بسمه تعالی

استنتاج علّی
گزارش پروژه تحقیقاتی نظری

نیمسال اول ۹۸-۹۷
شماره دانشجویی: ۹۵۱۰۹۵۶۴

Additive Noise Models: Identifiability Theorems, Learning Algorithms, Hidden Variables and Time Series

بهراد منیری

۱ مقدمه

در این پروژه به بررسی مدل‌های نویز جمعی، Additive Noise Models، خواهیم پرداخت. به طور خاص، قضایای قابل شناسایی بودن جهت درست علّی را بررسی خواهیم کرد، مروری بر الگوریتم‌های یادگیری ساختار علّی خواهیم داشت و همچنین درباره‌ی مدل‌های نویز جمعی در وجود متغیرهای پنهان و در سری‌های زمانی بحث خواهیم کرد.

مدل‌های نویز جمعی از اهمیت بالایی در استنتاج علّی برخوردارند. این روش که معادل جست و جو در یک فضای توابع محدود شده است شباهیت زیادی با یادگیری ماشین پیدا می‌کند. وجود قضایای قابل شناسایی بودن گراف با داشتن توزیع مشاهداتی که به ما امکان دستیابی به اطلاعاتی بیش از کلاس هم‌ارزی مارکوف گراف مولد را می‌دهد، یکی از جذابیت‌های مدل های نویز جمعی است [۲]. الگوریتم‌های بهینه‌تری نسبت به الگوریتم‌هایی مانند PC برای یادگیری ساختار علّی SCM هایی که در آن‌ها مدل‌های نویز جمعی وجود دارد [۶]. به نظر می‌رسد که در جهان واقعی، خیلی از پدیده‌ها از مدل نویز جمعی پیروی می‌کنند [۶، ۷]. به طرز شگفت‌آوری، مدل‌های مبتنی بر نویز جمعی در یادگیری سری‌های زمانی نیز توانسته‌اند به نتایج تجربی امیدوارکننده‌ای دست پیدا کنند [۵].

۲ تعریف مسئله

یک SCM، C، در نظر بگیرید. می‌گوییم این SCM از مدل نویز جمعی پیروی می‌کند اگر

$$S_j: \quad f_j = f_j(\mathbf{PA}_j) + N_j, \qquad j = 1, 2, ..., p \tag{۱}$$

که در آن $\mathbf{PA}_j$ مجموعه‌ی والدین X_j هستند و متغیرهای نویز، N_j دارای چگالی‌احتمالی اکیداً مثبت می‌باشند و مستقل‌اند. همچنین در همه‌ی بخش‌ها فرض می‌شود گراف مولد داده‌ها یک DAG است مگر به طور صریح خلاف این موضوع ذکر شود.

۳ قضایای قابل‌شناسایی بودن

در این بخش به بررسی مدل‌های نویز جمعی در حالاتی که هیچ Common Cause مشاهده‌نشده‌ای در سیستم وجود ندارد می‌پردازیم. در مدل‌های نویز جمعی، هدف محدود کردن مجموعه‌ی توابعی است که در آن به جست‌وجوی توابع تولید کننده SCM می‌گردیم. مراجع ما در این بخش، مقالات [۲]، [۶] و [۷] هستند. این مقالات به بررسی قضایای Identifiability در مدل‌های نویز جمعی پرداخته اند. ما در این بخش به بررسی قضایای مطرح شده در این مقالات پرداخته و در آخر محدودیت‌های آن‌ها را با هم مقایسه می‌کنیم.

قضیه ۱.۳. درمدل‌های نویز جمعی، شرط *Causal Minimality* معادل این است که توابع f_j نسبت به هیچ یک از متغیرهایشان ثابت نباشند.

۱.۳ حالت دو متغیره

Hoyer et al. در مقاله‌ی [۲] قضیه‌ی زیر را در مورد قابل شناسایی بودن ANM در حالت دو متغیره اثبات می‌کند. در ادامه به بررسی دقیق این قضیه پرداخته و بحث خواهیم کرد که در چه شرایطی، در یک مدل جمعی، از روی چگالی احتمال مشترک قادر به شناسایی کامل جهت علّی نیستیم.

