. این الگوریتم‌ها نیز جاسازی رئوس را محاسبه می‌کنند و برای اضافه کردن قابلیت فعالیت روی گراف‌های ناهمگن، باید همه دنباله‌های رئوس بر اساس فرامسیرها انتخاب شده باشند.

الگوریتم GCN^{۱۸} [۱۵] را می‌توان نقطه آغاز پژوهش روی شبکه‌های عصبی گرافی دانست. این الگوریتم با استفاده از یک تقریب مرتبه اول محلی از کانولوشن‌های طیفی، به طور مستقیم روی ساختار گراف عمل کرده و به یادگیری جاسازی رئوس می‌پردازد. GCN در هر لایه ویژگی‌های هر رأس را با میانگین‌گیری از ویژگی‌های همسایگانش ترکیب کرده و این فرایند به صورت تکراری در لایه‌های متعدد ادامه می‌یابد. امروزه الگوریتم‌های مبتنی بر شبکه‌های عصبی گراف فرصت استفاده روی مسائل پایین‌دست مختلف را فراهم آورده‌اند. به عنوان مثال، در MECCH [۱۶] یک معماری جدید برای پردازش گراف‌های ناهمگن مبنی بر فرامسیر ارائه شده که فرایند یادگیری آن بر پایه شبکه‌های عصبی گراف است. در این الگوریتم پس از پیش‌پردازش ویژگی‌ها، حوزه‌های فرامسیر مختلفی برای هر رأس ساخته می‌شود تا

می‌توان از روش یادگیری با مجموعه داده محدود^۸ بهره برد که در آن از داده‌های آموزشی برچسب‌دار محدود استفاده می‌شود [۶]. در یادگیری با مجموعه داده محدود، تمرکز بر استخراج دانش (شبه-دانش) از مسائلی است که برای آن‌ها داده برچسب‌دار وجود ندارد.

یکی از رویکردهای مورد استفاده برای مسائل یادگیری که اخیراً مورد استفاده پژوهشگران قرار گرفته است، روش پیش‌آموزش-تنظیم دقیق^۹ می‌باشد [۷]. در این روش یک شبکه در فرایند پیش‌آموزش، آموزش داده شده و سپس این شبکه برای مسائل پایین‌دست^{۱۰}، تنظیم دقیق می‌شود. این فرایند برای طیف بسیاری از مسائل به خوبی انجام می‌شود ولی اگر بین مسأله‌ی پیش‌آموزش و مسائل پایین‌دست، اختلاف و شکاف زیادی وجود داشته باشد، نتایج خوبی به دنبال نخواهد داشت. هم‌چنین اگر مدل پیش‌آموزش بزرگ بوده و یا دسترسی به این مدل میسر نباشد، عملاً فرایند تنظیم دقیق مدل پیش‌آموزش نمی‌تواند محقق شود. یادگیری به روش پیش‌آموزش-تنظیم دقیق با مجموعه داده محدود در گراف‌های ناهمگن به دلیل وجود مفاهیمی مانند شکاف ناهمگنی^{۱۱} و تغییرات توزیع^{۱۲} موجب اضافه شدن مشکلات جدیدی می‌شود. شکاف ناهمگنی به اختلاف بین انواع رأس و یال در گراف ناهمگن موجود در فرایند پیش‌آموزش و مسائل پایین‌دست و تغییرات توزیع به اختلاف بین ساختار و ویژگی‌های گراف‌های پیش‌آموزش و مسائل پایین‌دست اشاره دارد. این اختلاف‌ها می‌تواند تأثیر فرایند انتقال دانش را کم‌تر کند [۸].

در روش یادگیری پرامپت^{۱۳} برای هم‌سویی بیشتر بین مسأله‌ی پیش‌آموزش و مسائل پایین‌دست، سعی می‌شود که فرموله‌بندی جدیدی از مسائل پایین‌دست در نظر گرفته شود [۹]. در این رویکرد، مدل پیش‌آموزش ثابت فرض شده و وزن‌های آن بروزرسانی نشده و به ازای هر مسأله‌ی پایین‌دست تنها یک بردار کوچک تنظیم خواهد شد. در الگوریتم Hgprompt [۱۰] یادگیری پرامپت روی گراف‌های ناهمگن ارائه شده است. در این الگوریتم تلاش شده است تا شکاف بین گراف‌های همگن و ناهمگن در فرایند پیش‌آموزش و مسائل پایین‌دست، حل شود. هدف از به کارگیری فرایند قالب-دوگانه^{۱۴} در این الگوریتم، طراحی مشابه مسائل پایین‌دست و گراف ناهمگن مرحله پیش‌آموزش است. قالب-دوگانه شامل قالب‌بندی گراف و قالب‌بندی مسأله است. در قالب‌بندی گراف، گراف ناهمگن ورودی به زیرگراف‌های همگن تقسیم‌بندی شده و در قالب‌بندی مسأله از پیش‌بینی شباهت بین زیرگراف‌ها برای ایجاد هم‌ریختی بین مسائل مختلف استفاده می‌شود. در این الگوریتم، رویکرد پرامپت-دوگانه^{۱۵} نیز برای تنظیم هر مسأله پایین‌دست به کار می‌رود. پرامپت-دوگانه شامل پرامپت ویژگی و پرامپت ناهمگنی است که پرامپت ویژگی برای تعیین میزان اهمیت هر ویژگی بر حسب مسأله و پرامپت ناهمگنی

few-shot learning^۸
pre-training and fine-tuning^۹
downstream tasks^{۱۰}
heterogeneity gap^{۱۱}
distribution shifts^{۱۲}
prompt learning^{۱۳}
dual-template^{۱۴}
dual-prompt^{۱۵}

embedding^{۱۶}
metapath^{۱۷}
graph convolutional network^{۱۸}

ساختارهای همسایگی تشکیل شده توسط فرامسیرها را توصیف کند. سپس به ازای یک رأس، جاسازی‌های مختلفی برای هر حوزه‌ای که به آن تعلق دارد حاصل می‌شود. در انتها به واسطه یک لایه تجمیع کانولوشن یک بعدی، جاسازی‌های مختلف به دست آمده برای هر رأس، یکپارچه‌سازی می‌شود. این الگوریتم نیز مانند سایر الگوریتم‌های مبتنی بر فرامسیر، با چالش‌های موجود در تعریف فرامسیرها مواجهه است.

