یادگیری ساختار شبکه بیزی با استفاده از شبکه عصبی و استخراج قوانین انجمنی

امیرحسین کوچه‌بیوکی [۱]، سید محمود طاهری [۲] و سید مرتضی امینی [۳]

[۱] دانشکده علوم مهندسي، دانشکدگان فني، دانشگاه تهران،
a.koochehbiuki@ut.ac.ir

[۲] دانشکده علوم مهندسي، دانشکدگان فني، دانشگاه تهران،
sm_taheri@ut.ac.ir

[۳]دانشکده ریاضی آمار و علوم کامپیوتر، دانشکدگان علوم، دانشگاه تهران،
morteza.amini@ut.ac.ir

چکیده– در این مقاله به بررسی یادگیری ساختار شبکه بیزی با استفاده از شبکه عصبی به روش DAG-GNN و استخراج قوانین انجمنی از آن می‌پردازیم. از آنجا که روش یادگیری ساختار شبکه بیزی انتخابی روشی تقریبی و از جمله روش‌های یادگیری بدون نظارت قرار می‌گیرد، به منظور درک عملکرد روش، در اجرای اول و دوم، مجموعه‌داده‌هایی را مورد پژوهش قرار دادیم که ساختار حقیقی‌آنها در تولید نمونه‌ها در دسترس باشد. در دو سناریو اجرا عملکرد روش را با داده‌های پیوسته و داده‌های گسسته بررسی می‌کنیم. در اجرای سوم، برای یک مجموعه‌داده تراکنشی که به‌صورت دستی ایجاد شده‌است، روش را اجرا و خروجی را در کنار خروجی الگوریتم Apiori نمایش می‌دهیم و مقایسه می‌کنیم.

کلمات کلیدی– شبکه بیزی، شبکه عصبی کدگذار خودکار تغییراتی، گراف دانش، گراف شبکه عصبی، قوانین انجمنی

۱. مقدمه

الگوریتم‌های کاوش قوانین انجمنی، مستلزم تولید تعداد بسیار زیاد قوانین هستند. حتی با تعداد کم اقلام داده‌ها نیز با حجم وسیعی از قوانین روبرو هستیم. چنانچه فرض کنیم کلیه الگوها مفید هستند، برای کاربر امکان پذیر نیست تا قضاوت مناسبی میان آن‌ها داشته باشد. بدین علت نیاز به الگوریتم‌های موثر جهت محدود نمودن این فضای وسیع و همچنین معیارهایی جهت ارزیابی قوانین انجمنی احساس می‌شود. برای نمونه زمانی که یک سیستم یا محیط پیچیده موردمطالعه قرار می‌گیرد، با پرسش چه اتفاقی برای X، در صورت تغییر فاکتور Y، رخ می‌دهد؟ مواجه هستیم. سیستم پاسخ‌دهی به این‌گونه سؤالات نیازمند رمزگشایی روابط علّیت بین مؤلفه‌های مختلف سیستم، که معمولاً از تجربه یا مشاهدات دادگان است، می‌باشد. یکی از محبوب‌ترین مدل‌های علّیت که به‌عنوان شبکه بیزی (BN) شناخته می‌شود، استقلال مشروط بین متغیرها را با استفاده از گراف‌های جهت‌دار بی‌دور (DAG)، مدل‌بندی می‌کند.

۲. علّیت و شبکه بیزی

به این نکته توجه شود که وابستگی با علّیت متفاوت است. در مثال زیر فرض کنید قصد داریم متغیر Y را با داشتن متغیرهای X و Z پیش‌بینی کنیم، به گونه‌ای که

$$X, E_Y, E_Z \sim Uniform(0,1)$$
$$Y \leftarrow 0.5X + E_Y$$
$$Z \leftarrow Y + E_Z$$

در مثال بالا E_Z و E_Y متغیرهای نوفه جمعی هستند، و Uniform به معنی توزیع یکنواخت است. مدل‌بالا به ما می‌گوید که مقادیر Y تابعی از مقادیر X هستند، یا در حقیقت X علّت Y است. همچنین مقادیر Z تابعی از مقادیر Y هستند، یا Y علّت Z است. بنابراین برای کنترل متغیر Y، تنها نیاز است که X را کنترل کنیم، و تغییر Z بر روی Y تأثیری نخواهد داشت.

بااین‌حال، Z یک نرخ سیگنال–به–نویز قوی‌تر از X برای پیش‌بینی Y فراهم می‌کند و به نظر می‌رسد که پیش‌بینی‌کننده بهتری برای Y نسبت به X است. با این دید، بهترین پاسخ رگرسیون

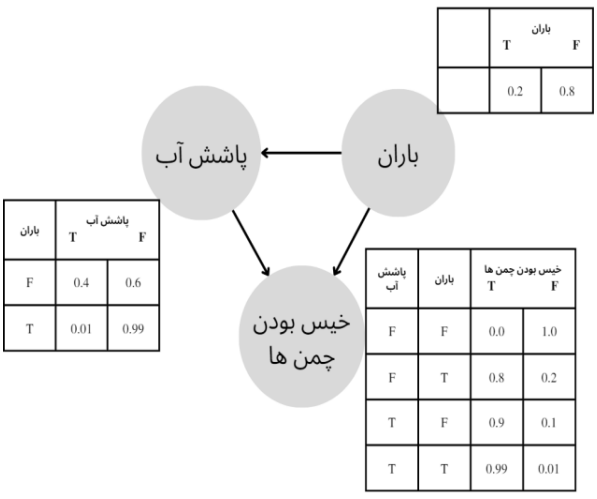
برای Y برپایه ملاک کمترین توان‌های دوم خطا به‌صورت زیر خواهد بود

$$\hat{Y} = 0.25X + 0.5Z$$

در مدل رگرسیون بالا به‌اشتباه این موضوع که تغییرات در Z سبب تغییرات در Y می‌شود، برداشت خواهد شد.