قضیه ۲.۳. فرض کنید داشته باشیم

$$\begin{cases} X = N_x \\ Y = f(X) + N_y \end{cases}$$

اگر برای هر x, y با شرط $\nu''(y - f(x))f'(x) \neq 0$ حل معادله‌ی دیفرانسیل زیر نباشد:

$$\xi''' = \xi'' \left(-\frac{\nu''' f'}{\nu''} + \frac{f''}{f'} \right) - 2\nu'' f'' f' + \nu' f''' + \frac{\nu' \nu''' f'' f'}{\nu''} - \frac{\nu' (f'')^2}{f'}\,, \tag{۲}$$

که در آن $\xi := \log(p_X)$ و $\nu := \log(p_{N_Y})$ است، مدلی با نویز جمعی پیوسته در جهت دیگر وجود نداشته و جهت علّی، با داشتن توزیع مشترک متغیرها قابل شناسایی است. در شرط فوق، آرگومان توابع و مشتق توابع ν، ξ و f به ترتیب $y - f(x)$، x و x هستند.

اثبات. فرض کنید که گراف قابل شناسایی نباشد و دو گراف با جهت‌های متضاد بر این داده‌ها قابل برازش باشد. در این صورت خواهیم داشت:

$$p(x, y) = p_n(y - f(x)) p_x(x) = p_{\tilde{n}}(x - g(y)) p_y(y)\,. \tag{۳}$$

π را برابر تابع log-liklihood در نظر بگیرید:

$$\pi(x, y) := \log p(x, y) = \nu(y - f(x)) + \xi(x)\,, \tag{۴}$$

از طرف دیگر، بنا بر معادله‌ی (۳) داریم:

$$\pi(x, y) = \tilde{\nu}(x - g(y)) + \eta(y) \tag{۵}$$

با مشتق گیری از این معادله خواهیم داشت:

$$\frac{\partial^2 \pi}{\partial x \partial y} = -\tilde{\nu}''(x - g(y)) g'(y) \quad \text{و} \quad \frac{\partial^2 \pi}{\partial x^2} = \tilde{\nu}''(x - g(y))\,.$$

در نتیجه داریم:

$$\frac{\partial}{\partial x} \left(\frac{\partial^2 \pi / \partial x^2}{\partial^2 \pi / (\partial x \partial y)} \right) = -\frac{\partial}{\partial x} g'(y) = 0\,. \tag{۶}$$

از معادله‌ی (۴) به دست می‌آید:

$$\frac{\partial^2 \pi}{\partial x \partial y} = -\nu''(y - f(x)) f'(x)\,, \tag{۷}$$

و همچنین

$$\frac{\partial^2 \pi}{\partial x^2} = \frac{\partial}{\partial x} \left(-\nu'(y - f(x)) f'(x) + \xi'(x) \right) = \nu'' (f')^2 - \nu' f'' + \xi''\,, \tag{۸}$$

که در آن آرگومان‌ها برای خوانایی بیشتر حذف شده‌اند. از معادله‌ی (۷) و (۸)

$$\frac{\partial}{\partial x} \left(\frac{\frac{\partial^2 \pi}{\partial x^2}}{\frac{\partial^2 \pi}{\partial x \partial y}} \right) = -2f'' + \frac{\nu' f'''}{\nu'' f'} - \xi''' \frac{1}{\nu'' f'} + \frac{\nu' \nu''' f''}{(\nu'')^2} - \frac{\nu' (f'')^2}{\nu'' (f')^2} - \xi'' \frac{\nu'''}{(\nu'')^2} + \xi'' \frac{f''}{\nu'' (f')^2}\,.$$

بر اساس معادله‌ی (۶)، عبارت فوق برابر صفر است یعنی:

$$\xi''' = \xi'' \left(-\frac{\nu''' f'}{\nu''} + \frac{f''}{f'} \right) - 2\nu'' f'' f' + \nu' f''' + \frac{\nu' \nu''' f'' f'}{\nu''} - \frac{\nu' (f'')^2}{f'}$$

با بازنویسی این معادله بر حسب توابع اصلی مساله داریم:

$$p_x''' = p_x' p_n'' f' \left(-\frac{p_n'''}{(p_n'')^2} + \frac{f''}{p_n'' (f')^2} \right) + p_n'' f' \left(-2f'' + \frac{p_n' f'''}{p_n'' f'} + \frac{p_n' p_n''' f''}{(p_n'')^2} - \frac{p_n' (f'')^2}{p_n'' (f')^2} \right).$$