اخیرا رویکرد پیش‌آموزش-تنظیم دقیق در گراف ارائه شده است [۷] که در آن بدون در اختیار داشتن داده‌های برچسب‌دار، فرایند یادگیری خودناظر^{۱۹} انجام می‌شود. در مرحله پیش‌آموزش، یادگیری بدون وابستگی به مسأله‌ای خاص انجام شده و در مرحله‌ی تنظیم دقیق بر اساس مسأله پایین‌دست، وزن‌های مدل پیش‌آموزش بروزرسانی می‌شود. این دو مرحله به مدل این امکان را می‌دهند که ابتدا دانش عمومی را بر اساس ساختار گراف کسب کند و سپس آن را برای حل مسائل خاص‌تر و پیچیده‌تر تنظیم کند که این امر منجر به بهبود عملکرد در وظایف پایین‌دستی می‌شود. طی سالیان اخیر با تمرکز بر مفهوم پیش‌آموزش-تنظیم دقیق فعالیت‌هایی روی جنبه‌های مختلف گراف مانند ویژگی‌های رئوس [۱۷]، یال‌ها [۱۸] و الگوهای موجود در گراف [۱۷] ارائه شده است. در این پژوهش‌ها شکاف بین مسائل پیش‌آموزش و پایین‌دست در نظر گرفته نشده و همچنین قابلیت کاربرد روی گراف‌های ناهمگن هم وجود ندارد.

یادگیری با مجموعه داده محدود که نوعی یادگیری متا^{۲۰} محسوب می‌شود، یک رویکرد قانع‌کننده برای آموزش طبقه‌بندهای مؤثر با استفاده از نمونه‌های آموزش محدود و با تعداد کم است [۶]. Hmnet [۱۹] یک روش یادگیری با مجموعه داده محدود است که مسأله پیش‌بینی ایجاد یال روی گراف دانش را در شرایطی که تعداد داده برچسب‌دار آموزشی کمی در دسترس باشد بررسی می‌کند. این روش از اطلاعات ساختاری (ارتباطات بین رئوس) برای غنی کردن گراف بهره می‌برد.

در [۹] پژوهشی برای استفاده از پرامپت در حوزه پردازش زبان طبیعی ارائه شده است. این پژوهش نشان داد که مدل‌های زبانی بزرگ، مانند GPT-3، می‌توانند با استفاده از ورودی‌های متنی مناسب و بدون نیاز به تنظیم دقیق، به طور مؤثر وظایف مختلف پردازش زبان طبیعی را انجام دهند. در این راستا، مفهوم پرامپت به عنوان ابزاری کلیدی برای هدایت رفتار مدل‌ها و استخراج اطلاعات مورد نظر در زمینه‌های مختلف مطرح می‌شود. البته این موضوع در حوزه پردازش گراف هنوز با محدودیت‌های زیادی مواجهه است [۶].

در Hgprompt [۱۰] ابتدا بر اساس مسأله پیش‌بینی ایجاد یال، مرحله پیش‌آموزش روی گراف انجام می‌شود. پیش‌آموزش بدون استفاده از داده برچسب‌دار انجام شده و صرفاً از ساختار گراف استفاده می‌کند. دو مکانیزم قالب-دوگانه و پرامپت-دوگانه را می‌توان پایه‌های این الگوریتم دانست. یک قالب، مسئولیت مشابه‌سازی مسائل پایین‌دست با مسأله پیش‌آموزش را برعهده دارد و قالب دیگر گراف ناهمگن را به مجموعه‌ای از گراف‌های همگن تفکیک می‌کند. با

استفاده از یادگیری پرامپت و ویژگی، تأثیر هر ویژگی در هر رأس به ازای مسائل پایین‌دست تنظیم شده و با پرامپت ناهمگنی، میانگین‌گیری وزن‌دار روی مجموعه گراف‌های همگن انجام می‌شود. در مرحله تفکیک گراف ناهمگن به چند زیرگراف همگن، حجم زیادی از اطلاعات از بین می‌رود که با در نظر گرفتن گراف اولیه به صورت همگن (حذف تنوع رئوس و یال‌ها) تا حدودی این مشکل بهبود داده شده است. ولی اگر اطلاعات ساختاری کمی از گراف استخراج شود، چالش‌های موجود در آموزش با داده‌های محدود تشدید خواهد شد.

۳. الگوریتم پیشنهادی

در دنیای امروز بسیاری از سیستم‌ها از موجودیت‌های متنوعی تشکیل شده‌اند که به روش‌هایی پیچیده با یکدیگر در تعامل هستند. تعاملات و موجودیت‌های این سیستم‌ها را نمی‌توان با یک گراف ساده نشان داد و باید از ساختار گراف‌های ناهمگن بهره برد. در گراف ناهمگن مجموع تعداد نوع رئوس و یال‌ها باید بیش از دو باشد و این نقطه تمایز بین گراف‌های همگن و ناهمگن است. راهکار سنتی برای مواجهه با گراف‌های ناهمگن استفاده از فرامسیرها بوده است که در آن معمولاً با چالش‌هایی مانند تعرف فرامسیرها، ارزیابی و تلفیق نتایج رو به رو خواهیم شد.

هدف از این بخش ارائه یک چارچوب کلی برای پردازش گراف‌های ناهمگن بدون اتکا به فرامسیرها است. الگوریتم پیشنهادی را می‌توان به پنج مرحله اصلی تقسیم کرد:

(۱) **تولید زیرگراف‌های ناهمگن:** تجزیه گراف ورودی به زیرگراف‌هایی ناهمگن که هر کدام شامل یک نوع رأس و چند نوع یال باشند. این امر کمک می‌کند تا بدون نیاز به استفاده از فرامسیرها، به بررسی گراف ناهمگن بپردازیم. در گذشته با تجزیه گراف ناهمگن به زیرگراف‌های همگن به عنوان قالب‌بندی گراف، حجم قابل توجهی از ارتباطات موجود در نظر گرفته نمی‌شدند.

(۲) **پیش‌آموزش:** هسته اصلی این بخش رمزگذار گراف است که مبنی بر شبکه عصبی گراف، هر یک از زیرگراف‌های ناهمگن تحت یادگیری قرار می‌گیرند و به ازای هر رأس، جاسازی مجزایی به دست می‌آید.

