

گزارش پروژه درس یادگیری عمیق

۱- مشخصات اعضای گروه

نام و نام خانوادگی: پارسا عباسی سرابستانی
شماره دانشجویی: ۹۹۷۲۲۰۹۴

نام و نام خانوادگی: محمد ابراهیمی
شماره دانشجویی: ۹۹۷۲۲۰۰۴

۲- مشخصات دستیار آموزشی راهنما

نام و نام خانوادگی: هومن مهرآفرین

۳- موضوع پروژه

شبکه‌های توجه گرافی – Graph Attention Networks (GATs)

۴- واژگان کلیدی

شبکه عصبی گرافی، مکانیزم‌های توجه، شبکه توجه گرافی، شبکه عصبی، تحلیل گراف، گراف نامتجانس

Graph Neural Network, Attention Mechanisms, Neural Network
Graph Attention Network, Graph Analysis, Heterogeneous Graph

۵- مقالات انتخاب شده

- [1] [Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, & Yoshua Bengio \(2018\). Graph Attention Networks. International Conference on Learning Representations.](#)
- [2] [Xiao Wang, Houye Ji, Chuan Shi, Bai Wang, Peng Cui, P. Yu, & Yanfang Ye. \(2021\). Heterogeneous Graph Attention Network.](#)

فهرست مطالب

صفحه	عنوان
۱	۱) شبکه‌های توجه گرافی
۱	۱.۱) معماری لایه
۴	۱.۲) تحلیل و مقایسه مدل
۶	۱.۳) مجموعه داده‌ها
۷	۱.۴) نتایج
۱۰	۲) شبکه توجه گرافی نامتجانس
۱۲	۲.۱) مدل پیشنهادی
۱۳	۲.۱.۱) محاسبه توجه در سطح گره
۱۵	۲.۱.۲) محاسبه توجه در سطح معنایی
۱۷	۲.۲) تجزیه و تحلیل مدل پیشنهادی
۱۸	۲.۳) مجموعه داده‌ها و نتایج

(۱) شبکه‌های توجه گرافی^۱

مکانیزم‌های توجه امروزه به یک استاندارد بالفعل^۲ در بسیاری از مسائل ترتیبی تبدیل شده‌اند. یکی از مهم‌ترین مزایای مکانیزم‌های توجه این است که می‌توانند بر روی ورودی‌هایی با اندازه متغیر عمل کنند و همچنین جهت تصمیم‌گیری، بر روی قسمت‌های مناسب‌تر ورودی تمرکز کنند. شایان ذکر است هنگامیکه یک مکانیزم توجه جهت محاسبه بازنمایی یک دنباله واحد به کار گرفته می‌شود به آن خود-توجه^۳ یا درون-توجه^۴ گفته می‌شود.

با الهام از کارهای پیشین در این مقاله یک معماری بر پایه توجه معرفی شده است تا بتواند بر روی داده‌هایی با ساختار گراف عمل رده‌بندی در سطح راس^۵ را به انجام برساند. ایده اصلی این است که با بکارگیری استراتژی خود-توجهی، بازنمایی‌های پنهان هر راس در گراف، به کمک توجه بر روی همسایگان هر راس محاسبه شود.

معماری توجه دارای چندین خصیصه جالب و مفید است:

۱. عمل توجه کارآمد^۶ است، زیرا که قابل موازی‌سازی بر روی جفت رئوس همسایه است.
۲. با مشخص کردن وزن‌های دلخواه بر روی همسایگان، می‌تواند بر روی راس‌هایی با درجه‌های متفاوت اعمال شود (برخلاف رویکردهای پیشین که به دنبال تعریف یک اندازه مشخص همسایگی بودند).
۳. این مدل می‌تواند بصورت مستقیم بر روی مسائل یادگیری استنتاجی^۷ مورد استفاده قرار بگیرد. در چنین مسائلی مدل باید بر روی گراف‌هایی که تاکنون اصلاً دیده نشده‌اند نیز قابل تعمیم^۸ باشد.

(۱.۱) معماری لایه

ورودی لایه GAT مجموعه‌ای از ویژگی راس‌ها، $h = \{\vec{h}_1, \vec{h}_2, \dots, \vec{h}_N\}$ و $\vec{h}_1 \in \mathbb{R}^F$ است که N بیانگر تعداد رئوس و F تعداد ویژگی هر راس می‌باشد.

لایه مربوطه مجموعه‌ای جدید از ویژگی رئوس، $h' = \{\vec{h}'_1, \vec{h}'_2, \dots, \vec{h}'_N\}$ و $\vec{h}'_1 \in \mathbb{R}^{F'}$ به عنوان خروجی خود تولید می‌کند که F' در آن به معنی کاردینالیتی جدید می‌باشد.

¹ Graph Attention Networks (GATs)

² de facto

³ Self-attention

⁴ Intra-attention

⁵ Node classification

⁶ Efficient

⁷ Inductive learning

⁸ Generalize

به منظور اینکه مدل قدرت کافی جهت انتقال ویژگی‌های ورودی را به یک سطح بالاتر ویژگی‌ها داشته باشد، حداقل یک انتقال خطی⁹ قابل آموزش نیاز است. بدین منظور به عنوان یک گام اولیه، از یک انتقال خطی اشتراکی که با ماتریس وزن $W \in \mathbb{R}^{F' \times F}$ پارامتری شده است بر روی هر کدام از رئوس اعمال می‌شود.

هم‌اکنون می‌توان یک مکانیزم اشتراکی خود-توجه $a: \mathbb{R}^{F'} \times \mathbb{R}^{F'} \rightarrow \mathbb{R}$ را برای محاسبه ضرایب¹⁰ توجه بر روی رئوس به کار گرفت:

$$e_{ij} = a(W\vec{h}_i, W\vec{h}_j)$$

که e_{ij} به معنی اهمیت ویژگی‌های راس j بر روی راس i است.

در حالت عمومی، هر راس می‌تواند بر روی تمام رئوس دیگر گراف توجه نشان دهد و از اطلاعات کل ساختار گراف استفاده کند، اما در این مقاله از رویکرد توجه ماسک شده استفاده شده است، که e_{ij} را فقط برای رئوس $j \in \mathcal{N}_i$ محاسبه می‌کند و در آن \mathcal{N}_i به معنای همسایه‌های راس i در گراف است. در آزمایش‌های این مقاله، دقیقاً از همسایه‌های مرتبه-اول (رئوسی که مستقیماً با یک یال به راس i متصل هستند) استفاده شده است.