در حقیقت، مدل‌های بر پایه همبستگی[1] منجر به نتیجه‌گیری اشتباه در بحث کنترل متغیرها می‌شوند. عدم تفکیک بین همبستگی و علّیت برای کسانی که قصد دارند تنها با داده‌های مشاهداتی تصمیم‌گیری و برنامه‌ریزی کنند، ممکن است گمراه کننده باشد.

یکی از مدل‌های قدرتمند استنتاج علّی، شبکه‌های بیزی هستند. هدف استنتاج، محاسبه احتمال پسین برای متغیرهای تصادفی شبکه می‌باشد. شبکه‌های بیزی، ترکیبی از نظریه گراف و نظریه احتمال، است. این شبکه‌ها نشان‌دهنده روابط علّی و معلولی میان متغیرها هستند. ساختار گراف یک شبکه بیزی برای صورت‌بندی توزیع احتمال توأم متغیرهای شبکه بکار می‌رود. هنگامی که ساختار گراف معلوم باشد، مدل‌های احتمالاتی می‌توانند برای استدلال و پیش‌بینی دربارهٔ متغیرها بکار روند. چنانچه ساختار گراف نامشخص باشد، با استفاده از این مدل‌ها می‌توان به بهینه‌کردن الگوریتم‌های یادگیری ماشین پرداخت و آنگاه استدلال و پیش‌بینی دربارهٔ متغیرها را انجام داد. برای نمونه شکل۱ شبکه بیزی مربوط به ۳ متغیر و جداول احتمال مشروط مربوط به هرکدام را نشان می‌دهد.



شکل ۱: مثالی از شبکه بیزی

۳. روش پیشنهادی

ازآنجاکه خروجی روش‌های یادگیری علّیت بیشتر گراف‌های جهت‌دار فاقد دور (DAG) هستند، در نظر داریم تا ابتدا مدلی از نوع شبکه بیزی روی داده‌ها، به‌منظور یادگیری ساختار بیزی، آموزش دهیم. در ادامه به‌منظور کشف قوانین به‌جای مراجعه مکرر به داده‌ها به‌منظور محاسبه مقدار پشتیبان و اطمینان، با به‌کارگیری الگوریتم‌های گراف و یا روش‌های استنتاج علّی روی مدل و یادگیری پارامترها، خروجی به شکل قانون را تولید می‌کنیم.

۴. یادگیری ساختار DAG عصبی (DAG-GNN)

با انگیزه گرفتن از موفقیت گسترده یادگیری عمیق و توانایی آن در نگاشت‌های غیرخطی پیچیده، در این روش یک مدل مولد عمیق با اعمال نوعی از محدودیت ساختاری برای یادگیری یک DAG، پیشنهاد شده است. در داخل مدل مولد، یک خود رمزگذار تغییراتی[2] (VAE) که با یک معماری جدید گراف شبکه عصبی پارامتری شده، وجود دارد. این روش را چن و همکاران DAG-GNN نامیده‌اند[۱]. از مزایای این روش، برخورد مناسب با مقادیر گسسته به‌خوبی مقادیر برداری است.

از مزایا و ویژگی‌های روش DAG-GNN می‌توان موارد زیر را نام برد:

- این روش بر مبنای یکی از پرکاربردترین مدل‌های مولد یعنی خود رمزگذار تغییراتی (VAE) ساخته شده است. این مدل قادر به یادگرفتن توزیع‌های پیچیده از داده و نمونه‌برداری از آن‌ها می‌باشد.
- تحت فرضیات گراف، در این روش ماتریس همسایگی وزن‌دار گراف با دیگر پارامترهای شبکه عصبی یاد گرفته می‌شود.
- چارچوب VAE قادر به کارکردن با مقادیر مختلف داده پیوسته و گسسته است.
- به دلیل استفاده از شبکه‌های عصبی گراف برای پارامتری کردن، هر متغیر (گره) نه‌تنها می‌تواند مقادیر اسکالر باشد، همچنین می‌تواند مقادیر برداری باشد. این متغیرها به‌عنوان ویژگی‌های گره در ورودی/خروجی گراف‌های شبکه عصبی (GNN) در نظر گرفته می‌شوند.
- این روش گونه‌ای جدید از محدودیت بی‌دور بودن را پیشنهاد می‌کند که برای پیاده‌سازی در پلتفرم‌های یادگیری عمیق فعلی مناسب است. این محدودیت مرتبه

[1] Correlation-Based Methods

[2] Variatonal autoencoder

چندجمله‌ای دارد که هم در عمل و هم به‌صورت عددی مناسب‌تر است.

این روش، عمل یادگیری ماتریس همسایگی وزن‌دار یک DAG را با استفاده از یک مدل مولد که تعمیم دهنده یک مدل معادله ساختاری (SEM) خطی است را، انجام می‌دهد.