(۳) **یادگیری پرامپت و ویژگی:** طی مرحله پیش‌آموزش، هر رأس با توجه به ساختار زیرگرافی که در آن حضور دارد و رئوسی که در همسایگی‌اش قرار دارند، جاسازی خواهد شد. این جاسازی برای مسأله پیش‌آموزش به دست می‌آید و باید برای مسائل پایین‌دست تطبیق گردد. برای تطبیق جاسازی، از مکانیزم پرامپت و ویژگی‌ها استفاده می‌شود.

(۴) **لایه تجمیع کانولوشنی:** پس از تجزیه‌ی گراف ورودی به تعدادی زیر گراف، نیاز است اطلاعات در یک مکان تجمیع شود. در تجمیع اطلاعات زیرگراف‌های ناهمگن، امکان استخراج اطلاعات ترکیبی به صورت محلی و سراسری فراهم می‌آید. وجود لایه تجمیع می‌تواند برای انتقال بهتر دانش از مسأله پیش‌آموزش به مسائل پایین‌دست، مؤثر باشد.

^{۱۹} self-supervised learning^{۲۰} meta learning

برای فاز پیش‌آموزش، مسأله «پیش‌بینی ایجاد یال» انتخاب می‌شود. چون در این مسأله کم‌ترین وابستگی به داده‌های برچسب‌دار آموزشی و هم‌چنین کم‌ترین وابستگی بین مسأله پیش‌آموزش و مسائل پایین‌دست وجود دارد. در واقع مسأله پیش‌بینی ایجاد یال می‌تواند صرفاً بر اساس ساختار گراف انجام شود و انتظار می‌رود جاسازی حاصل از این مرحله بتواند به خوبی دانش مرتبط با ارتباطات بین رئوس را در خود تعبیه کند. هسته اصلی این بخش درک ارتباطات شکل گرفته و شکل نگرفته بین زوج رئوس در هر زیرگراف است. این مسأله را می‌توان به صورت یک یادگیری سه-گانه (v, b, c) برای رأس هدف v که به b متصل و به c متصل نیست، دید و تابع هزینه آن را به گونه‌ای فرموله کرد که نشانگر وجود یال بین v و b و عدم وجود یال بین v و c باشد. در این‌جا برای پیش‌بینی ایجاد یال، مدل یاد می‌گیرد از شباهت زیرگراف‌ها استفاده کند. ابتدا زیرگراف‌های موقت $Stemp(v)$ ، $Stemp(b)$ و $Stemp(c)$ از گراف اصلی که به ترتیب تنها شامل رئوس v ، b و c و برخی از همسایگان‌شان هستند در نظر گرفته می‌شود و سپس فرایند جاسازی رئوس به گونه‌ای انجام می‌شود که رابطه زیر برقرار باشد:

$$\text{sim}(Stemp(v), Stemp(b)) > \text{sim}(Stemp(v), Stemp(c)) \quad (1)$$

در این فرمول برای محاسبه شباهت می‌توان از معیار فاصله کسینوسی استفاده کرد و تابع هزینه با رویکرد خودناظر تضاد^{۲۱} به صورت زیر تعریف خواهد شد:

- بیشینه‌سازی شباهت روی زیرگراف‌هایی که رئوس آن‌ها به یکدیگر متصل هستند (زوج مثبت^{۲۲}).
- کمینه‌سازی شباهت روی زیرگراف‌هایی که رئوس آن‌ها به یکدیگر اتصال ندارند (زوج منفی^{۲۳}).

زمانی که پیش‌آموزش به اتمام رسد، وزن‌های شبکه ثابت و بدون تغییر خواهند ماند و امکان بروزرسانی برای آن‌ها وجود ندارد.

همان‌طور که در بالا اشاره شد، برای یکسان‌سازی طیف مسائل پایین‌دست، محاسبه شباهت زیرگراف به عنوان مسأله پایه در نظر گرفته شده است. با استفاده از محاسبه شباهت زیرگراف می‌توان هر سه نوع مسأله پیش‌بینی ایجاد یال، دسته‌بندی رئوس و گراف را با دیدگاهی یکسان حل کرد.

۳-۳. یادگیری پرامپت ویزگی

برای انتقال مؤثر دانش بین مسأله پیش‌آموزش و مسائل پایین‌دست، باید علاوه بر طراحی بستر یکنواخت و عام منظوره، برای هر مسأله پایین‌دست نیز به صورت خاص بهینه‌سازی انجام شود. با همین رویکرد، پرامپت در پردازش زبان طبیعی طراحی و استفاده شده است [۹]. اما طراحی پرامپت در گراف متفاوت است. زیرا در مسائل گراف جاسازی رئوس و ارتباطات بین رئوس وجود دارد. پرامپت در گراف انتزاعی بوده و مانند پرامپت پردازش زبان طبیعی از نوع زبانی نیست. پس امکان تعریف پرامپت گراف به صورت دستی وجود ندارد و صرفاً باید با فرایند یادگیری آموخته شود.

(۵) **پیش‌بینی مسائل پایین‌دست:** برای انواع مسائل مانند دسته‌بندی رئوس و گراف باید رویکردی مشابه با مسأله پیش‌آموزش معرفی شود. در هر مسأله جاسازی رأس یا زیرگراف با جاسازی‌های کلاس‌های مختلف مقایسه و کلاس مشابه به عنوان برچسب ارائه می‌شود.

در شکل ۱ نمای کلی الگوریتم پیشنهادی در مرحله پیش‌آموزش و برای مسأله پایین‌دست دسته‌بندی رئوس آمده است. هم‌چنین در ادامه به تشریح هر مرحله خواهیم پرداخت.

۳-۱. تولید زیرگراف‌های ناهمگن

فرض کنید گراف $G^H = (V, E, T_v, T_e)$ یک گراف ناهمگن بی‌وزن و بدون جهت است که در آن V مجموعه رئوس و E مجموعه یال‌ها می‌باشد. T_e و T_v به ترتیب نمایانگر مجموعه انواع رئوس و یال‌ها هستند. به ازای هر $a_k \in T_v$ ، زیرگراف S_{a_k} از گراف G^H را به گونه‌ای تعریف می‌کنیم که فقط شامل رئوس از نوع a_k است و داریم $S = \{S_{a_1}, S_{a_2}, \dots\}, \forall a_k \in T_v$.

تولید زیرگراف‌ها در دو مرحله انجام می‌شود. در مرحله اول به زیرگراف‌های همگن و در مرحله دوم پس از غنی‌سازی، به زیرگراف‌های ناهمگن خواهیم رسید.