به جهت اینکه ضرایب بر روی رئوس مختلف قابل مقایسه باشند، به کمک یک تابع *Softmax* بر روی تمام انتخاب‌های ممکن j نرمال می‌شوند:

$$\alpha_{ij} = \text{softmax}_j(e_{ij}) = \frac{\exp(e_{ij})}{\sum_{k \in \mathcal{N}_i} \exp(e_{ik})}$$

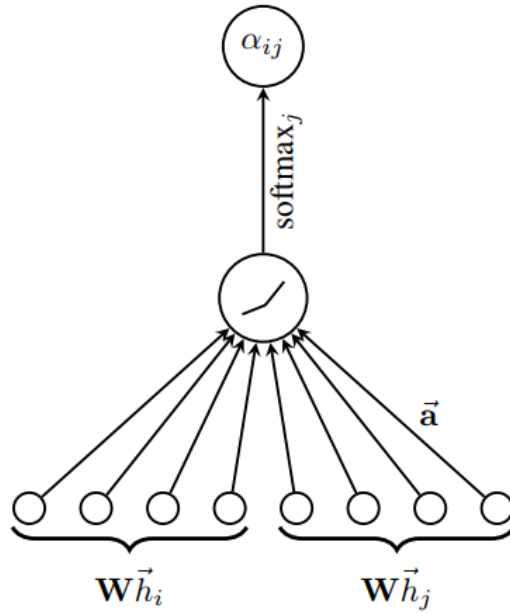
در آزمایش‌های این مقاله مکانیزم توجه a یک لایه شبکه عصبی رو به جلو در نظر گرفته شده که با بردار وزن $\vec{a} \in \mathbb{R}^{2F'}$ پارامتری شده و یک تابع غیرخطی *LeakyReLU* (با ضریب $\alpha = 0.2$ برای ورودی منفی) نیز روی آن اعمال می‌شود. در نتیجه خواهیم داشت:

$$\alpha_{ij} = \frac{\exp(\text{LeakyReLU}(\vec{a}^T [W\vec{h}_i \parallel W\vec{h}_j]))}{\sum_{k \in \mathcal{N}_i} \exp(\text{LeakyReLU}(\vec{a}^T [W\vec{h}_i \parallel W\vec{h}_k]))}$$

رابطه بالا در شکل ۱.۱ بصورت بصری نمایش داده شده است:

⁹ Linear transformation

¹⁰ Coefficients



شکل ۱.۱: مکانیزم توجه $a(W\vec{h}_i, W\vec{h}_j)$

پس از محاسبه ضرایب توجه نرمال شده می‌توان یک ترکیب خطی از ویژگی‌های متناظر با آن‌ها را محاسبه کرده و (پس از اعمال یک غیرخطی‌ساز) به عنوان خروجی نهایی ویژگی هر راس ارائه داد.

$$\vec{h}'_i = \sigma\left(\sum_{j \in \mathcal{N}_i} \alpha_{ij} W\vec{h}_j\right)$$

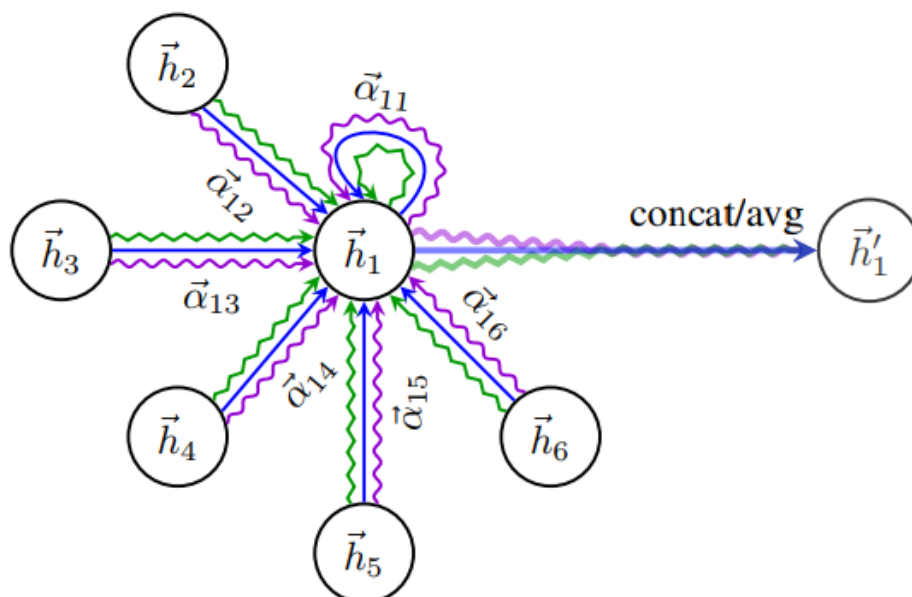
همچنین به منظور تثبیت فرآیند یادگیری خود-توجه، این مکانیزم می‌تواند از توجه *multi-head* نیز بهره ببرد. بدین منظور می‌توان K مکانیزم توجه مستقل از یکدیگر تعریف کرده که هر کدام طبق روابط معرفی شده در بالا یک خروجی نهایی ویژگی تولید کنند. سپس می‌توان خروجی نهایی هر کدام از آن‌ها را با یکدیگر الحاق کرد:

$$\vec{h}'_i = \parallel_{k=1}^K \left(\sigma\left(\sum_{j \in \mathcal{N}_i} \alpha_{ij}^k W^k \vec{h}_j\right) \right)$$

لازم به ذکر است که خروجی نهایی h' شامل KF' ویژگی (بجای F') خواهد بود. بنابراین در صورتیکه از روش *multi-head* بر روی لایه نهایی (پیش‌بینی) استفاده کنیم دیگر نمی‌توان از تکنیک الحاق بردارها استفاده کرد و پیشنهاد می‌شود که از میانگین‌گیری و سپس یک تابع غیرخطی استفاده شود:

$$\vec{h}'_i = \sigma\left(\frac{1}{K} \sum_{k=1}^K \sum_{j \in \mathcal{N}_i} \alpha_{ij}^k W^k \vec{h}_j\right)$$

فرآیند تجميع هر کدام از خروجی‌ها در یک لایه توجه گرافي *multi-head* در شکل ۱.۲ نمایش داده شده است.



شکل ۱.۲: یک نمایش بصری از توجه *multi-head* با $K=3$ بر روی راس شماره ۱

۱.۲ تحلیل و مقایسه مدل

لایه معرفی شده در این مقاله می‌تواند مستقیماً چندین معضل موجود در شبکه‌های عصبی پیشین که با داده‌های گرافي کار می‌کردند را رفع نماید. اکنون فهرست‌وار به این موارد که در واقع مزایای شبکه گرافي توجه هستند می‌پردازیم:

▪ بسیار کارآمد از نظر محاسباتی:

- عمل‌های مورد استفاده در روابط لایه خود-توجه می‌توانند بر روی تمام یال‌ها موازی‌سازی شوند.
- محاسبات ویژگی‌های نهایی می‌تواند بر روی تمام رئوس موازی‌سازی شود.
- نیاز به محاسبات سنگین و هزینه‌بر ماتریسی خصوصاً تجزیه‌ویژه یا تجزیه طیفی^{۱۱} نیست.
- پیچیدگی زمانی مدل برابر است با $O(|V|FF' + |E|F')$.
- استفاده از رویکرد توجه *multi-head* باعث می‌شود که حافظه و پارامترها با ضریب K افزایش یابند.

¹¹ Eigendecomposition

▪ جهشی در ظرفیت یادگیری مدل:

○ برخلاف GCN بصورت ضمنی اهمیت‌های مختلفی به رئوس یک همسایگی مشترک داده می‌شود.

○ با تحلیل وزن‌های توجه یادگرفته شده می‌توان به تفسیری از نحوه یادگیری مدل دست یافت.

▪ بدون وابستگی به ساختار کل گراف:

○ مکانیزم توجه بصورت اشتراکی بر روی تمام یال‌های گراف اعمال می‌شود و بنابراین به ساختار کل گراف یا ویژگی تمام رئوس وابسته نخواهد بود.

○ نیازی نیست که گراف لزوماً بدون جهت باشد (کافیست در صورت عدم وجود یال $j \rightarrow i$ ، ضریب توجه α_{ij} محاسبه نشود).

○ امکان استفاده از این تکنیک بصورت مستقیم بر روی مسائل یادگیری استنتاجی نیز وجود خواهد داشت. در چنین مسائلی مدل بر روی گراف‌هایی مورد ارزیابی قرار می‌گیرد که تاکنون به هیچ‌وجه آن‌ها را در طول آموزش خود مشاهده نکرده است.