۴-۱- *مدل معادله ساختاری (SEM) خطی*

فرض کنید $A \in \mathbb{R}^{m\times m}$ ماتریسی از یک DAG با m رأس و $X \in \mathbb{R}^{m\times d}$ یک نمونه از یک توزیع توام از m متغیر به‌گونه‌ای که هر سطر مرتبط با یک متغیر باشد، باشند. در مدل SEM خطی رابطه زیر

$$X = XA + Z = A^T X + Z \qquad (۱)$$

به صورتی که $Z \in \mathbb{R}^{m\times d}$ ماتریس نوفه باشد، برقرار است. زمانیکه رأس های گراف به ترتیب توپولوژیکی مرتب شوند، ماتریس A بالا مثلثی می‌شود. از همین رو، نمونه برداری اجدادی[3] از DAG برابر با تولید یک نوفه تصادفی طبق رابطه زیر است

$$X = (I - A^T)^{-1} Z \qquad (۲)$$

۴-۲- *مدل شبکه عصبی گراف پیشنهادی*

رابطه (۲) را می‌توان به فرم $X = f_A(Z)$ نوشت، که خلاصه‌ای از گراف‌های شبکه عصبی پارامتری شده است که ویژگی‌های رأس Z را به عنوان ورودی گرفته و X را به عنوان یک خروجی برمی‌گرداند. تقریبا تمام شبکه‌های عصبی گرافی می‌توانند به این فرم نوشته شوند.

باتوجه‌به ساختار خاص رابطه (۲)، در این روش یک معماری جدید شبکه عصبی

$$X = f_2((I - A^T)^{-1} f_1(Z)) \qquad (۳)$$

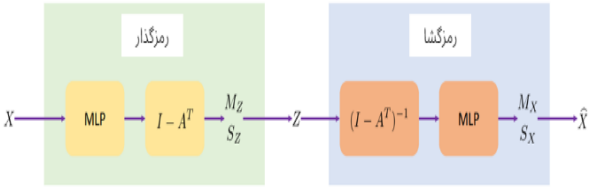
توسط چن و همکاران، ۲۰۱۹ [۱] ارائه شده است. توابع پارامتری شده f_1 و f_2 به ترتیب روی Z و X انتقال‌هایی (احتمالاً غیرخطی) را انجام می‌دهند. اگر f_2 معکوس‌پذیر باشد، آنگاه رابطه (۳) برابر با $f_2^{-1}(X) = A^T f_2^{-1}(X) + f_1(Z)$ که فرم تعمیم یافته SEM خطی رابطه (۱) است، می‌شود[۱].

۴-۳- *معماری و تابع زیان*

برای تکمیل VAE می‌بایست توزیع‌های زیر را مشخص کنیم

$$L_{ELBO} = \frac{1}{n}\sum_{k=1}^{n} L^k_{ELBO},$$
$$L^k_{ELBO} \equiv$$
$$-D_{KL}\left(q(Z \mid X^k) \,\|\, p(Z)\right) +$$
$$E_{q(Z|X^k)}\left[\log p(X^k \mid Z)\right] \qquad (۴)$$

به یاد داشته باشیم که X^k و Z هر دو ماتریس‌های m × d هستند. برای سادگی، توزیع پیشین معمولاً به‌عنوان یک ماتریس نرمال استاندارد $p(Z) = MN_{m\times d}(0, I, I)$ مدل‌بندی می‌شود. شکل ۲ معماری کلی روش را نشان می‌دهد.



شکل۲: معماری(برای متغیرهای پیوسته). برای متغیرهای گسسته خروجی از M_X و S_X به P_X تغییر می‌کند.

برای مدل استنتاج (قسمت رمزگذار)، فرض‌کنید f_3 یک پرسپترون چندلایه[4] (MLP) و f_4 یک نگاشت اصلیت باشند. آنگاه توزیع پسین متغیر $q(Z|X)$، یک توزیع گوسی با میانگین $M_Z \in \mathbb{R}^{m\times d}$ و انحراف معیار $S_Z \in \mathbb{R}^{m\times d}$ هست که توسط رابطه زیر محاسبه می‌شود

$$[M_Z \mid \log S_Z] = (I - A^T)MLP(X, W^1, W^2) \qquad (۶)$$

که در آن $MLP(X, W^1, W^2) := \mathrm{ReLU}(XW^1)W^2$ ، و W^1 و W^2 ماتریس‌های پارمتر هستند.

برای مدل مولد (قسمت رمزگشا)، در نظر بگیریم f_1 یک نگاشت اصلیت و f_2 یک MLP باشند. آنگاه، درستنمایی $p(X|Z)$ یک توزیع گوسی با میانگین $M_X \in \mathbb{R}^{m\times d}$ و انحراف معیار $S_X \in \mathbb{R}^{m\times d}$ هست که توسط رابطه زیر محاسبه می‌شود

$$[M_X \mid \log S_X] = MLP((I - A^T)^{-1}Z, W^3, W^4) \qquad (۷)$$

به صورتی که W^3 و W^4 ماتریس‌های پارامتر باشند.