مرحله اول: به ازای هر نوع رأس a_k ، زیرگراف S_{a_k} انتخاب می‌شود که هدف از این کار در اختیار داشتن زیرگراف‌هایی با تنوع رأس محدود است، درحالی‌که ارتباطات بین رئوس نیز حفظ شده باشد.

مرحله دوم: به ازای هر جفت رأس $(n_i, n_j) \in V_{S_{a_k}}$ زمانی که یالی بین آن‌ها وجود ندارد یعنی $(n_i, n_j) \notin E_{S_{a_k}}$ ، جستجو برای یافتن رأس r در گراف اصلی G^H صورت می‌گیرد به طوری که شروط زیر برای r برقرار باشد:

- $r \in \{N(n_i) \cap N(n_j)\}$,
- $type(k) \neq a_k$,

که $N(n)$ رئوس همسایه رأس n و $type(k)$ نوع رأس k را نشان می‌دهد. بر اساس این شرط‌ها، ایجاد یال بین رئوس n_j و n_i تنها زمانی صورت می‌گیرد که یک یا بیش از یک رأس میانی (r) شناسایی شده باشد. به یال اضافه شده بین n_j و n_i **یال غنی‌سازی** می‌گوییم. یال غنی‌سازی می‌تواند از انواع مختلفی باشد که قبلاً در زیرگراف وجود نداشته است و این تنوع یال سبب تغییر حالت زیرگراف همگن به زیرگراف ناهمگن می‌شود. در این حالت اگرچه هیچ رأس دیگری به زیرگراف اضافه نشده اما چگالی زیرگراف افزایش یافته است که این افزایش چگالی ناشی از افزودن اطلاعات ساختاری است. در واقع این ارتباطات در گراف G^H به صورت ارتباطات پنهان وجود داشته و افزودن آن‌ها در شرایطی که از هیچ داده برچسب‌دار آموزشی استفاده نشده می‌تواند شکاف ناهمگنی و تغییرات توزیع را بهبود دهد.

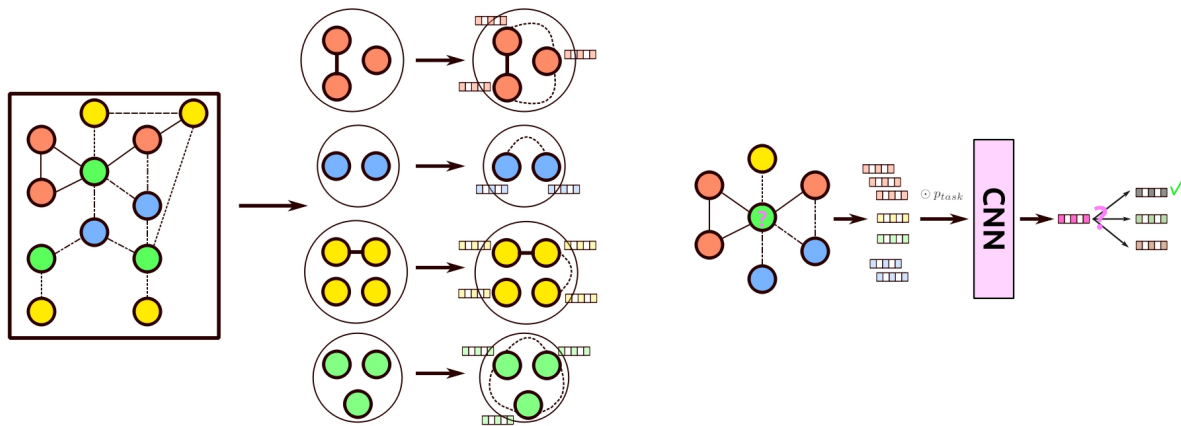
۳-۲. پیش‌آموزش

پس از تجزیه‌ی گراف G^H به مجموعه‌ای از زیرگراف‌های ناهمگن غنی‌شده، به ازای هر رأس n_i^k که در زیرگراف S_{a_k} قرار دارد، بردار جاسازی $\vec{X}(n_i^k)$ با روش یادگیری خودناظر محاسبه می‌شود. معمولاً

^{۲۱} self-supervised contrastive learning

^{۲۲} positive pairs

^{۲۳} negative pairs



شکل ۱: نمای کلی الگوریتم پیشنهادی: ابتدا گراف ناهمگن به چند زیرگراف همگن تجزیه و پس از شناسایی مسیرهای غنی‌ساز در گراف اصلی، یال‌های غنی‌سازی به زیرگراف‌ها اضافه می‌شود. فرایند پیش‌آموزش با رویکرد یادگیری خودناظر تضاد انجام شده و جاسازی‌های رئوس محاسبه می‌شود. در مسأله پایین‌دست دسته‌بندی رئوس، جاسازی‌های رئوس همسایگان رأس هدف، پس از اعمال پرامپت ویژگی به لایه تجمیع ارسال شده و در نهایت جاسازی خروجی با جاسازی‌های کلاس‌های نمونه مقایسه شده و برچسب کلاس رأس هدف تعیین می‌شود.

فرایند آموزش این لایه مبتنی بر مجموعه داده محدود است. به ازای هر مسأله پایین‌دست، با در اختیار داشتن داده محدود برچسب‌دار آموزشی، یادگیری وزن‌های این شبکه انجام می‌شود. انتظار می‌رود دانش ساختاری که در زیرگراف‌ها حاصل شده است، با کمک یادگیری وزن‌های خاص منظوره لایه تجمیع، به نتایج خوبی منتج شود. در واقع این لایه علاوه بر آن‌که ارتباط بین زیرگراف‌های مختلف را درک و تأثیر هرکدام را به ازای مسائل مختلف تغییر می‌دهد، می‌تواند اطلاعات مورد نیاز برای انطباق بهتر جاسازی رئوس و مسائل پایین‌دست را هم اضافه کند.

به منظور انسجام قوی در چارچوب ارائه شده، فرایند یادگیری پرامپت ویژگی و شبکه عصبی کانولوشن (لایه تجمیع) به صورت هم‌زمان انجام می‌شود. این یادگیری هم‌زمان، کمک می‌کند تا یکنواختی مناسبی برای حل هر مسأله پایین‌دست حاصل آید.