▪ در نظر گرفتن همسایگی کامل:

○ در روش‌های پیشین معرفی شده یا یک اندازه همسایگی ثابت در نظر گرفته می‌شود که در اینصورت به راس اجازه داده نمیشد تا از اطلاعات تمام همسایه‌های خود استفاده کند، یا اینکه از تجمیع‌کننده‌های همسایه‌ای بر پایه LSTM استفاده شده است که در اینصورت نیز فرض بر این بوده که لزوماً یک ترتیبی بین رئوس گراف وجود داشته است.

○ اما GAT اصلاً با چنین مشکل‌هایی مواجه نیست. بر پایه همسایگی کامل عمل کرده و همچنین از پیش هیچ ترتیبی برای آن‌ها فرض نمی‌کند.

▪ بازنویسی فرمول به شکل MoNet:

○ روابط لایه GAT را همچنین می‌توان به شکل یک نمونه مشخص از معماری MoNet نیز بازنویسی کرد.

○ با این حال، لایه GAT از ویژگی رئوس جهت محاسبات مشابهت استفاده می‌کند (برخلاف استفاده از خصوصیات ساختاری رئوس در MoNet که نیاز به اطلاعات کامل گراف دارد).

برخی مشکلاتی که در مقاله به آن‌ها اشاره شده نیز عبارتند از:

▪ **محدودیت در امکان دسته کردن^{۱۲}:**

- ابزاری که در این مقاله برای ضرب تانسورها از آن استفاده شده تنها ضرب ماتریس‌های تُنک^{۱۳} تا تانسور درجه-۲ را پشتیبانی می‌کند.

▪ **مزیت پایین GPU:**

- با توجه به نظم ساختارهای گراف مورد استفاده و خصوصاً در سناریوی ماتریس‌های تُنک ممکن است استفاده از GPU مزیت‌های بیشتری را نسبت به CPU برای ما فراهم نکند.

▪ **اندازه Receptive field:**

- کران بالای اندازه receptive field در این معماری برابر است با عمق شبکه (همچون مدل‌های پیشین مانند GCN).
- می‌توان از تکنیک‌هایی نظیر skip connections جهت بسط عمق شبکه استفاده کرد.

▪ **افزونگی محاسبات:**

- در صورت موازی‌سازی بر روی یال‌های گراف، خصوصاً به شکل توزیع‌شده، از آنجا که بسیاری از همسایگی‌ها با همدیگر هم‌پوشانی دارند ممکن است محاسبات زائد زیادی رخ دهد.

۱.۳ مجموعه داده‌ها

چهار مجموعه داده مختلف در این مقاله مورد استفاده قرار گرفته‌اند که سه مجموعه جهت یادگیری Transductive و یک مجموعه دیگر جهت یادگیری Inductive منظور شده است. جزئیات این مجموعه داده‌ها در جدول ۱.۱ آمده است:

	Cora	Citeseer	Pubmed	PPI
Task	Transductive	Transductive	Transductive	Inductive
# Nodes	2708 (1 graph)	3327 (1 graph)	19717 (1 graph)	56944 (24 graphs)
# Edges	5429	4732	44338	818716
# Features/Node	1433	3703	500	50
# Classes	7	6	3	121 (multilabel)
# Training Nodes	140	120	60	44906 (20 graphs)
# Validation Nodes	500	500	500	6514 (2 graphs)
# Test Nodes	1000	1000	1000	5524 (2 graphs)

جدول ۱.۱: خلاصه‌ای از مجموعه داده‌های مورد آزمایش

¹² Batching capabilities

¹³ Sparse matrix multiplication

در سه مجموعه داده نخست (Cora، Pubmed و Citeseer) جزئیات زیر برقرار است:

- رئوس = اسناد (متون)
- یال‌ها (بدون جهت) = ارجاعات
- ویژگی رئوس = المان‌های بازنمایی bag-of-words
- هر راس دارای یک برچسب کلاس است.
- تنها ۲۰ راس به ازای هر کلاس در طول آموزش مورد استفاده قرار گرفته است.
- به دلیل استفاده از رویکرد یادگیری Transductive، الگوریتم یادگیری به بردارهای ویژگی تمام رئوس دسترسی دارد.

برای مجموعه داده Protein-Protein Interaction (PPI) نیز می‌توان به جزئیات زیر اشاره کرد:

- گراف‌ها = بافت‌های^{۱۴} انسان‌های مختلف
- ۲۰ گراف برای آموزش، ۲ گراف اعتبارسنجی و ۲ گراف آزمون
- گراف‌های آزمون کاملاً در طول آموزش نادیده گرفته می‌شوند.
- به منظور ساخت گراف از داده‌های پیش‌پردازش شده استفاده شده است.
- ویژگی رئوس = ۵۰ ویژگی بیولوژی مختلف
- ۱۲۱ برچسب برای هر راس طبق آنتولوژی ژن در نظر گرفته شده است.
- یک راس می‌توان بصورت همزمان شامل چندین برچسب باشد.

۱.۴ نتایج

نتایج بدست آمده توسط شبکه گرافی توجه بر روی مجموعه داده‌های معرفی شده در جداول ۱.۲ و ۱.۳ آمده است. مواردی که با علامت * مشخص شده‌اند بدین معنی است که در تعداد لایه‌ها یا پارامترها در مقایسه با مقالات اصلی معرفی‌کننده آن‌ها تغییری داده شده است تا کاملاً عادلانه و معادل با ساختار GAT مورد استفاده باشند. همچنین منظور از Const-GAT این است که تمامی ضرایب توجه ثابت و برابر با ۱ در نظر گرفته شده‌اند. همانطور که از نتایج مشاهده می‌شود مدل‌های بر پایه GAT توانسته‌اند بهترین نتایج را نسبت به مدل‌های پیشین کسب کنند. حتی مدل Const-GAT با ضرایب توجه ثابت نیز توانسته در مسئله Inductive به نتایج بسیار بهتری نسبت به مدل‌های پیشین دست یابد.