□

سوال مهمی که در اینجا مطرح می‌شود این است که در چه شرایطی، قادر به تشخیص جهت از روی چگال مشترک نیستیم. قضیه‌ی زیر به ما می‌گوید که در صورتی که به صورت کاملاً تصادفی از مدل‌های نویز جمعی یک مدل را انتخاب کنیم، احتمال قابل شناسایی نبودن جهت علّی صفر است.

قضیه ۳.۳. اگر نویزها گاوسی باشند، یعنی $\xi''' = \nu''' = 0$، مدل نویز جمع‌شونده در هر دو جهت وجود دارد اگر f خطی باشد.

اثبات. نتیجه‌ی مستقیم قضیه‌ی (۲.۳). □

قضیه ۴.۳. اگر برای مجموعه‌ی توابع f و توزیع نویزهای خارجی داده‌شده، برای یک y خاص، $v''(y-f(x))f'(x)=0$ شمارا جواب داشته باشد، یا جوابی نداشته باشد، توزیع $\mathbf{X}$ در یک فضای سه بعدی زندگی می‌کند.

از آنجا که مجموعه‌ی توزیع‌های پیوسته بی‌نهایت بعدی است، برای «اکثر» مدل‌های نویز جمعی، جهت علّی قابل شناسایی است.

اثبات. y فیکس‌ای در نظر بگیرید به طوری که $\nu''(y-f(x))f'(x) \neq 0$ در همه‌ی xها به جز تعدادی شمارا برقرار باشد. برای هر f, ν داده‌شده، بنا بر قضیه‌ی ۲.۳ یک معادله‌ی دیفرانسیل برای ξ به دست می‌آوریم:

$$\xi'''(x) = \xi''(x)G(x,y) + H(x,y)\,, \tag{۹}$$

که در آن H و G برابرند با

$$G := -\frac{\nu''' f'}{\nu''} + \frac{f''}{f'}$$

و

$$H := -2\nu'' f'' f' + \nu' f''' + \frac{\nu'\nu''' f'' f'}{\nu''} - \frac{\nu'(f'')^2}{f'}\,,$$

با حل این معادله‌ی دیفرانسیل برای ξ'' داریم

$$\xi''(x) = \xi''(x_0)e^{\int_{x_0}^{x} G(\tilde{x},y)d\tilde{x}} + \int_{x_0}^{x} e^{\int_{\hat{x}}^{x} G(\tilde{x},y)d\tilde{x}} H(\widehat{x},y)d\widehat{x}\,. \tag{۱۰}$$

مجموعه‌ی توابع ξای که در معادله‌ی دیفرانسیل مذکور صدق می‌کنند، در یک زیرفضای سه بعدی آفین زندگی می‌کنند که با سه عدد $\xi(x_0), \xi'(x_0), \xi''(x_0)$ تابع به طور یکتا تعیین می‌شود. در نتیجه اثبات کردیم که برای یک تابع و مجموعه‌ی توزیع نویزهای خارجی داده‌شده، مجموعه‌ی تویزیع احتمال Xهایی که به اجازه‌ی وجود یک مدل برعکس می‌دهند، در یک زیرفضای سه بعدی از فضای بی‌نهایت بعدی توزیع‌های پیوسته هستند. □

با وجود اینکه احتمال اینکه برای یک مدل نویز جمعی، مدل نویز جمعی دیگری در جهت مخالف وجود داشته باشد بسیار نادر است، این سوال مطرح است که در چه شرایطی برای یک مدل نویز جمعی چنین اتفاقی رخ می‌دهد. Zhang et al. در [۷] پنج دسته مدل نویز جمعی معرفی می‌کند و اثبات می‌کند هر مدل نویز جمعی غیر قابل شناسایی از تابع چگالی احتمال، به ناچار در یکی از این دسته‌ها قرار می‌گیرد.