۳-۵. پیش‌بینی مسائل پایین‌دست

مشابه الگوریتم Hgprompt [۱۰]، مسأله پیش‌بینی شباهت روی زیرگراف، برای یکسان‌سازی مسأله پیش‌آموزش و مسائل پایین‌دست مورد استفاده قرار می‌گیرد و باید همه مسائل پایین‌دست به همین فرم تبدیل شوند. از مسائل پایین‌دست معمول، می‌توان مسأله دسته‌بندی رئوس و دسته‌بندی گراف را در نظر گرفت که در ادامه به توضیح هر یک خواهیم پرداخت.

- **مسأله دسته‌بندی رئوس:** مسأله دسته‌بندی رئوس، مسأله‌ای در حوزه رأس است که هدف آن تخمین برچسب رئوس می‌باشد. فرض کنید مجموعه محدود $D = \{(n_1, c_1), (n_2, c_2), \dots\}$ که در آن c_i برچسب رأس n_i است، در اختیار باشد. جاسازی کلاس C_i که از میانگین‌گیری روی نمونه رئوس دارای برچسب c_i به دست می‌آید، به صورت زیر تعریف می‌شود:

$$\bar{\mathbf{X}}(C_i) = \frac{1}{|C_i|} \sum_{(n_j, c_j) \in D, c_j = c_i} \bar{\mathbf{Y}}(n_j). \quad (4)$$

پرامپت ویژگی برای انطباق مناسب‌تر بین مسأله پیش‌آموزش و مسائل پایین‌دست، به ویژگی‌های رئوس مرتبط با مسأله پایین‌دست وزن بیش‌تری می‌دهد. به عبارت دیگر، برای دسته مسائل مختلف، برخی از ویژگی‌های رئوس مهم‌تر از سایر ویژگی‌ها هستند که قرار است پرامپت ویژگی، این اهمیت را یاد بگیرد. به عنوان مثال زمانی که با مسأله دسته‌بندی رئوس مواجه هستیم، ارتباطات بین همسایه‌ها بسیار مهم‌تر از ارتباطات سراسری گراف است. برای این منظور هر المان جاساز رأسی در المان متناظرش در بردار پرامپت ضرب شده و مطابق فرمول زیر بردارهای جدید جاسازی به دست می‌آیند.

$$\bar{\mathbf{Y}}(n_i^k) = \bar{\mathbf{X}}(n_i^k) \odot p_{task}. \quad (2)$$

در این‌جا بردار پرامپت (p_{task}) به ازای هر مسأله پایین‌دست به صورت مستقل محاسبه خواهد شد.

۳-۴. لایه تجمیع کانولوشنی

زمانی که گراف G^H به مجموعه‌ای از زیرگراف‌های ناهمگن تجزیه می‌شود، باید روشی برای ترکیب اطلاعات حاصل از زیرگراف‌ها نیز طراحی شود. هدف از لایه تجمیع، ترکیب و استخراج اطلاعات محلی و سراسری از زیرگراف‌ها و ارائه یک جاسازی یکنواخت برای کل گراف است. این تجمیع انعطاف زیادی را برای چارچوب فراهم می‌آورد تا برای طیف مختلف مسائل پایین‌دست به نتایج مطلوبی برسد.

لایه تجمیع از یک شبکه عصبی کانولوشنی سبک (کم عمق) ساخته شده و بار محاسباتی کمی را بر چارچوب تحمیل می‌کند. این لایه می‌تواند ارتباطات پیچیده را به خوبی شناسایی کند. ورودی این لایه کانولوشنی از خروجی جاسازی رئوس زیرگراف‌ها تأمین می‌شود و خروجی آن جاسازی $\bar{\mathbf{X}}(S_{test}^H)$ برای زیرگراف ناهمگن S_{test}^H از گراف پایه G^H است:

$$\bar{\mathbf{X}}(S_{test}^H) = CNN\left(\bar{\mathbf{Y}}(n_i^k) \mid n_i^k \in S_{test}^H\right) \quad (3)$$

(۲) DBLP: مجموعه داده DBLP نیز یک مجموعه داده استادی است که در آن مقالاتی در چهار حوزه پایگاه داده، داده‌کاوی، هوش مصنوعی و بازیابی اطلاعات گرد آمده و مؤلفین به عنوان هدف دسته‌بندی مورد بررسی قرار می‌گیرند. این مجموعه داده به گرافی ناهمگن با ۴ نوع رأس مؤلف، مقاله، کنفرانس و عبارت کلیدی و ۳ نوع یال مقاله-مؤلف، مقاله-عبارت کلیدی و مقاله-کنفرانس تبدیل می‌شود.

(۳) Freebase: مجموعه داده Freebase یک پایگاه دانش مشترک بزرگ متشکل از داده‌هایی ساختاریافته از انواع کتاب، فیلم، موسیقی و ... بوده که توسط اعضای جامعه آن ایجاد شده است. در این‌جا نوع کتاب مورد بررسی قرار می‌گیرد و هشت نوع برچسب کلاس مختلف برای آن وجود دارد. گراف ناهمگن این مجموعه داده، شامل رئوس کتاب، فیلم، موسیقی، ورزشی، جامعه‌شناسی، مکان، سازمان و مراکز کسب و کار و ۳۶ نوع یال از انواع ارتباطات بین این رئوس مانند کتاب-کتاب، کتاب-فیلم، کتاب-ورزش، کتاب-مکان، کتاب-سازمان و فیلم-فیلم است.

در جدول‌های ۲ و ۳ نتایج حاصل از اجرای روش پیشنهادی روی سه مجموعه داده معرفی شده و مقایسه آن‌ها با هشت الگوریتم ارائه شده است. الگوریتم‌هایی که برای مقایسه روش پیشنهادی انتخاب شده‌اند را می‌توان از منظر قابلیت مواجه با مجموعه داده‌های ناهمگن، به دو دسته تقسیم کرد. یک دسته، روش‌هایی هستند که قابلیت تشخیص ناهمگنی گراف را نداشته و کل گراف ورودی را به صورت همگن پردازش می‌کنند. دسته دیگر الگوریتم‌هایی هستند که امکان پردازش گراف‌های ناهمگن را داشته و تفاوت بین نوع رئوس و یال‌ها را درک می‌کنند. در ادامه مختصراً به توضیح هر یک از این روش‌ها می‌پردازیم.