[3] Ancestral sampling

[4] Multilayer perceptron

بر اساس رابطه (۶) و (۷)، واگرایی کولبک–لایبلر بین توزیع پیشین و پسین رمز که در کران واگرایی (ELBO) استفاده می‌شود، برابر است با:

$$D_{KL}\left(q(Z \mid X) \,||\, p(Z)\right) = \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{d}(S_Z)_{ij}^2 + (M_Z)_{ij}^2 - 2\log(S_Z)_{ij} - 1 \qquad (۸)$$

همچنین دقت بازسازی رابطه ELBO که می‌توان آن را با شبیه‌سازی مونت‌کارلو محاسبه کرد، برابر است با

$$E_{q(Z|X)}\left[\log p(X \mid Z)\right] \approx \frac{1}{L}\sum_{l=1}^{L}\sum_{i=1}^{m}\sum_{j=1}^{d} -\frac{(X_{ij} - (M_X^{(l)})_{ij})^2}{2(S_X^{(l)})_{ij}^2} - \log(S_X^{(l)})_{ij} - c \qquad (۹)$$

به صورتی که c یک ثابت، $M_X^{(l)}$ و $S_X^{(l)}$ خروجی‌های رمزگشا رابطه (۷) با ورودی نمونه‌های شبیه‌سازی شده مونت‌کارلو $Z^{(l)} \sim q(Z \mid X), l = 1, \ldots, L$ هستند.

توجه داشته باشید که تحت چارچوب خودرمزگذار[5]، Z یک متغیر نهان در نظر گرفته می‌شود (به‌جای اینکه یک نویز در SEM خطی در نظر گرفته شود). ازاین‌رو ممکن است ابعاد ستون Z از d متفاوت باشد. از دیدگاه شبکه عصبی، تغییر ابعاد Z تنها در اندازه ماتریس‌های پارامتر W^2 و W^3 تأثیر می‌گذارد. کاربرد این موضوع زمانی است که شخصی مشاهده کند که داده ابعاد ذاتی کمتری دارد، و بخواهد که عدد کوچک‌تر از d استفاده کند [۱].

۴-۴- تعمیم به متغیرهای گسسته

برای مقادیر گسسته، هر سطر از X را به‌صورت یک بردار نشانگر one-hot در نظر گرفته می‌شود. در این بردار همه درایه‌ها به‌جز مقدار مرتبط با متغیر که یک است، صفر هستند. همچنان از ماتریس نرمال استاندارد برای مدل کردن توزیع پیشین و از توزیع گوسی برای مدل‌بندی توزیع پسین متغیر در همان رابطه (۶) قسمت رمزگذار، استفاده می‌کنیم.

برای $p(X|Z)$ ، یک توزیع چندجمله‌ای با ماتریس احتمال P_X به شکلی که هر سطر یک بردار احتمال مقدار مربوطه باشد، در نظر می‌گیریم. برای دست‌یافتن به این موضوع، تابع f_2 از تابع نگاشت به یک تابع سطری softmax تغییر می‌دهیم. رابطه (۷) در قسمت رمزگشا به‌صورت

$$P_X = \text{softmax}\left(MLP((I - A)^{-1}Z, W^3, W^4)\right) \qquad (۱۰)$$

مشخص می‌شود [۱].

۴-۵- ارتباط با SEM خطی

ابتدا به بررسی چگونگی توسعه مدل پیشنهادی از یک SEM خطی می‌پردازیم. در این روش ما یک غیرخطی بودن به روال نمونه‌برداری رابطه (۲) از SEM اعمال کردیم. نتایج مدل مولد را به‌عنوان یک رمزگشا تلقی کرده و با اضافه‌کردن یک رمزگذار متغیر، یک یادگیری قابل‌اجرا فراهم کردیم. در مقایسه با یک کدگذار خودکار ساده، نسخه متغیر به ما اجازه می‌دهد که یک مدل‌سازی فضای نهان که از آن نمونه‌ها تولید می‌شوند، را داشته باشیم[۱].

۴-۶- محدودیت بی‌دور بودن

هیچ‌کدام یک از بیشینه کردن کران پایین واگرایی یا کمینه کردن خطای توان دوم نمی‌تواند بی‌دور بودن گراف نتیجه شده از ماتریس A، را تضمین کند. ژنگ و همکاران [۲] یک محدودیت که بی‌دور بودن را تضمین می‌کند، به تابع زیان اضافه کردند.

این ایده بر اساس این حقیقت است که، مثبت بودن درایه (i,j) امین توان ماتریس همسایگی نامنفی B، نشان‌دهنده یک مسیر به طول k بین رأس‌های i و j ، می‌باشد. ازاین‌رو مثبت بودن قطر B^k بیانگر دورها است. ماتریس نمایی، بسط تیلور (مجموع وزن‌دار توان‌های صحیح نامنفی ماتریس) را می‌پذیرد. ضریب توان صفرم (ماتریس همانی $I_{m\times m}$) ۱ است. ازاین‌رو اثر (trace) نمایی B برای یک DAG باید m باشد. برای نتیجه مطلوب، نامنفی بودن B را مربع عناصر A می‌گذاریم، که $B = A \circ A$ می‌شود.

اگرچه این ایده از لحاظ ریاضی برازنده است، اما ممکن است پشتیبانی از ماتریس نمایی در همه بسترهای یادگیری عمیق مقدور نباشد. ازاین‌رو، یک محدودیت جایگزین که در عمل راحت‌تر است پیشنهاد می‌دهیم.

قضیه ۱. فرض کنید $A \in \mathbb{R}^{m\times m}$ ماتریس همسایگی وزن‌دار یک گراف جهت‌دار باشد. برای هر $\alpha > 0$، گراف بی‌دور است اگر و تنها اگر

$$tr\left[(I - \alpha A \circ A)^m\right] - m = 0 \qquad (۱۳)$$

باشد[۱].