¹⁴ Tissues

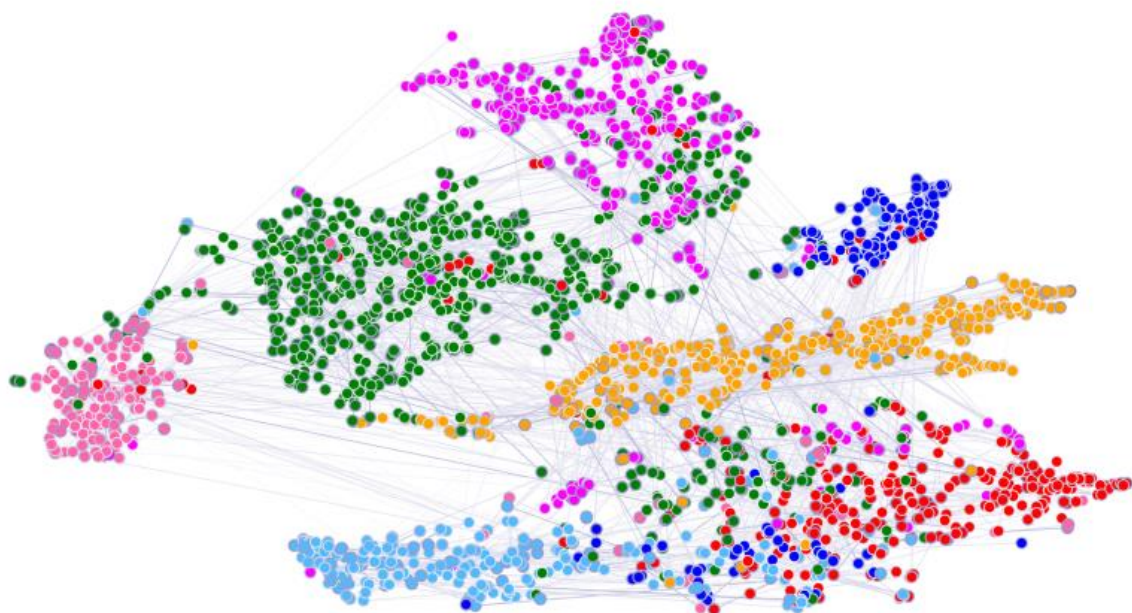
<i>Transductive</i>			
Method	Cora	Citeseer	Pubmed
MLP	55.1%	46.5%	71.4%
ManiReg (Belkin et al., 2006)	59.5%	60.1%	70.7%
SemiEmb (Weston et al., 2012)	59.0%	59.6%	71.7%
LP (Zhu et al., 2003)	68.0%	45.3%	63.0%
DeepWalk (Perozzi et al., 2014)	67.2%	43.2%	65.3%
ICA (Lu & Getoor, 2003)	75.1%	69.1%	73.9%
Planetoid (Yang et al., 2016)	75.7%	64.7%	77.2%
Chebyshev (Defferrard et al., 2016)	81.2%	69.8%	74.4%
GCN (Kipf & Welling, 2017)	81.5%	70.3%	79.0%
MoNet (Monti et al., 2016)	81.7 \pm 0.5%	—	78.8 \pm 0.3%
GCN-64*	81.4 \pm 0.5%	70.9 \pm 0.5%	79.0 \pm 0.3%
GAT (ours)	83.0 \pm 0.7%	72.5 \pm 0.7%	79.0 \pm 0.3%

جدول ۱.۲: نتایج بدست آمده بر روی مسئله Transductive

<i>Inductive</i>	
Method	PPI
Random	0.396
MLP	0.422
GraphSAGE-GCN (Hamilton et al., 2017)	0.500
GraphSAGE-mean (Hamilton et al., 2017)	0.598
GraphSAGE-LSTM (Hamilton et al., 2017)	0.612
GraphSAGE-pool (Hamilton et al., 2017)	0.600
GraphSAGE*	0.768
Const-GAT (ours)	0.934 \pm 0.006
GAT (ours)	0.973 \pm 0.002

جدول ۱.۳: نتایج بدست آمده بر روی مسائل Inductive

در شکل ۱.۳ می‌توان عملکرد نهایی مدل را که به کمک تکنیک t-SNE بصری شده است مشاهده کرد.

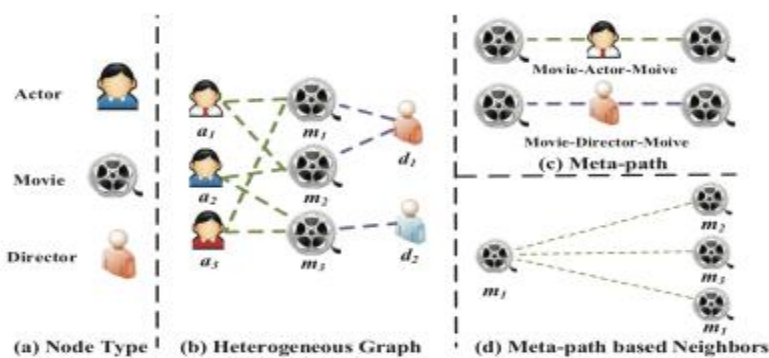


شکل ۱.۳: بازنمایی‌های ویژگی محاسبه شده توسط یک مدل GAT از پیش آموزش دیده بر روی لایه مخفی اول در مجموعه داده Cora. رنگ رئوس نمایانگر برچسب (کلاس) آن‌ها و ضخامت یال‌ها بیانگر ضرایب توجه هستند.

(۲) شبکه توجه گرافی نامتجانس^{۱۵}

داده های دنیای واقعی معمولاً ساختار گرافی دارند، مانند شبکه های اجتماعی، شبکه های استنادی و شبکه جهانی وب (وب جهان گستر). شبکه عصبی گرافی (GNN)، به عنوان یک روش یادگیری بازنمایی عمیق قدرتمند برای چنین داده های گرافی، عملکرد فوق العاده ای از خود نشان داده است. اخیراً، یکی از شگفت انگیزترین پیشرفتها در یادگیری عمیق، مکانیزم توجه است که با داده هایی با اندازه متغیر سروکار دارد و مدل را به تمرکز بر مهمترین قسمت های داده تشویق می کند. شبکه توجه گرافی (GAT)، یک شبکه عصبی گرافی جدید مبتنی بر روش کانولوشن، از مکانیزم توجه برای گراف متجانس استفاده می کند که فقط شامل یک نوع رأس یا یال است. با وجود موفقیت مکانیزم توجه در یادگیری عمیق، در چارچوب شبکه عصبی گرافی برای گراف نامتجانس به طور کامل در نظر گرفته نشده است.

نامتجانس بودن یعنی داشتن گره ها و یال های متنوع. به عنوان مثال، انواع مختلف گره ها صفات متفاوتی دارند و ممکن است ویژگی های آنها در فضای ویژگی متفاوتی بگیرند. برای مثال در مجموعه داده ی فیلم IMDB ویژگی بازیگر ممکن است شامل جنس، سن و ملیت باشد. از طرف دیگر، ویژگی فیلم ممکن است شامل طرح (plot) و بازیگران باشد. یک گراف نامتجانس، که با $G = (V, E)$ مشخص می شود، از یک مجموعه شی V و یک مجموعه یال E تشکیل شده است. یک گراف نامتجانس همچنین با یک تابع نگاشت نوع گره: $\phi: V \rightarrow A$ و یک تابع نگاشت نوع یال $\psi: E \rightarrow R$ مرتبط است. A و R مجموعه ای از انواع شی و انواع یال از پیش تعریف شده را نشان می دهد، بطوری که $|A| + |R| > 2$.



شکل ۲.۱: مثالی از گراف نامتجانس (IMDB). (a) سه نوع گره (یعنی، بازیگر، فیلم، کارگردان). (b) گراف نامتجانس IMDB شامل سه نوع گره و دو نوع اتصال است. (c) دو فرا-مسیر درگیر در IMDB (به عنوان مثال فیلم-بازیگر-فیلم و فیلم-کارگردان-فیلم). (d) فیلم $m1$ و همسایگان متا-مسیر آن (به عنوان مثال، $m1$ ، $m2$ و $m3$).

¹⁵ Heterogeneous Graph Attention Network (HAN)

در این مقاله یک شبکه عصبی گرافی نامتجانس مبتنی بر توجه سلسله مراتبی شامل توجه در سطح گره و توجه در سطح معنایی را ارائه می‌دهیم. قبل از پرداختن به این دو سطح از توجه مفهوم فرا-مسیر (meta-path) را تعریف می‌کنیم. فرا-مسیر یک رابطه ترکیبی است که دو شی را به هم متصل می‌کند:

فرا-مسیر Φ به عنوان یک مسیر به شکل $A_1 \xrightarrow{R_1} A_2 \xrightarrow{R_2} \dots \xrightarrow{R_l} A_{l+1}$ (به اختصار $A_1 A_2 \dots A_{l+1}$)، که یک رابطه ترکیبی $R = R_1 \circ R_2 \circ \dots \circ R_l$ بین اشیا A_1 و A_{l+1} را توصیف می‌کند، جایی که \circ عملگر ترکیب را بر روابط نشان می‌دهد. فرا-مسیر ساختاری است که به طور گسترده‌ای برای ثبت معناها استفاده می‌شود. برای مثال داده‌های فیلم IMDB که در شکل ۲.۱ (a) نشان داده شده است، شامل سه نوع گره فیلم، بازیگر و کارگردان است. رابطه‌ی بین دو فیلم را می‌توان با فرا-مسیر فیلم-بازیگر-فیلم (MAM) نشان داد که رابطه بازیگر مشترک بودن در دو فیلم را توصیف می‌کند، در حالی که فیلم-کارگردان-فیلم (MDM) به معنای کارگردانی آنها توسط همان کارگردان است.