• روش‌های همگن: الگوریتم‌های GCN [۱۵] و GAT [۲۱]

تنها قابلیت پردازش گراف‌های همگن را دارند و جزء روش‌های کلاسیک پردازش گراف محسوب می‌شوند. GCN الهام‌گرفته از شبکه عصبی پیچشی است که قابلیت پیچش روی ساختمان داده گراف برای آن طراحی شده است. در این روش خصوصیات هر همسایه با میانگین‌گیری از خصوصیات همسایگانش بروزرسانی می‌شود. زمانی که بتوان در پیچش تأثیر هر همسایه را کم یا زیاد کرد به الگوریتم GAT خواهیم رسید.

الگوریتم GraphCL [۲۲] مبتنی بر یادگیری خود ناظر تقابلی است. در این الگوریتم برای پیش‌آموزش در رویکرد یادگیری پیش‌آموزش-تنظیم دقیق، هیچ داده برچسب‌دار آموزشی استفاده نشده و بر اساس ارتباطات ساختاری گراف، یادگیری خودناظر صورت می‌گیرد. این یادگیری بر پایه سه استراتژی مختلف رأس، یال و زیرگراف استوار است. در زمان تنظیم دقیق و برای مسائل پایین دست، بر اساس داده‌های برچسب‌دار آموزشی، وزن‌های شبکه پیش‌آموزش بروزرسانی می‌شوند. اگرچه این الگوریتم قابلیت مواجه با مسائل پایین دستی در سطح

برای تخمین برچسب رأس جدیدی مانند v_{test} ، ابتدا زیرگرافی شامل همه همسایه‌های مستقیم v_{test} در نظر گرفته شده و سپس در لایه تجمیع کانولوشنی، جاسازی آن یعنی $\vec{X}(S_{test}^H)$ محاسبه می‌شود. در نهایت مطابق فرمول زیر، جاسازی این زیرگراف با جاسازی‌های کلاس‌های C_i مقایسه و برچسب آن برابر با برچسب کلاسی می‌شود که بیش‌ترین شباهت برداری بین آن‌ها وجود داشته باشد.

$$c_{v_{test}} = \arg \max_{C_i} \text{sim}(\vec{X}(S_{test}^H), \vec{X}(C_i)). \quad (5)$$

• **مسئله دسته‌بندی گراف:** برای مسئله دسته‌بندی گراف، فرایند مشابه دسته‌بندی رأس است با این تفاوت که به علت فقدان داده‌های برچسب‌دار آموزشی برای مسئله دسته‌بندی گراف، می‌توان از داده‌های برچسب‌دار مسئله دسته‌بندی رئوس استفاده کرد [۲۰]. برای این منظور، به صورت تصادفی یک رأس از مجموعه D انتخاب می‌شود که آن را رأس آگو می‌نامیم. زیرگرافی که شامل رأس آگو و همسایگان آن باشد را شبکه آگو نامیده و آن را با S_{test}^H نشان می‌دهیم. یک برچسب کلاس انجام می‌شود. ابتدا بردار جاسازی آگو از میانگین‌گیری جاسازی‌های همه رئوس حاضر در آن محاسبه می‌شود. سپس مانند روش دسته‌بندی رئوس، بردار جاسازی آگو با جاسازی‌های کلاس‌های C_i مقایسه شده و مطابق فرمول (۵) برچسب آن برابر با برچسب کلاسی می‌شود که بیش‌ترین شباهت برداری بین آن‌ها وجود داشته باشد. زمانی که تعداد برچسب‌های آموزشی محدودی در دسترس باشد این رویکرد می‌تواند مفید باشد. زیرا در این الگو تمرکز بر ساختارهای محلی است و با در اختیار داشتن تعداد کمی نمونه آموزشی می‌توان گراف را به صورت محلی (شبکه‌های آگو) برچسب‌دار کرد. هم‌چنین در این روش برای کنترل پیچیدگی محاسباتی، می‌توان با تغییر پارامترهای کنترلی، اندازه شبکه‌های آگو را کوچک یا بزرگ کرد.

۴. ارزیابی و مقایسه

پیاده‌سازی این چارچوب با پایتورچ انجام شده و مسئله پیش‌بینی ایجاد یال به عنوان مسئله پیش‌آموزش و مسائل دسته‌بندی رئوس و دسته‌بندی گراف به عنوان مسائل پایین‌دست در نظر گرفته شده است. برای ارزیابی عملکرد الگوریتم، آزمایشاتی روی سه مجموعه داده شناخته شده زیر که آمار آن‌ها در جدول ۱ ارائه شده، صورت گرفته است.

(۱) ACM: مجموعه داده ACM شامل اطلاعاتی از تعدادی مقاله، مؤلفان آن‌ها، عناوین و عبارات کلیدی آن‌ها بوده که از پنج کنفرانس و در سه حوزه داده‌کاوی، پایگاه داده و ارتباطات بی‌سیم گردآوری شده است. در این مجموعه داده مقاله به عنوان هدف دسته‌بندی و سه حوزه بالا به عنوان برچسب کلاس، مورد بررسی قرار می‌گیرد. برای پردازش، این مجموعه داده را به گرافی ناهمگن با ۴ نوع رأس مقاله، مؤلف، عنوان و عبارت کلیدی و ۴ نوع یال مقاله-مقاله (ارجاع)، مقاله-مؤلف (نگارش شده) مقاله-عنوان و مقاله-عبارت کلیدی تبدیل می‌کنیم.

مرحله پیش آموزش) منتج به دقت بهتری در مسائل دسته‌بندی رئوس و دسته‌بندی گراف شده است. به علت شباهت مسأله پیش‌آموزش در GraphCL (یادگیری ساختار کلی گراف) به دسته‌بندی گراف، نتایج این الگوریتم روی مسأله دسته‌بندی گراف مناسب است. همچنین به علت توانایی روش CPT-HG در مواجهه با گراف‌های ناهمگن، نتایج این الگوریتم در هر دو مسأله و هر سه مجموعه داده نسبت به GraphCL برتری دارد.

الگوریتم پیشنهادی در هر سه مجموعه داده و روی هر دو مسأله دسته‌بندی رئوس و دسته‌بندی گراف، از الگوریتم‌های کلاسیک و مدرن پردازش گراف‌های همگن و ناهمگن برتری دارد. برتری روش پیشنهادی به علت انتشار مناسب اطلاعات از مرحله پیش‌آموزش به مسائل پایین دست، قابلیت استخراج دانش مناسب از گراف ناهمگن ورودی و توانایی لایه تجمیع در ترکیب و وزن‌دهی مناسب به زیرگراف‌های ناهمگن است.

جدول ۱: اطلاعات مجموعه داده‌ها.