ما از رابطه (۱۳) به‌عنوان یک عبارت محدودیت در زمان بیشینه کردن کران پایین واگرایی استفاده می‌کنیم. محاسبات هر

[5] Autoencoder

دو $(I+\alpha B)^m$ و $\exp(B)$ ممکن است زمانیکه مقدارهای ویژه ماتریس B بزرگ هستند، سختی عددی داشته باشند اما با انتخاب α مناسب محاسبات اولی سختی کمتری به نسبت دومی دارد[۱].

۷-۴- *آموزش*

بر اساس مطالب بیان شده، مسئله یادگیری عبارت است از

$$\min_{A,\theta} \quad f(A,\theta) \equiv -L_{ELBO}$$
$$\text{s.t.} \quad h(A) \equiv tr\left[(I+\alpha A \circ A)^m\right] - m = 0$$

که در آن، مقادیر نامعلوم شامل ماتریس A و تمام پارامترهای θ از VAE ($\theta = \{W^1, W^2, W^3, W^4\}$) هستند. حال از رویکرد لاگرانژ برای حل مسئله غیرخطی محدودیت‌دارمان استفاده می‌کنیم.

تابع لاگرانژ مسئله بالا به‌صورت زیر است

$$L_c(A,\theta,\lambda) = f(A,\theta) + \lambda h(A) + \frac{c}{2}|h(A)|^2$$

که در آن λ ضریب لاگرانژ و c پارامتر جریمه باشد. زمانیکه c = $+\infty$ باشد، کمینه کننده $L_c(A,\theta,\lambda)$ باید در قید $h(A)=0$ صدق کند، که در این صورت $L_c(A,\theta,\lambda)$ برابر با تابع هدف $f(A,\theta)$ می‌شود. از این رو، استراتژی این است که به تدریج c را افزایش دهیم تا زمانیکه تساوی بدون محدودیت کمینه شود. ضریب لاگرانژ λ متقابلا بروز می‌شود، بنابراین تساوی به یک نقطه که در شرایط بهینه قرار دارد همگرا می‌شود.

چند روش برای افزایش c و بروز کردن λ وجود دارد، اما یک روش معمولی موثر به صورت زیر است

$$\left(A^k, \theta^k\right) = \arg\min_{A,\theta} L_{c^k}(A,\theta,\lambda^k), \qquad (۱۴)$$
$$\lambda^{k+1} = \lambda^k + c^k h(A^k), \qquad (۱۵)$$
$$c^{k+1} = \begin{cases} \eta c^k, & \text{if } |h(A^k)| > \gamma |h(A^{k-1})| \\ c^k, & otherwise \end{cases} \qquad (۱۶)$$

که در آن $\eta > 1$ و $\gamma < 1$ پارامترهای تنظیم هستند. دیده شده است که اغلب $\eta = 10$ و $\gamma = 1/4$ کارکرد مناسبی دارد[۱].

۵. استنتاج از مدل یادگرفته شده

ویژگی مارکوف و لایه مارکوف، دو مفهوم مورد استفاده ما برای استنتاج از گراف یادگرفته شده می‌باشند.

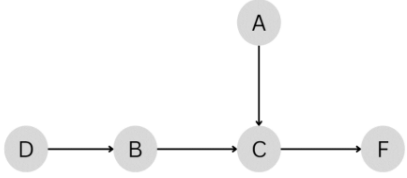
بنابر خاصیت مارکوف در توزیع احتمال شرطی، هر راس فقط به والدین خود وابسته است و هر استقلال با فقدان یک یال در گراف مشخص می‌شود.

منظور از لایه مارکوفی یک متغیر، مجموعه والدهای این متغیر، فرزندانش و والدهای فرزندانش هستند. یک متغیر به طور مشروط مستقل از بقیه متغیرها با توجه به لایه مارکوف آن می‌باشد.

از همین رو با توجه به مفاهیم شبکه بیزی، موارد زیر جهت استخراج دانش از مدل یادگرفته شده، کاربردی می‌باشد:

- در نظر گرفتن گره‌های بدون یال ورودی به عنوان گره‌های مستقل
- در نظر گرفتن لایه مارکوف یک گره به عنوان رأس‌های در ارتباط
- در نظر گرفتن یال‌ها به عنوان علّت مستقیم
- در نظر گرفتن ساختار سر به دم متشکل از سه گره به عنوان علّت غیر مستقیم (درصورت داده شدن گره میانی، گره‌های ابتدا و انتها مستقل هستند)
- درنظر گرفتن یک یال به عنوان قانون با توجه به آستانه ضریب همبستگی در نظر گرفته شده، و نمایش آن در خروجی
- در صورت گسسته بودن داده ورودی، انجام عمل یادگیری جدول احتمال‌های شرطی (CPT) با توجه به ساختار یادگرفته شده و نمایش قانون برای هر مقدار گره با توجه به والدین آن گره و حد آستانه در نظرگرفته شده

یادآوری این نکته که مسئله یادگیری ساختار، مسئله‌های سخت است و جواب بدست آمده تقریبی هست، ما را متوجه می‌کند که دانش بدست آمده از موارد فوق، به تنهایی نمی‌تواند مبنای طراحی سامانه‌های تصمیم‌گیرنده یا فازی باشد و باید آزموده و بررسی شود. خروجی این بخش می‌تواند برای سامانه‌های توصیه‌گر در نظر گرفته شود.