با توجه به فرا-مسیر Φ ، مجموعه‌ای از همسایگان مبتنی بر فرا-مسیر هر گره وجود دارد که می‌تواند اطلاعات مختلف ساختار و معانی غنی را در یک گراف نامتجانس نشان دهد.

تعریف همسایگان مبتنی بر فرا-مسیر. با توجه به گره i و فرا-مسیر Φ در گراف نامتجانس، همسایگان مبتنی بر فرا-مسیر گره i (N_i^Φ) به عنوان مجموعه‌ای از گره‌ها تعریف می‌شوند که از طریق فرا-مسیر Φ به گره i متصل می‌شوند. توجه داشته باشید که همسایگان گره خود گره را نیز شامل می‌شود.

در شکل ۲.۱ (d) با توجه به فرا-مسیر فیلم-بازیگر-فیلم، همسایگان مبتنی بر فرا-مسیر $m1$ شامل (خود) $m1$ ، $m2$ و $m3$ هست. به طور مشابه، همسایگان $m1$ بر اساس فرا-مسیر فیلم-کارگردان-فیلم شامل $m1$ و $m2$ است. ما می‌توانیم که همسایگان مبتنی بر فرا-مسیر را با ضرب دنباله‌ای از ماتریس‌های مجاورت بدست آوریم.

هدف توجه در سطح گره یادگیری اهمیت بین گره و همسایگان مبتنی بر فرا-مسیر آن است، در حالی که توجه در سطح معنایی قادر به یادگیری اهمیت فرا-مسیرهای مختلف است.

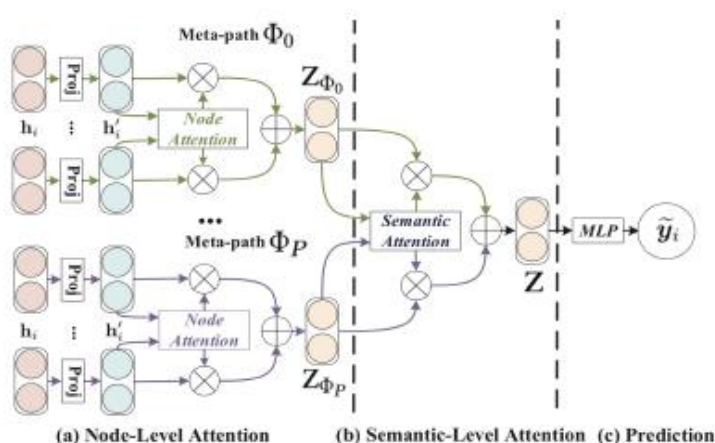
توجه در سطح معنایی: اطلاعات معنایی معنی‌دار و پیچیده مختلفی در گراف نامتجانس دخیل هستند که معمولاً توسط فرا-مسیرها منعکس می‌شوند. هدف توجه در سطح معنایی آن است که اهمیت هر فرا-مسیر را بیاموزد و وزن‌های مناسبی را به آنها اختصاص دهد. برای مثال مجموعه داده‌ی IMDB را در نظر می‌گیریم، The Terminator یا می‌تواند از طریق فیلم-بازیگر-فیلم (MAM) به The Terminator 2 متصل شود (در هر دو شوارتزنگر بازیگر است) یا از طریق فیلم-سال ساخت-فیلم (MYM) به Birdy متصل شود (هر دو در سال

۱۹۸۴ فیلمبرداری شده‌اند). با این حال ، هنگام شناسایی ژانر فیلم The Terminator، MAM معمولاً نقش مهم‌تری دارد تا MYM. بنابراین ، برخورد یکسان با فرا-مسیرهای مختلف سودمند نیست و اطلاعات معنایی ارائه شده توسط برخی از فرا-مسیرهای مفید را ضعیف می‌کند.

توجه در سطح گره: در یک گراف نامتجانس ، گره‌ها را می‌توانند از طریق انواع مختلفی از رابطه ، یعنی، فرا-مسیر متصل باشند. با توجه به یک فرا-مسیر ، هر گره همسایگان مبتنی بر فرا-مسیر زیادی دارد. تشخیص تفاوت ظریف این همسایگان و انتخاب برخی همسایگان حاوی اطلاعات سودمند الزامی است. برای هر گره هدف توجه در سطح گره ، یادگیری اهمیت همسایگان مبتنی بر فرا-مسیر و اختصاص دادن مقادیر مختلف توجه به آنها است. به عنوان مثال در مجموعه داده‌ی IMDB می‌گیریم ، هنگام استفاده از فرا-مسیر فیلم-کارگردان-فیلم (Movie-Director-Movie) (فیلم‌ها با یک کارگردان مشترک) ، The Terminator از طریق کارگردان James Cameron به Titanic و The Terminator 2 متصل می‌شود. برای شناسایی بهتر ژانر The Terminator به عنوان فیلم علمی-تخیلی ، مدل باید بیش از Titanic به The Terminator 2 توجه کند.

(۲.۱) مدل پیشنهادی

در این بخش ، ما یک شبکه عصبی گرافی نیمه نظارت شده (semi-supervised) جدید را برای گراف نامتجانس پیشنهاد می‌کنیم. مدل ما از یک ساختار توجه سلسله مراتبی پیروی می‌کند: توجه در سطح گره \leftarrow توجه در سطح معنایی. شکل ۲.۲ چارچوب کلی HAN را نشان می‌دهد.



شکل ۲.۲: چارچوب کلی HAN پیشنهادی. (a) انواع گره‌ها به یک فضای مشخصه واحد منتقل می‌شوند و وزن جفت گره مبتنی بر فرا-مسیر را می‌توان از طریق توجه در سطح گره یاد گرفت. (b) یادگیری مشترک وزن هر فرا-مسیر و ترکیب کردن تعبیه گره (Node Embedding) خاص معنایی (semantic-specific) از طریق توجه در سطح معنایی. (c) محاسبه ضرر و بهینه سازی انتها به انتها برای HAN پیشنهادی.

در ابتدا، ما توجه در سطح گره را برای یادگیری وزن همسایگان مبتنی بر فرا-مسیر و جمع‌آوری آنها برای بدست آوردن تعبیه گره خاص معنایی را پیشنهاد می‌کنیم. پس از آن، HAN می‌تواند از طریق توجه در سطح معنایی، تفاوت فرا-مسیرها را تشخیص دهد و از ترکیب بهینه وزن دار تعبیه گره خاص معنایی برای کار مشخص، استفاده کند.

۲.۱.۱ محاسبه توجه در سطح گره

توجه در سطح گره می‌تواند اهمیت همسایگان مبتنی بر فرا-مسیر را برای هر گره در یک گراف نامتجانس بیاموزد و بازنمایی این همسایگان معنی دار را برای تشکیل تعبیه رأس جمع کند.