مجموعه داده	تعداد رأس	تعداد یال	نوع رأس هدف	تعداد کلاس‌ها
ACM	۱۰,۹۴۲	۵۴۷,۸۷۲	مقاله	۳
DBLP	۲۶,۱۲۸	۲۳۹,۵۶۶	مؤلف	۴
Freebase	۱۸۰,۰۹۸	۱,۰۵۷,۶۸۸	کتاب	۸

جدول ۲: نتایج دسته‌بندی رئوس.

مجموعه داده	ACM		DBLP		Freebase	
	MacroF	MicroF	MacroF	MicroF	MacroF	MicroF
GCN	۰/۵۰	۰/۴۴	۰/۵۱	۰/۴۸	۰/۱۷	۰/۱۵
GAT	۰/۳۸	۰/۲۸	۰/۶۵	۰/۶۲	۰/۱۷	۰/۱۶
GraphCL	۰/۵۸	۰/۵۵	۰/۶۶	۰/۶۴	۰/۱۸	۰/۱۶
GPPT	۰/۵۴	۰/۵۱	۰/۶۱	۰/۶۳	۰/۱۷	۰/۱۶
HAN	۰/۶۲	۰/۵۷	۰/۶۲	۰/۶۰	۰/۱۸	۰/۱۶
Simple-HGN	۰/۴۵	۰/۴۰	۰/۵۹	۰/۵۷	۰/۱۷	۰/۱۶
CPT-HG	۰/۶۲	۰/۵۸	۰/۷۰	۰/۶۷	۰/۱۹	۰/۱۷
Hgprompt+	۰/۷۵	۰/۷۳	۰/۸۲	۰/۷۷	۰/۲۲	۰/۲۱
روش پیشنهادی	۰/۸۰	۰/۷۹	۰/۸۹	۰/۸۷	۰/۲۴	۰/۲۳

جدول ۳: نتایج دسته‌بندی گراف.

مجموعه داده	ACM		DBLP		Freebase	
	MacroF	MicroF	MacroF	MicroF	MacroF	MicroF
GCN	۰/۳۳	۰/۱۷	۰/۳۸	۰/۳۵	۰/۱۷	۰/۱۶
GAT	۰/۳۳	۰/۱۷	۰/۴۶	۰/۴۰	۰/۱۶	۰/۱۶
GraphCL	۰/۳۳	۰/۲۱	۰/۵۰	۰/۴۴	۰/۱۸	۰/۱۷
GPPT	-	-	-	-	-	-
HAN	۰/۳۵	۰/۲۲	۰/۴۷	۰/۴۲	۰/۱۸	۰/۱۷
Simple-HGN	۰/۳۳	۰/۱۶	۰/۴۴	۰/۳۹	۰/۱۷	۰/۱۶
CPT-HG	۰/۳۵	۰/۲۷	۰/۵۳	۰/۴۶	۰/۱۸	۰/۱۷
Hgprompt+	۰/۳۹	۰/۳۲	۰/۶۰	۰/۵۲	۰/۲۱	۰/۲۰
روش پیشنهادی	۰/۴۶	۰/۴۲	۰/۶۱	۰/۶۰	۰/۲۲	۰/۲۱

رأس، یال و زیر گراف را دارد، اما نمی‌تواند روی گراف‌های ناهمگن پردازش انجام دهد. در GPPT [۲۳] هدف پرکردن شکاف بین مسأله پیش‌آموزش و مسائل پایین‌دست است. این الگوریتم مبتنی بر پرامپت بوده و در مرحله پیش‌آموزش، مسأله پیش‌بینی ایجاد یال را فرا می‌گیرد. در این الگوریتم برای بهبود مشکل تنظیم دقیق کل مدل پیش‌آموزش، تنها قسمت‌های مورد نیاز مسأله پایین دست توسط پرامپت شناسایی شده و فقط همان بخش‌ها تنظیم دقیق می‌شوند.

● **روش‌های ناهمگن:** برای پردازش گراف‌های ناهمگن الگوریتم HAN [۲۴] نقطه آغاز روش‌های مبتنی بر شبکه عصبی گرافی است. این الگوریتم بر طراحی مناسب فرامسیرها استوار است و قابلیت وزن‌دهی به فرامسیرها با مکانیزم توجه^{۲۵} در آن دیده شده است. الگوریتم Simple-HGN [۲۰] به فرامسیر وابستگی ندارد و بر خلاف الگوریتم GAT که مکانیزم توجه آن روی رئوس است، در این الگوریتم این مکانیزم روی یال‌ها اعمال شده و می‌تواند بین ارتباطات مختلف وزن‌دهی کند. این روش تعداد پارامترهای زیادی دارد و زمانی که تعداد محدودی داده آموزشی برچسب‌دار در دسترس باشد، یادگیری به خوبی انجام نخواهد شد. الگوریتم CPT-HG [۲۵] بر پیش‌آموزش به صورت خودناظر متکی است و دو سطح ارتباطات بین رئوس و زیر گراف را فرا می‌گیرد. این روش از ارتباطات موجود به عنوان نمونه‌های مثبت و برای تعریف نمونه‌های منفی از ارتباطات ناسازگار و بی‌ربط ساختگی استفاده می‌کند. در الگوریتم Hgprompt+ [۱۰] رویکرد قالب-دوگانه و پرامپت-دوگانه عاملی برای قابلیت پردازش گراف‌های ناهمگن هستند. اگرچه این الگوریتم می‌تواند با داده‌های آموزشی برچسب‌دار محدود برای تنظیم پرامپت کار کند، اما دانشی که در مرحله قالب گراف از دست می‌رود سبب کاهش دقت آن شده است.

در این مقاله، برای جلوگیری از خطا در الگوریتم‌های قبلی، همه آن‌ها بر اساس سورس‌کد رسمی‌شان پیاده‌سازی مجدد شده‌اند و تنها تفاوت به‌وجودآمده مربوط به هماهنگ‌سازی بارگذاری داده‌ها و مسائل پایین دست است.

در جدول‌های ۲ و ۳ برای ارزیابی الگوریتم ارائه‌شده از معیارهای MicroF و MacroF [۲۰] استفاده شده است. این دو معیار بر اساس معیار F1 که میانگین precision و recall است طراحی شده‌اند. در MicroF همه مجموعه داده یکباره محاسبه شده و به همه کلاس‌ها وزن‌های یکسانی داده می‌شود، درحالی‌که در MacroF ابتدا برای هر کلاس به طور مجزا محاسبه انجام می‌شود و سپس به هر کلاس برحسب اندازه‌اش وزن داده می‌شود.