قضیه ۵.۳. فرض کنید $X_2 = f_2(X_1) + N_2$ باشد و N_2 نیز نویزی *full support* و مستقل از X_1 باشد، تابع f_2 سه بار مشتق‌پذیر بوده و همچنین معادله‌ی $\frac{d}{dx_1}f_2(x_1)\frac{d^2}{dx_1^2}\log(p_{N_2}(x_2)) = 0$ تنها در تعدادی متناهی نقطه‌ی (x_1, x_2) برقرار باشد. در صورتی که یک مدل در جهت بر عکس وجود داشته باشد، به این معنا که $X_1 = g_1(X_2) + \tilde{N}_1$ که X_2 و $\tilde{N}_1$ در آن مستقل باشد، یکی از پنج حالت زیر برقرار است.

I. X_1 is Gaussian, N_2 is Gaussian and f is linear.

II. X_1 is log-mix-lin-exp, N_2 is log-mix-lin-exp and f is linear.

III. X_1 is log-mix-lin-exp, N_2 is one-sided asymptotically exponential and f is strictly monotonic with $f'(x_1) \to 0$ as $x_1 \to \infty$ or as $x_1 \to -\infty$.

IV. X_1 is log-mix-lin-exp, N_2 is generalized mixture of two exponentials and f is strictly monotonic with $f'(x_1) \to 0$ as $x_1 \to \infty$ or as $x_1 \to -\infty$.

V. X_1 *is generalized mixture of two exponentials,* N_2 *is two-sided asymptotically exponential and* f *is strictly monotonic with* $f'(x_1) \to 0$ *as* $x_1 \to \infty$ *or as* $x_1 \to -\infty$.

تعاریف دقیق عبارات به کار رفته در جدول فوق را در تعریف زیر آورده‌ایم:

تعریف ۱.۳. فرض کنید p چگالی احتمال یک توزیع پیوسته P باشد.

- P یک *log-mix-lin-exp* است اگر وجود داشته باشند c_1, c_2, c_3, c_4 به نحوی که $c_1 < 0$ و $c_2c_3 > 0$ به صورتی که:

$$\log p(x) = c_1 \exp(c_2 x) + c_3 x + c_4.$$

- P *one-sided asymptotically exponential* است اگر وجود داشته باشد $c \neq 0$ به نحوی که

$$\frac{d}{dx} \log p(x) \to c$$

وقتی $x \to -\infty$ یا $x \to \infty$.

- P یک *two-sided asymptotically exponential* است اگر وجود داشته باشند $c_1 \neq 0$ و $c_2 \neq 0$ به نحوی که

$$\frac{d}{dx} \log p(x) \to c_1$$

وقتی $x \to -\infty$ و

$$\frac{d}{dx} \log p(x) \to c_2$$

وقتی $x \to \infty$.

- P یک *generalized mixture of two exponentials* است اگر $d_1, d_2, d_3, d_4, d_5, d_6$ وجود داشته باشند به نحوی که $d_4 > 0$، $d_3 > 0$، $d_1 d_5 > 0$ و $d_2 < -\frac{d_1}{d_5}$ و داشته باشیم:

$$\log p(x) = d_1 x + d_2 \log(d_3 + d_4 \exp(d_5 x)) + d_6.$$

۲.۳ حالت چند متغیره

تا اینجا برای حالت دو بعدی، نشان دادیم در حالت generic یک توزیع احتمال، اجازه‌ی وجود مدل نویز جمعی در هر دو طرف را نمی‌دهد. در این بخش به تعمیم این قضیه از دوبعدی به حالت چندبعدی می‌پردازیم. مرجع اصلی ما در این بخش مقاله‌ی [۶] است. این مقاله قضیه‌ای بسیار جالب را مطرح می‌کند که عنوان می‌کند هنگامی یک قضیه‌ی Identifiability دو بعدی داریم در چه صورتی می‌توان آن را به حالت چندبعدی تعمیم داد. برای ورود به این بحث مقاله‌ی [۶] مثالی جالب را مطرح کرده که در این گزارش نیز به همان شیوه‌ی مقاله عمل می‌کنیم.