به دلیل نامتجانس بودن گره‌ها، گره‌های مختلف دارای فضای ویژگی متفاوتی هستند. بنابراین برای هر نوع گره (مثلاً گره با نوع ϕ_i) ماتریس تبدیل (Transformation Matrix) وابسته به نوع M_{ϕ_i} (type-specific) را طراحی می‌کنیم تا ویژگی گره‌های متفاوت را به یک فضای ویژگی یکسان تصویر کند. فرایند تصویر کردن می‌تواند به صورت زیر باشد:

$$\mathbf{h}'_i = \mathbf{M}_{\phi_i} \cdot \mathbf{h}_i,$$

که \mathbf{h}_i و \mathbf{h}'_i به ترتیب ویژگی اصلی و تصویر شده گره i هستند. با عمل تصویر کردن وابسته به نوع، توجه در سطح گره می‌تواند گره با نوع دلخواه را کنترل کند. پس از آن، ما از توجه به خود (self-attention) استفاده می‌کنیم تا وزن بین انواع گره‌ها را یاد بگیریم. اگر جفت گره (i, j) که از طریق فرا-مسیر Φ به هم وصل هستند را داشته باشیم، توجه در سطح گره e_{ij}^Φ می‌تواند اهمیت e_{ij}^Φ را بیاموزد، این به این معنی است که گره j برای گره i چقدر مهم خواهد بود. اهمیت جفت گره (i, j) مبتنی بر فرا-مسیر را می‌توان به صورت زیر فرموله کرد:

$$e_{ij}^\Phi = \text{att}_{\text{node}}(\mathbf{h}'_i, \mathbf{h}'_j; \Phi).$$

در اینجا att_{node} نشان دهنده شبکه‌ی عصبی عمیقی است که توجه در سطح گره را انجام می‌دهد. با توجه به فرا-مسیر Φ ، att_{node} برای همه‌ی جفت گره‌های مبتنی بر فرا-مسیر به اشتراک گذاشته می‌شود. این به این دلیل است که برخی الگوهای اتصال مشابه در زیر یک فرا-مسیر وجود دارد. معادله فوق نشان می‌دهد که با توجه به فرا-مسیر Φ ، وزن جفت گره (i, j) مبتنی بر فرا-مسیر به ویژگی‌های آنها بستگی دارد. لطفاً توجه داشته باشید که e_{ij}^Φ نامتقارن است، یعنی، اهمیت گره i برای گره j و اهمیت گره j برای گره i می‌تواند کاملاً متفاوت باشد. این نشان می‌دهد توجه در سطح گره می‌تواند نامتقارن بودن که از ویژگی‌های مهم گراف نامتجانس است را حفظ کند. سپس اطلاعات ساختاری را از طریق توجه پوشیده masked attention به مدل تزریق می‌کنیم، بدین معنی که ما فقط e_{ij}^Φ را برای گره‌های $j \in N_i^\Phi$ محاسبه می‌کنیم، که N_i^Φ نشان‌دهنده همسایگان

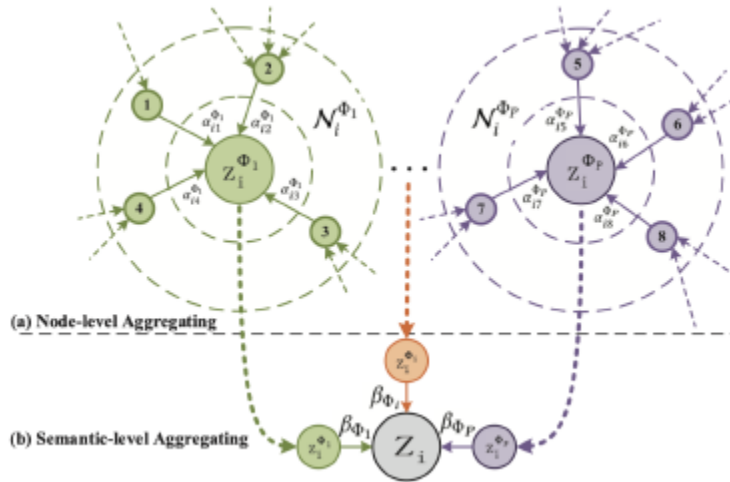
مبتنی بر فرا-مسیر گره i است (شامل خود i). پس از به دست آوردن اهمیت بین جفت گره‌های مبتنی بر فرا-مسیر ، آنها را نرمال می‌کنیم تا ضریب وزن α_{ij}^Φ را از طریق تابع Softmax بدست آوریم:

$$\alpha_{ij}^\Phi = \text{softmax}_j(e_{ij}^\Phi) = \frac{\exp(\sigma(a_\Phi^\top \cdot [h'_i \| h'_j]))}{\sum_{k \in N_i^\Phi} \exp(\sigma(a_\Phi^\top \cdot [h'_i \| h'_k]))}, \quad (3)$$

که در آن σ نشانگر تابع فعال سازی است ، $\|$ نشان دهنده عملیات اتصال می باشد و a_Φ بردار توجه در سطح گره برای فرا-مسیر Φ است. همانطور که از معادله می بینیم. (۳) ، ضریب وزن (i, j) به ویژگی های آنها بستگی دارد. همچنین توجه داشته باشید که ضریب وزن α_{ij}^Φ نامتقارن است به این معنی که آنها کمک متفاوتی به یکدیگر می‌کنند. نه تنها به دلیل ترتیب اتصال در صورت ، بلکه به این دلیل که آنها همسایگان متفاوتی دارند بنابراین ترم نرمال (مخرج) کاملاً متفاوت خواهد بود. سپس ، تعبیه مبتنی بر فرا-مسیر گره i می‌تواند توسط ویژگی‌های تصویر شده همسایه با ضرایب متناظر به شرح زیر جمع شود:

$$\mathbf{z}_i^\Phi = \sigma \left(\sum_{j \in N_i^\Phi} \alpha_{ij}^\Phi \cdot \mathbf{h}'_j \right). \quad (4)$$

که در آن \mathbf{z}_i^Φ تعبیه یاد گرفته شده‌ی گره i برای فرا-مسیر Φ است.



شکل ۲.۳: توضیح فرآیند جمع کردن (انباشتن) در سطح گره و معنایی.

هر تعبیه گره با همسایگانش جمع می‌شود. از آنجایی که وزن توجه α_{ij}^Φ برای یک فرا-مسیر واحد تولید می‌شود ، خاص معنایی است و می‌تواند نوعی از اطلاعات معنایی را ضبط کند. از آنجا که گراف نامتجانس خاصیت بدون مقیاس بودن را دارد ، واریانس (پراکندگی) داده‌های گراف کاملاً زیاد است. برای مقابله با مشکل فوق ، ما توجه

در سطح گره را به توجه چند سره (multi-head) گسترش می‌دهیم تا روند آموزش با ثبات‌تر باشد. به طور خاص، ما توجه در سطح گره را برای K بار تکرار می‌کنیم و تعبیه‌های آموخته شده را به عنوان تعبیه ویژه معنایی به هم متصل می‌کنیم:

$$\mathbf{z}_i^\Phi = \parallel \sigma \left(\sum_{j \in N_i^\Phi} \alpha_{ij}^\Phi \cdot \mathbf{h}_j' \right). \quad (5)$$

با توجه به مجموعه فرا-مسیر $\{\Phi_1, \dots, \Phi_P\}$ ، پس از دادن ویژگی‌های گره به توجه در سطح گره، می‌توانیم P گروه از تعبیه گره‌های خاص معنایی را بدست آوریم، به عنوان $\{Z_{\Phi_1}, \dots, Z_{\Phi_P}\}$ نشان داده شده است.