همان‌طور که در جدول نتایج مشاهده می‌شود، نتایج الگوریتم‌های ناهمگن از دقت بهتری برخوردار است. از مقایسه روش‌های GCN و GAT مشخص می‌شود که مکانیزم توجه روی مجموعه داده DBLP (در

۵. نتیجه‌گیری

در این مقاله پردازش گراف ناهمگن مبتنی بر شبکه عصبی گرافی بررسی شده است. روش پیش‌آموزش - تنظیم پرامپت به عنوان رویکرد اصلی در نظر گرفته شده است که در مرحله پیش‌آموزش با حل مسأله پیش‌بینی ایجاد یال روی زیرگراف‌های غنی‌شده، جاسازی‌های رئوس محاسبه می‌شود. آموزش هم‌زمان پرامپت ویژگی و لایه تجمیع توانست به خوبی جاسازی‌های رئوس در زیرگراف‌های ناهمگن را برحسب مسأله پایین دست، ترکیب کند. دقت مناسب الگوریتم پیشنهادی در مقایسه با سایر الگوریتم‌های کلاسیک و مدرن پردازش گراف نشان داد که در روش پیشنهادی شکاف بین مسأله پیش‌آموزش و مسائل پایین دست رفع شده است.

مراجع

- [10] X. Yu, Y. Fang, Z. Liu, and X. Zhang, "Hgprompt: Bridging homogeneous and heterogeneous graphs for few-shot prompt learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol.38, pp.16578–16586, 2024.
- [11] B. Perozzi, R. Al-Rfou, and S. Skiena, "Deepwalk: Online learning of social representations," in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp.701–710, 2014.
- [12] A. Grover and J. Leskovec, "node2vec: Scalable feature learning for networks," in *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pp.855–864, 2016.
- [13] Y. Dong, N. V. Chawla, and A. Swami, "metapath2vec: Scalable representation learning for heterogeneous networks," in *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, pp.135–144, 2017.
- [14] T.-y. Fu, W.-C. Lee, and Z. Lei, "Hin2vec: Explore meta-paths in heterogeneous information networks for representation learning," in *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pp.1797–1806, 2017.
- [15] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.
- [16] X. Fu and I. King, "MECCH: metapath context convolution-based heterogeneous graph neural networks," *Neural Networks*, vol.170, pp.266–275, 2024.
- [17] W. Hu, B. Liu, J. Gomes, M. Zitnik, P. Liang, V. Pande, and J. Leskovec, "Strategies for pre-training graph neural networks," *arXiv preprint arXiv:1905.12265*, 2019.
- [18] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," *Advances in neural information processing systems*, vol.30, 2017.
- [19] S. Xiao, L. Duan, G. Xie, R. Li, Z. Chen, G. Deng, and J. Nummenmaa, "HMNet: hybrid matching network for few-shot link prediction," in *International Conference on Database Systems for Advanced Applications*, pp.307–322, Springer, 2021.
- [20] Q. Lv, M. Ding, Q. Liu, Y. Chen, W. Feng, S. He, C. Zhou, J. Jiang, Y. Dong, and J. Tang, "Are we really making much progress? revisiting, benchmarking and refining heterogeneous graph neural networks," in *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*, pp.1150–1160, 2021.
- [21] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Lio, Y. Bengio, et al., "Graph attention networks," *stat*, vol.1050, no.20, pp.10–48550, 2017.
- [1] Y. Dong, J. Tang, S. Wu, J. Tian, N. V. Chawla, J. Rao, and H. Cao, "Link prediction and recommendation across heterogeneous social networks," in *2012 IEEE 12th International conference on data mining*, pp.181–190, IEEE, 2012.
- [2] A. Ma, X. Wang, J. Li, C. Wang, T. Xiao, Y. Liu, H. Cheng, J. Wang, Y. Li, Y. Chang, et al., "Single-cell biological network inference using a heterogeneous graph transformer," *Nature Communications*, vol.14, no.1, p.964, 2023.
- [3] W. Liu, Y. Zhang, J. Wang, Y. He, J. Caverlee, P. P. Chan, D. S. Yeung, and P.-A. Heng, "Item relationship graph neural networks for e-commerce," *IEEE Transactions on Neural Networks and Learning Systems*, vol.33, no.9, pp.4785–4799, 2021.
- [4] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and S. Y. Philip, "A comprehensive survey on graph neural networks," *IEEE transactions on neural networks and learning systems*, vol.32, no.1, pp.4–24, 2020.
- [5] C. Zhang, D. Song, C. Huang, A. Swami, and N. V. Chawla, "Heterogeneous graph neural network," in *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pp.793–803, 2019.
- [6] X. Yu, Y. Fang, Z. Liu, Y. Wu, Z. Wen, J. Bo, X. Zhang, and S. C. Hoi, "Few-shot learning on graphs: from meta-learning to pre-training and prompting," *arXiv preprint arXiv:2402.01440*, 2024.
- [7] J. Xia, Y. Zhu, Y. Du, and S. Z. Li, "A survey of pretraining on graphs: Taxonomy, methods, and applications," *arXiv preprint arXiv:2202.07893*, 2022.
- [8] P. Ding, Y. Wang, and G. Liu, "Few-shot learning on heterogeneous graphs: Challenges, progress, and prospects," *arXiv preprint arXiv:2403.13834*, 2024.
- [9] T. B. Brown, "Language models are few-shot learners," *arXiv preprint arXiv:2005.14165*, 2020.

- [22] Y. You, T. Chen, Y. Sui, T. Chen, Z. Wang, and Y. Shen, "Graph contrastive learning with augmentations," *Advances in neural information processing systems*, vol.33, pp.5812–5823, 2020.
- [23] M. Sun, K. Zhou, X. He, Y. Wang, and X. Wang, "Gppt: Graph pre-training and prompt tuning to generalize graph neural networks," in *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp.1717–1727, 2022.
- [24] X. Wang, H. Ji, C. Shi, B. Wang, Y. Ye, P. Cui, and P. S. Yu, "Heterogeneous graph attention network," in *The world wide web conference*, pp.2022–2032, 2019.
- [25] X. Jiang, Y. Lu, Y. Fang, and C. Shi, "Contrastive pre-training of GNNs on heterogeneous graphs," in *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pp.803–812, 2021.