مثال ۱.۳. SCM زیر را در نظر بگیرید:

$$\begin{cases} X_1 = N_1 \\ X_2 = f_2(X_1) + N_2 \\ X_3 = f_3(X_1) + aX_2 + N_3 \end{cases} \tag{۱۱}$$

که در آن $N_1 \sim \mathrm{t-student}(\nu = 3)$، $N_2 \sim \mathrm{Normal}(0, \sigma_2^2)$ و $N_3 \sim \mathrm{Normal}(0, \sigma_3^2)$. در این‌جا X_2 و X_3 غیر گاوسی هستند اما

$$X_3 | X_2 = x_2 = c + aX_2 | X_1 = x_1 + N_3$$

برای هر x_1 یک معادله‌ی خطی-گاوسی است در حالی که هیچ‌یک از معادلات اصلی SCM گاوسی-خطی نیستند.
می‌توانیم SCM دیگری بسازیم که در توزیع مشاهداتی تفاوتی با SCM اصلی نداشته باشد:

$$\begin{cases} X_1 = M_1 \\ X_2 = g_2(X_1) + bX_3 + M_2 \\ X_3 = g_3(X_1) + aX_2 + M_3 \end{cases} \tag{۱۲}$$

به نظر می‌رسد باید شرطی بر روی توزیع‌های شرطی قرار دهیم!

برای بیان شرط قابل شناسایی بودن از توزیع احتمال مشاهداتی، به یک تعریف نیاز داریم:

تعریف ۲.۳. یک مدل نویز جمعی با n متغیر را در نظر بگیرید. این مدل را یک مدل نویز جمعی محدودشده می‌نامیم اگر برای هر $j \in \mathbf{V}$ و $i \in \mathbf{PA}_j$ و تمام مجموعه‌های $\mathbf{S} \subseteq \mathbf{V}$ به طوری که $\mathbf{PA}_j \setminus \{i\} \subseteq \mathbf{S} \subseteq \mathbf{ND}_j \setminus \{i, j\}$ وجود داشته باشد $x_\mathbf{S}$ که $p_\mathbf{S}(x_\mathbf{S}) > 0$ به نحوی که

$$\Big(f_j(x_{\mathbf{PA}_j \setminus \{i\}}, \underbrace{\cdot}_{X_i}), P(X_i | X_\mathbf{S} = x_\mathbf{S}), P(N_j)\Big)$$

در شرایط قضیه‌ی (۲.۳) صدق کند.

قضیه ۶.۳. فرض کنید که $\mathbf{X} = \{X_1, X_2, \dots, X_n\}$ توسط یک مدل نویز جمعی محدودشده با گراف G_0 تولید شده‌باشند و فرض کنید $P(\mathbf{X})$ نسبت به G_0 شرط *Causal Minimality* را ارضا کند. در این صورت G_0 از روی توزیع احتمال $P(\mathbf{X})$ قابل شناسایی است.

برای اثبات این قضیه نیاز به یک لم گرافی داریم که Chickering در سال ۱۹۹۵ اثبات کرده است [۱].

لم ۱.۳. فرض کنید G و G' دو DAG روی مجموعه‌ی متغیرهای $\mathbf{X}$ باشند. فرض کنید $P(\mathbf{X}$ چگالی احتمالی همواره مثبت دارد که نسبت به G و G' مارکوف هستند و شرط *Causal Minimality* را ارضا می‌کنند. در این صورت متغیرهای L و Y وجود دارند که برای مجموعه‌های $\mathbf{Q} := \mathbf{PA}_L^G \setminus \{Y\}$، $\mathbf{R} := \mathbf{PA}_Y^{G'} \setminus \{L\}$ و $\mathbf{S} := \mathbf{Q} \cup \mathbf{R}$ داشته باشیم:

- $Y \to L$ در G و $L \to Y$ در G'.
- $\mathbf{S} \subseteq \mathbf{ND}_L^G \setminus \{Y\}$ و $\mathbf{S} \subseteq \mathbf{ND}_Y^{G'} \setminus \{L\}$