۲.۱.۲ محاسبه توجه در سطح معنایی

به طور کلی، هر گره در یک گراف نامتجانس شامل انواع مختلفی از اطلاعات معنایی است و تعبیه گره خاص معنایی فقط می‌تواند گره را از یک جنبه منعکس کند. برای یادگیری یک تعبیه گره جامع‌تر، باید چندین معنا را که می‌تواند توسط فرا-مسیرها مشخص شود، ترکیب کنیم. برای پرداختن به چالش انتخاب فرا-مسیر و ترکیب معنایی در یک گراف نامتجانس، ما توجه جدیدی در سطح معنایی را پیشنهاد می‌کنیم تا به طور خودکار اهمیت فرا-مسیرهای مختلف را یاد بگیریم و آنها را برای کار خاص ترکیب کنیم. با در نظر گرفتن P گروه تعبیه گره‌های خاص معنایی از توجه در سطح گره به عنوان ورودی، وزن‌های یاد گرفته شده برای هر فرا-مسیر $(\beta_{\Phi_1}, \dots, \beta_{\Phi_P})$ را می‌توان به شرح زیر نشان داد:

$$(\beta_{\Phi_1}, \dots, \beta_{\Phi_P}) = att_{sem}(Z_{\Phi_1}, \dots, Z_{\Phi_P}). \quad (6)$$

در اینجا att_{sem} نشان دهنده شبکه عصبی عمیق است که توجه در سطح معنایی را انجام می‌دهد. این نشان می‌دهد که توجه در سطح معنایی می‌تواند انواع مختلفی از اطلاعات معنایی را در پشت یک گراف نامتجانس ضبط کند. برای یادگیری اهمیت هر فرا-مسیر، ما ابتدا تعبیه خاص معنایی را از طریق یک تبدیل غیرخطی (به عنوان مثال، MLP یک لایه) تبدیل می‌کنیم. سپس ما اهمیت تعبیه خاص معنایی را به عنوان شباهت تعبیه تبدیل شده با یک بردار توجه در سطح معنایی \mathbf{q} اندازه گیری می‌کنیم. علاوه بر این، از اهمیت همه‌ی تعبیه گره‌های خاص معنایی میانگین می‌گیریم، که می‌تواند به عنوان اهمیت هر یک از فرا-مسیرها توضیح داده شود.

اهمیت هر فرا-مسیر، که به عنوان w_{Φ_i} مشخص می‌شود، به شرح زیر نشان داده شده است:

$$w_{\Phi_P} = \frac{1}{|\mathcal{V}|} \sum_{i \in \mathcal{V}} \mathbf{q}^T \cdot \tanh(\mathbf{W} \cdot \mathbf{z}_i^{\Phi_P} + \mathbf{b}), \quad (7)$$

که W ماتریس وزن است ، b بردار بایاس است ، q بردار توجه در سطح معنایی است. برای مقایسه معنی دار ، تمام پارامترهای فوق برای تمام فرا-مسیرها و تعبیه خاص معنا به اشتراک گذاشته می‌شوند. پس از به دست آوردن اهمیت هر فرا-مسیر ، آنها را از تابع Softmax نرمال می‌کنیم. وزن فرا-مسیر Φ_i ، که به عنوان β_{Φ_i} مشخص می‌شود ، می‌تواند با نرمال سازی اهمیت تمام فرا-مسیرهای بالا با استفاده از تابع Softmax بدست آید ،

$$\beta_{\Phi_p} = \frac{\exp(w_{\Phi_p})}{\sum_{p=1}^P \exp(w_{\Phi_p})}, \quad (8)$$

که می‌تواند به عنوان سهم فرا-مسیر Φ_p برای یک کار خاص تفسیر شود. بدیهی است که هرچه β_{Φ_p} بیشتر باشد ، فرا-مسیر Φ_p مهمتر است. برای کارهای متفاوت ، فرا-مسیر Φ_p ممکن است وزن های مختلفی داشته باشد. با وزن های آموخته شده به عنوان ضرایب ، می‌توانیم این تعبیه های خاص معنایی را برای بدست آوردن تعبیه نهایی Z به صورت زیر ادغام کنیم:

$$Z = \sum_{p=1}^P \beta_{\Phi_p} \cdot Z_{\Phi_p}. \quad (9)$$

برای درک بهتر روند جمع در سطح معنایی ، ما همچنین توضیح مختصری در شکل ۲.۳ (b) ارائه می‌دهیم. تعبیه نهایی توسط جمع تمام تعبیه های خاص معنایی می‌شود. سپس می‌توانیم تعبیه نهایی را برای کارهای خاص اعمال کنیم و توابع ضرر مختلفی را طراحی کنیم. برای طبقه بندی گره نیمه نظارت شده ، ما می‌توانیم Cross-Entropy را روی تمام گره های برچسب خورده بین مقدار واقعی و پیش بینی به حداقل برسانیم:

$$L = - \sum_{l \in \mathcal{Y}_L} Y^l \ln(C \cdot Z^l), \quad (10)$$

که در آن C پارامتر طبقه بند است ، \mathcal{Y}_L مجموعه اندیس های گره های است که دارای برچسب هستند ، Y_l و Z_l برچسب ها و تعبیه های گره های دارای برچسب هستند. با راهنمایی داده های دارای برچسب ، ما می‌توانیم مدل پیشنهادی را از طریق backpropagation بهینه کرده و تعبیه گره ها را یاد بگیریم. روند کلی HAN در الگوریتم ۱ نشان داده شده است.

۲.۲ تجزیه و تحلیل مدل پیشنهادی:

Algorithm 1: The overall process of HAN.

Input : The heterogeneous graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$,
The node feature $\{h_i, \forall i \in \mathcal{V}\}$,
The meta-path set $\{\Phi_0, \Phi_1, \dots, \Phi_P\}$.
The number of attention head K ,
Output : The final embedding Z ,
The node-level attention weight α ,
The semantic-level attention weight β .

```

1 for  $\Phi_i \in \{\Phi_0, \Phi_1, \dots, \Phi_P\}$  do
2   for  $k = 1 \dots K$  do
3     Type-specific transformation  $h'_i \leftarrow M_{\Phi_i} \cdot h_i$ ;
4     for  $i \in \mathcal{V}$  do
5       Find the meta-path based neighbors  $N_i^\Phi$ ;
6       for  $j \in N_i^\Phi$  do
7         Calculate the weight coefficient  $\alpha_{ij}^\Phi$ ;
8       end
9       Calculate the semantic-specific node embedding
10       $z_i^\Phi \leftarrow \sigma\left(\sum_{j \in N_i^\Phi} \alpha_{ij}^\Phi \cdot h'_j\right)$ ;
11    end
12    Concatenate the learned embeddings from all
13    attention head  $z_i^\Phi \leftarrow \big\|_{k=1}^K \sigma\left(\sum_{j \in N_i^\Phi} \alpha_{ij}^\Phi \cdot h'_j\right)$ ;
14  end
15  Calculate the weight of meta-path  $\beta_{\Phi_i}$ ;
16  Fuse the semantic-specific embedding
17   $Z \leftarrow \sum_{i=1}^P \beta_{\Phi_i} \cdot Z_{\Phi_i}$ ;
18 end
19 Calculate Cross-Entropy  $L = -\sum_{l \in \mathcal{Y}_L} Y_l \ln(C \cdot Z_l)$ ;
20 Back propagation and update parameters in HAN;
21 return  $Z, \alpha, \beta$ .
```

- مدل پیشنهادی می‌تواند با انواع مختلفی از گره‌ها و روابط کار کند و معانی غنی موجود در گراف نامتجانس را ترکیب کند. اطلاعات می‌توانند از طریق ارتباط متنوع از یک نوع گره به نوع دیگری از گره منتقل شوند.
- HAN بسیار کارآمد است و میتوان آن را به راحتی به صورت موازی اجرا کرد. محاسبه توجه می‌تواند به صورت جداگانه در تمام گره‌ها و فرا-مسیرها محاسبه شود. پیچیدگی کلی نسبت به تعداد گره‌ها و جفت گره‌های فرا-مسیر خطی است.
- توجه سلسله مراتبی برای کل گراف نامتجانس مشترک است، به این معنی که تعداد پارامترها به مقیاس گراف نامتجانس وابسته نیست و مدل می‌تواند برای گره‌های دیده نشده قبلی یا حتی گراف دیده نشده تعبیه گره ایجاد کند.