شکل ۳: نمونه DAG

برای نمونه از مدل DAG شکل ۳، قوانین زیر بدست می‌آید:

۱- اگر D آنگاه B با احتمال مشروط $P(B|D)$ $(D \rightarrow B)$

۲- اگر B آنگاه C با احتمال مشروط $P(C|B)$ $(B \rightarrow C)$

۳- اگر C آنگاه F با احتمال مشروط $P(F|C)$ $(C \rightarrow F)$

۴- اگر A آنگاه C با احتمال مشروط $P(C|A)$ $(A \rightarrow C)$

۵- اگر A و B آنگاه C با احتمال مشروط $P(C|A,B)$ $(A , B \rightarrow C)$

۶. نتایج

۶-۱- *کلیات اجرای روش*

مقادیر مشخص شده در زیر برای تمام اجراها تنظیم شده و تنها پارامتر سایز batch با توجه به تعداد نمونه‌ها و عملکرد سیستم اجرا کننده (CPU: i5-10210U 1.60 - 2.11 GHz) برای هر مجموعه داده متفاوت بود.

- تعداد لایه‌های رمزگذار: ۲
- تعداد لایه‌های رمزگشا: ۲
- تعداد رأس‌های هر لایه رمزگذار: ۶۴
- تعداد رأس‌های هر لایه رمزگشا: ۶۴
- ابعاد متغیرپنهان (z): تعداد ویژگی‌های هر نمونه در مجموعه‌داده
- تعداد گام هر اجرا از یک اجرای شبکه عصبی: ۱۵۰
- الگوریتم بهینه‌ساز در اجرای شبکه عصبی و تساوی لاگرانژی: Adam
- کران پایین تغییر تساوی لاگرانژی به منظور اجرای بعدی: اجرای اول و دوم $1e^{-8}$ – اجرای سوم $1e^{-10}$
- مقدار کمینه ضریب همبستگی در ماتریس مجاورت جهت در نظر گرفتن یال در گراف: ۰/۲

مسئله پیدا کردن گراف به یک مسئله بهینه‌سازی تبدیل شده و در نهایت به صورت یک تساوی لاگرانژی نوشته شده است. هنگامی که ما روش را روی یک مجموعه داده اجرا می‌کنیم، روش به دنبال بهینه کردن تساوی می‌رود. هدف از به‌روزرسانی پارامترهای تساوی این است که $L_c(A,\theta,\lambda)$ هیچگونه نقض قیدی ($h(A)$) نداشته باشد و برابر با $f(A,\theta)$ شود. در هرگام از بهینه‌سازی هنگامی که به سراغ محاسبه $f(A,\theta)$ می‌رود، یک اجرا کامل از شبکه عصبی را انجام داده و مقادیر بهترین گامش را به تساوی بر می‌گرداند. آنقدر این کار تکرار می‌شود تا دیگر تساوی لاگرانژی کمتر از حد آستانه تعیین شده خود تغییر نکند.

۶-۲- *نتایج روی داده‌های حقیقی بیولوژیکی (داده‌های گسسته)*

روش DAG-GNN را روی داده‌های مسئله شبکه پروتئینی [۳] اعمال کردیم (مجموعه داده SACHS). از آنجا که هر سلول می‌تواند به عنوان یک مشاهده مستقل درنظرگرفته شود، داده‌های فلوسایتومتری، یک نمونه آماری بزرگ برای شبکه‌های بیزی فراهم می‌کند. بررسی شبکه‌های علامت‌دهی در جمعیت‌های تک‌سلولی‌ها، یک منبع از وابستگی‌های قوی آماری برای علّیت سیگنال‌دهی می‌باشد.

تعداد نمونه‌ها: ۱۰۰۰۰ – تعداد ویژگی‌ها (گره‌ها): ۱۱ – مقادیر ویژگی‌ها: گسسته – تعداد یال‌ها: ۱۷ – میانگین سایز پوشش مارکوفی: ۳/۰۹ – بیشترین درجه ورودی: ۳

نتایج بعد از خروجی گرفتن از روش به شرح زیر می‌باشد:

مقدار کران پایین واگراییی (ELBO) گام مربوط به گراف خروجی در آخرین اجرای شبکه عصبی، ۰/۰۰۰۳۹۲۲۶۴ بوده است. این گراف در زمان ۵۷ دقیقه بدست آمده است.

جدول ۱: **وضعیت یال‌های گراف خروجی روی SACHS**

عدم پیش‌بینی یال	پیش‌بینی یال نادرست	پیش‌بینی یال درست – جهت نادرست	پیش‌بینی یال درست – جهت درست
۵	۲	۴	۸

۶-۳- *نتایج روی داده‌های مصنوعی (داده‌های پیوسته)*

روش DAG-GNN را روی یک مجموعه‌داده مصنوعی نشان‌دهنده یک گراف [۴] اعمال کردیم (مجموعه‌داده G5_v3_numdata). ۵۰۰ نمونه از ۲۰ گراف علیت مصنوعی آموزش و ۲۰ آزمایش گراف‌های علیت مصنوعی از ۲۰ متغیر رسم شده است. هر متغیر دارای تعدادی از والدین است که به طور یکنواخت [۵،۰] کشیده شده‌اند.