اثبات. **قضیه‌ی** (۶.۳)
برهان خلف: فرض کنید دو مدل نویز جمعی محدود شده با این توزیع احتمال وجود داشته باشند. یکی با گراف G و دیگری با گراف متفاوت G'. دو متغیر L و Yرا بر اساس لم فوق انتخاب می‌کنیم و مشابه لم فوق، می‌گیریم $\mathbf{Q} := \mathbf{PA}_L^G \setminus \{Y\}$، $\mathbf{R} := \mathbf{PA}_Y^{G'} \setminus \{L\}$ و $\mathbf{S} := \mathbf{Q} \cup \mathbf{R}$ هر تحقق دلخواه $\mathbf{s} = (\mathbf{q}, \mathbf{r})$ را در نظر بگیرید و بنویسید $L^* := L | \mathbf{S} = \mathbf{s}$ و $Y^* := Y | \mathbf{S} = \mathbf{s}$ از لم فوق داریم: $\mathbf{S} \subseteq \mathbf{ND}_L^G \setminus \{Y\}$ و این بدین معناست که $\{Y\} \cup \mathbf{S} \subseteq \mathbf{ND}_L^G$ در نتیجه $N_L \perp\!\!\!\perp \{Y\} \cup \mathbf{S}$ در نتیجه می‌توان به معادله‌ی زیر رسید:

$$L^* = f_L(\mathbf{q}, Y^*) + N_L \quad N_L \perp\!\!\!\perp Y^*$$

برای G' هم با استدلال مشابه می‌توان به معادله‌ی زیر رسید:

$$Y^* = g_Y(\mathbf{r}, L^*) + N_Y \quad N_Y \perp\!\!\!\perp L^*$$

و این در تناقض با فرض مدل نویز جمعی محدود شده است پس فرض خلف باطل بوده و G و G' برابرند. □

نکته‌ی بسیار جالبی که در این قضیه وجود دارد این است که می‌توان آن را برای هر قضیه‌ی Indentifiability دیگر نیز به کار برد، به شرط عدم وجود دور. به طور مثال به کمک آن می‌توان LINGAM و یا Post Non-Linear Additive Noise Model، که در ادامه به آن پرداخته خواهد شد، را از حالت دو بعدی به حالت چند بعدی تعمیم داد.

۳.۳ Post Non-Linear Models

دسته‌ای دیگر از SCM ها وجود دارند که برای شناسایی آن‌ها از توزیع احتمال مشترک متغیرهایشان، قضیه‌ی identifiablity وجود دارد. این دسته از SCM ها، Post Non-Linear Models ها هستند. این مدل‌ها در [۷] معرفی شده‌اند.

تعریف ۳.۳. ***Post Non-Linear Models*** یک *SCM* را *Post Non-Linear Models* اگر روابط *Structural* آن به فرم

$$X_i = f_{i,2}(f_{i,1}(\mathbf{PA}_i) + N_i)$$

باشد.

مقاله‌ی [۷] با روشی کاملاً مشابه روش (۲.۳)، قضیه‌ی identifiablity زیر برای را برای Post Non-Linear Models ها اثبات می‌کند.

قضیه ۷.۳. فرض کنید $x_2 = f_2(f_1(x_1) + e2)$ و $x_1 = g_2(g_1(x_2) + e1)$. توابع غیرخطی، توزیع‌های چگالی احتمال داشته و هر کدام از توابع و توابع چگالی احتمال سه بار مشتق‌پذیر باشند، برای هر (x_1, x_2) که $\eta''h' \neq 0$:

$$t_1 = g_2^{-1}(x_1),\ z_2 = f_2^{-1}(x_2),\ h = f_1 \circ g_2,\ h_1 = g_1 \circ f_2$$

$$\eta_1(t_1) = \log p_{t_1}(t_1) \ \ \eta_2(e_2) = \log p_{e_2}(e_2)$$

$$\eta_1''' - \frac{\eta_1''h''}{h'} = (\frac{\eta_2'\eta_2'''}{\eta_2''} - 2\eta_2'').h'h'' - \frac{\eta_2'''}{\eta_2''}.h'\eta_1'' + \eta_2'.(h''' - \frac{h''^2}{h'})$$

و همچنین چنین رابطه‌ای نیز وجود خواهد داشت:

$$\frac{1}{h_1'} = \frac{\eta_1'' + \eta_2''h'^2 - \eta_2'h''}{\eta_2''h'}$$

با فرضیاتی که در مورد توابع و توزیع‌ها داشتیم، می‌توان تمام حالاتی که در آن گراف از روی تابع چگالی مشترک قابل شناسایی نیست را به دست آورد. شکل (۱) را ببینید.