- مدل پیشنهادی دارای قابلیت تفسیر بالقوه خوبی برای تعبیه گره یاد گرفته است که یک مزیت بزرگ برای تجزیه و تحلیل گراف نامتجانس است. براساس مقادیر توجه، می‌توانیم بررسی کنیم که کدام گره‌ها یا فرا-مسیرها مشارکت بیشتری (یا کم) برای کار ما دارند، که برای تحلیل و توضیح نتایج ما مفید است.

۲.۳) مجموعه داده‌ها و نتایج

جزئیات مجموعه‌های داده مورد استفاده به شرح زیر است:

نام	منبع داده	داده‌ها	روش برچسب‌گذاری	فرا-مسیرها ^{۱۶}	کلاس‌ها
DBLP	DBLP	۱۴۳۲۸ مقاله (P) ۴۰۵۷ نویسنده (A) ۲۰ کنفرانس (C) ۸۷۸۹ عبارت (T)	برچسب‌گذاری نویسندگان طبق کنفرانس‌هایی که در آنها مقاله ثبت کرده‌اند.	نویسنده-مقاله-نویسنده (APA) نویسنده-مقاله-کنفرانس-مقاله- نویسنده (APCPA) نویسنده-مقاله-عبارت-مقاله- نویسنده (APTPA)	پایگاه داده داده‌کاوی یادگیری ماشین بازیابی اطلاعات
ACM	KDD SIGMOD SIGCOMM MobiCOMM VLDB	۳۰۲۵ مقاله (P) ۵۸۳۵ نویسنده (A) ۵۶ موضوع (S)	برچسب‌گذاری مقالات طبق کنفرانس‌هایی که در آنها چاپ شده‌اند.	مقاله-نویسنده-مقاله (PAP) مقاله-موضوع-مقاله (PSP)	پایگاه داده ارتباطات بی‌سیم داده‌کاوی
IMDB	IMDB	۴۷۸۰ فیلم (M) ۵۸۴۱ بازیگر (A) ۲۲۶۹ کارگردان (D)	برچسب‌گذاری فیلم‌ها طبق ژانر آنها.	فیلم-بازیگر-فیلم (MAM) فیلم-کارگردان-فیلم (MDM)	اکشن کمدی درام

جدول ۱ - جزئیات مجموعه داده‌های مورد استفاده

عملکرد الگوریتم HAN روی مجموعه داده‌های فوق با معیار Micro-F1 و Macro-F1 ارزیابی می‌شود که نتایج آن در کار طبقه‌بندی گره به شرح زیر است:

Datasets	Metrics	Training	DeepWalk	ESim	metapath2vec	HERec	GCN	GAT	HAN _{nd}	HAN _{sem}	HAN
ACM	Macro-F1	20%	77.25	77.32	65.09	66.17	86.81	86.23	88.15	89.04	89.40
		40%	80.47	80.12	69.93	70.89	87.68	87.04	88.41	89.41	89.79
		60%	82.55	82.44	71.47	72.38	88.10	87.56	87.91	90.00	89.51
		80%	84.17	83.00	73.81	73.92	88.29	87.33	88.48	90.17	90.63
	Micro-F1	20%	76.92	76.89	65.00	66.03	86.77	86.01	87.99	88.85	89.22
		40%	79.99	79.70	69.75	70.73	87.64	86.79	88.31	89.27	89.64
		60%	82.11	82.02	71.29	72.24	88.12	87.40	87.68	89.85	89.33
		80%	83.88	82.89	73.69	73.84	88.35	87.11	88.26	89.95	90.54
DBLP	Macro-F1	20%	77.43	91.64	90.16	91.68	90.79	90.97	91.17	92.03	92.24
		40%	81.02	92.04	90.82	92.16	91.48	91.20	91.46	92.08	92.40
		60%	83.67	92.44	91.32	92.80	91.89	90.80	91.78	92.38	92.80
		80%	84.81	92.53	91.89	92.34	92.38	91.73	91.80	92.53	93.08
	Micro-F1	20%	79.37	92.73	91.53	92.69	91.71	91.96	92.05	92.99	93.11
		40%	82.73	93.07	92.03	93.18	92.31	92.16	92.38	93.00	93.30
		60%	85.27	93.39	92.48	93.70	92.62	91.84	92.69	93.31	93.70
		80%	86.26	93.44	92.80	93.27	93.09	92.55	92.69	93.29	93.99
IMDB	Macro-F1	20%	40.72	32.10	41.16	41.65	45.73	49.44	49.78	50.87	50.00
		40%	45.19	31.94	44.22	43.86	48.01	50.64	52.11	50.85	52.71
		60%	48.13	31.68	45.11	46.27	49.15	51.90	51.73	52.09	54.24
		80%	50.35	32.06	45.15	47.64	51.81	52.99	52.66	51.60	54.38
	Micro-F1	20%	46.38	35.28	45.65	45.81	49.78	55.28	54.17	55.01	55.73
		40%	49.99	35.47	48.24	47.59	51.71	55.91	56.39	55.15	57.97
		60%	52.21	35.64	49.09	49.88	52.29	56.44	56.09	56.66	58.32
		80%	54.33	35.59	48.81	50.99	54.61	56.97	56.38	56.49	58.51

HAN_{nd} : این یک نوع HAN است ، که توجه در سطح گره را از بین می برد و به هر همسایه اهمیت یکسانی را می دهد.

HAN_{sem} : این یک نوع HAN است ، که توجه در سطح معنایی را از بین می برد و به هر فرا-مسیر اهمیت یکسانی را می دهد.

HAN: شبکه عصبی نمودار نیمه نظارت شده پیشنهادی که به طور همزمان از توجه در سطح گره و توجه در سطح معنایی استفاده می کند.

نتایج الگوریتم HAN برای کار خوشه کردن گره ها به شرح زیر است:

Datasets	Metrics	DeepWalk	ESim	metapath2vec	HERec	GCN	GAT	HAN_{nd}	HAN_{sem}	HAN
ACM	NMI	41.61	39.14	21.22	40.70	51.40	57.29	60.99	61.05	61.56
	ARI	35.10	34.32	21.00	37.13	53.01	60.43	61.48	59.45	64.39
DBLP	NMI	76.53	66.32	74.30	76.73	75.01	71.50	75.30	77.31	79.12
	ARI	81.35	68.31	78.50	80.98	80.49	77.26	81.46	83.46	84.76
IMDB	NMI	1.45	0.55	1.20	1.20	5.45	8.45	9.16	10.31	10.87
	ARI	2.15	0.10	1.70	1.65	4.40	7.46	7.98	9.51	10.01