تعداد نمونه‌ها: ۵۰۰ – تعداد ویژگی‌ها (گره‌ها): ۲۱ – مقادیر ویژگی‌ها: پیوسته – تعداد یال‌ها: ۴۲ – بیشترین درجه ورودی: ۵

نتایج بعد از خروجی گرفتن از روش به شرح زیر می‌باشد:

مقدار کران پایین واگرایی (ELBO) گام مربوط به گراف خروجی در آخرین اجرای شبکه عصبی، ۰/۰۰۰۱۲۱۹۸۷۷ بوده است. این گراف در زمان ۱ ساعت و ۳ دقیقه بدست آمده است.

جدول ۲: **وضعیت یال‌های گراف خروجی روش روی G5-v3-numdata**

عدم پیش‌بینی یال	پیش‌بینی یال نادرست	پیش‌بینی یال درست ـ جهت نادرست	پیش‌بینی یال درست ـ جهت درست
۲۵	۱۵	۵	۱۲

۶-۴- *نتایج روی مجموعه‌داده تراکنشی*

یک مجموعه‌داده تراکنشی با ۵ تراکنش و ۱۱ قلم داده، ابتدا به‌صورت دستی با الگوریتم Apriori [۵] الگوهای مکرر دوتایی با حداقل پشتیبان ۴۰ درصد را پیدا و سپس قوانین انجمنی با حداقل اطمینان ۶۰ درصد را تولید می‌کنیم. در ادامه، مجموعه‌داده را به شکل رشته‌ای از صفر و یک به روش DAG-GNN می‌دهیم و خروجی را بررسی می‌کنیم.

جدول ۳: **مجموعه‌داده‌ی تراکنشی با ۵ تراکنش و ۱۱ قلم داده**

اقلام	شماره تراکنش
{I1, I2, I3, I4, I5, I6}	۱
{I2, I3, I4, I5, I6, I7}	۲
{I1, I4, I5, I8}	۳
{I1, I4, I6, I9, I10}	۴
{I2, I4, I5, I10, I11}	۵

۱-۴-۶- *اجرای الگوریتم Apriori روی مجموعه‌داده تراکنشی*

الگوریتم Apriori یک الگوریتم جستجوی سطحی است، که با پایان کاوش در سطح (مرحله) kام به سطح بعدی یعنی k+1 می‌رود. این عمل تا محقق شدن شرط یا شروط پایانی تکرار می‌شود. در مرحله kام مجموعه اقلام kتایی تولید خواهند شد. پس از محاسبه مقدار پشتیبان برای هرکدام و مقایسه آن با مقدار حداقل پشتیبان، الگوهای مکرر kتایی شناسایی می‌شوند. در مرحله بعدی، الگوریتم با کمک الگوهای مکرر kتایی، مجموعه اقلام (k+1)تایی کاندید که می‌توانند مکرر باشند را تولید می‌کند. با توجه به مقدار حداقل پشتیبان برخی حذف شده و باقی‌مانده مجموعه اقلام (k+1)تایی را تشکیل می‌دهند.

این الگوریتم از قاعده‌ای استفاده می‌کند که بدین صورت بیان می‌شود: اگر یک الگوی مکرر داشته باشیم، کلیه زیر مجموعه‌های آن نیز مکرر هستند. به بیان دیگر، اگر مجموعه اقلام I مکرر نباشد، هر مجموعه‌ای که شامل I است نیز نمی‌تواند مکرر باشد.

جدول ۴: **تولید قانون از الگوهای مکرر دوتایی با حداقل اطمینان ۶۰ درصد**

قانون	پشتیبانی	اطمینان
I1 → I4	۳ (۶۰٪)	۱۰۰٪
I4 → I1	۳ (۶۰٪)	۶۰٪
I1 → I5	۲ (۴۰٪)	۶۶٪
I1 → I6	۲ (۴۰٪)	۶۶٪
I6 → I1	۲ (۴۰٪)	۶۶٪
I2 → I3	۲ (۴۰٪)	۶۶٪
I3 → I2	۲ (۴۰٪)	۱۰۰٪
I2 → I4	۳ (۶۰٪)	۱۰۰٪
I4 → I2	۳ (۶۰٪)	۶۰٪
I2 → I5	۳ (۶۰٪)	۱۰۰٪
I5 → I2	۳ (۶۰٪)	۷۵٪
I2 → I6	۲ (۴۰٪)	۶۶٪
I6 → I2	۲ (۴۰٪)	۶۶٪
I3 → I4	۲ (۴۰٪)	۱۰۰٪
I3 → I5	۲ (۴۰٪)	۱۰۰٪
I3 → I6	۲ (۴۰٪)	۱۰۰٪
I6 → I3	۲ (۴۰٪)	۶۶٪
I4 → I5	۴ (۸۰٪)	۸۰٪
I5 → I4	۴ (۸۰٪)	۱۰۰٪
I4 → I6	۳ (۶۰٪)	۶۰٪
I6 → I4	۳ (۶۰٪)	۱۰۰٪
I10 → I4	۲ (۴۰٪)	۱۰۰٪
I6 → I5	۲ (۴۰٪)	۶۶٪

۲-۴-۶- *اجرای روش DAG-GNN روی مجموعه‌داده تراکنشی*

مجموعه‌داده را به شکل رشته‌ای از صفر و یک درآورده و روش را روی آن اجرا می‌کنیم. هر سطر بیانگر یک تراکنش و مقادیر یک و صفر بیانگر وجود یا عدم وجود یک قلم داده در آن تراکنش می‌باشد (شکل ۴).

```
1  I1,I2,I3,I4,I5,I6,I7,I8,I9,I10,I11
2  1,1,1,1,1,1,0,0,0,0,0
3  0,1,1,1,1,1,1,0,0,0,0
4  1,0,0,1,1,0,0,1,0,0,0
5  1,0,0,1,0,1,0,0,1,1,0
6  0,1,0,1,1,0,0,0,0,1,1
```

شکل ۴: فایل csv از مجموعه‌داده تراکنشی

نتایج بعد از خروجی گرفتن از روش به شرح زیر می‌باشد:

مقدار کران پایین واگرایی (ELBO) گام مربوط گراف خروجی در آخرین اجرای شبکه عصبی، ۰/۰۰۰۳۷۹۰۸۹۴ بوده است. این گراف در زمان ۱/۲ دقیقه بدست آمده است.

جدول ۵: **وضعیت یال‌های گراف خروجی روش روی مجموعه‌داده تراکنشی**

یال‌های گراف خروجی	مقدار ضریب همبستگی در ماتریس خروجی
I2 → I1	-۰/۴۳۹
I2 → I8	-۰/۶۵۲
I2 → I11	۰/۴۶۶
I3 → I1	۰/۳۸۶
I3 → I2	۰/۲۷۴
I3 → I6	۰/۵۶۳
I3 → I9	-۰/۳۷۰
I3 → I11	-۰/۴۷۵
I4 → I3	-۰/۲۲۶
I5 → I6	-۰/۶۱۰
I5 → I8	۰/۵۶۴
I5 → I10	-۰/۶۲۷
I6 → I9	۰/۳۲۳
I7 → I1	-۰/۶۵۷
I7 → I3	۰/۲۲۸
I9 → I1	۰/۲۴۰
I9 → I2	-۰/۲۶۴
I9 → I10	۰/۳۳۵
I11 → I10	۰/۵۷۶

۷. نتیجه‌گیری

در اجرای اول و دوم باتوجه به تعداد قابل توجه یال‌هایی که درست پیشبینی شده‌اند ولی جهت آن‌ها نادرست است، در اجراهای دیگر ابتدا خود مفهوم یال و بعد جهت یال پیشبینی شده، در نظر گرفته شود.

در اجرای اول، با بررسی مقادیر جدول‌های احتمال شرطی برای ساختار خروجی روش و ساختار حقیقی، این انگیزه ایجاد می‌شود که قبل از نتیجه گیری ابتدا توسط خبره سامانه، ساختار اصلاح شود.

در اجرای سوم، دیدیم که الگوریتم Apiori حتی با تعداد کم اقلام داده، حجم وسیعی از قوانین را تولید کرد. حتی در مواردی علاوه بر یک قانون در جهت عکس آن قانون نیز قانون تولید کرده بود. درحالیکه که روش DAG-GNN تعداد قانون محدودتر و تنها در یک جهت تولید کرد. همچنین بجای اینکه تنها تکرار یک قلم داده را در نظر بگیرد، توزیع آن قلم داده را حتی در زمانی که صفر هم هست، در نظر گرفته است.

۸. محدودیت‌ها و پیشنهادات

روش DAG-GNN نمی‌تواند با مجموعه‌داده حاوی مقادیر گسسته و پیوسته، کار کند و همگرا نمی‌شود. می‌بایست داده‌ها گسسته و یا پیوسته باشد. بنابراین توسعه این مدل به مدلی که بتواند به طور همزمان هر دو نوع متغیر را در نظر بگیرد به عنوان یک هدف پژوهشی در نظر گرفته شده است.

برای کاوش قوانین انجمنی داده‌ها می‌بایست به شکل رشته‌ای از صفر و یک درآورده و روش DAG-GNN را اجرا کنیم. هر سطر بیانگر یک تراکنش و مقادیر یک و صفر بیانگر وجود و یا عدم وجود یک قلم داده در آن تراکنش می‌باشد.

در ادامه میتوان کاربرد گراف‌های جهت‌دار بی‌دور در مسائل و ارائه راهکار برای آن مسائل برحسب داده مشاهده‌ای را بررسی نمود. همچنین روش‌های یادگیری ساختار شبکه بیزی را مورد پژوهش قرار داد و بهبود بخشید.

مراجع

[۱] Y. Yu, J. Chen, T. Gao, and M. Yu, "DAG-GNN: DAG structure learning with graph neural networks," in *International Conference on Machine Learning*, 2019: PMLR, pp. 7154-7163 .

[۲] X. Zheng, B. Aragam, P. K. Ravikumar, and E. P. Xing, "Dags with no tears :Continuous optimization for structure learning," *Advances in Neural Information Processing Systems*, vol. 31, 2018.

[۳] K. Sachs, O. Perez, D. Pe'er, D. A. Lauffenburger, and G. P. Nolan, "Causal protein-signaling networks derived from multiparameter single-cell data," *Science*, vol. 308, no. 5721, pp. 523-529, 2005.

[۴] O. Goudet. *Graph inference datasets. Replication Data for: "Learning Functional Causal Models with Generative Neural Networks"*, Harvard Dataverse, doi: doi:10.7910/DVN/UZMB69.

[۵] R. Agrawal and R. Srikant, "Fast algorithms for mining association rules," in *Proc. 20th int. conf. very large data bases, VLDB*, 1994, vol. 1215: Citeseer, pp. 487-499 .