Table 1: All situations in which the PNL causal model is not identifiable.

	p_{e_2}	p_{t_1} $(t_1 = g_2^{-1}(x_1))$	$h = f_1 \circ g_2$	Remark
I	Gaussian	Gaussian	linear	h_1 also linear
II	log-mix-lin-exp	log-mix-lin-exp	linear	h_1 strictly monotonic, and $h_1' \to 0$, as $z_2 \to +\infty$ or as $z_2 \to -\infty$
III	log-mix-lin-exp	one-sided asymptotically exponential (but not log-mix-lin-exp)	h strictly monotonic, and $h' \to 0$, as $t_1 \to +\infty$ or as $t_1 \to -\infty$	—
IV	log-mix-lin-exp	generalized mixture of two exponentials	Same as above	—
V	generalized mixture of two exponentials	two-sided asymptotically exponential	Same as above	—

شکل ۱: تمام حالاتی که در آن مدل قابل شناسایی نیست.

۴.۳ مقایسه‌ی نتایج

در این بخش سه مقاله‌ی اصلی که قضایای identifiablity مربوط به مدل‌های نویز جمعی ارائه داده‌اند را بررسی کردیم. در ادامه به مقایسه این مقالات می‌پردازیم.

مقاله Hoyer et. al برای اولین بار مدل‌های نویز جمعی را معرفی می‌کند و قضیه‌ی identifiablity دو بعدی را ثابت می‌کند که سنگ‌بنای تمام کارهای بعدی این حوزه است. این مقاله نتایج خود را با داده‌های واقعی و همچنین داده‌های مصنوعی نیز می‌سنجد. یک نقطه‌ی ضعف این مقاله این است که در آزمایش‌های تجربی خود، حالت‌هایی به جز حالت گاوسی-خطی که در آن identifiablity نداریم را بررسی نمی‌کند. این ایراد در تمام مقالات بعدی نیز وجود دارد.

مقاله Peters et al. با ارائه‌ی شیوه‌ای برای تعمیم قضایای identifiablity به چند بعد، قدم بزرگی در استنتاج‌علی بر می‌دارد که حتی در بیرون از مدل‌های نویز جمعی نیز کاربرد وسیع دارد. الگوریتم RESIT که در این مقاله معرفی می‌شود نیز تا کنون جز بهترین ابزارهای یادگیری ساختار علّی باقی‌مانده است و ایده‌ی آن در مقالاتی مثل [۵] نیز به کار رفته است. بخش آزمایش الگوریتم بر روی داده‌های واقعی این مقاله ضعف‌های اساسی دارد. این مقاله الگوریتم خود را تنها بر روی یک دیتاست می‌آزماید که در این دیتاست، جهت درست علّی به درستی مشخص نیست. در ادامه بیشتر به الگوریتم RESIT خواهیم پرداخت.

مقاله Zhang قضیه identifiablity بسیار قوی‌ای برای مدل‌های Post Non-Linear ارائه داده است که می‌توان با روش مقاله‌ی [۶] به راحتی آن را به حالت چند متغیره نیز تعمیم داد. برای یادگیری Post Non-Linear ها، تاکنون الگوریتم یادگیری مناسبی ارائه نشده است. به نظر من Post Non-Linear ها هنوز به بلوغ کامل نرسیده‌اند و باید کار بسیار بیشتری بر روی الگوریتم‌های یادگیری آن‌ها انجام شود.

۴ الگوریتم‌های یادگیری

در این بخش به بررسی الگوریتم‌های یادگیری گراف مربوط به یک SCM از داده‌ی محدود به فرض اینکه در SCM مدل نویز جمعی برقرار باشد، پرداخته و الگوریتم‌های مختلف مطرح شده را با هم مقایسه می‌کنیم. مراجع ما در این بخش مقالات [۶] و [۴] هستند.

۱.۴ الگوریتم RESIT

این الگوریتم توسط Peters et al.، سال ۲۰۱۴، در [۶] معرفی شده است. ایده‌ی اصلی این الگوریتم این است که برای هر X_i اگر X_i یک Sink Node باشد داریم $N_i \perp\!\!\!\perp \mathbf{X} \setminus \{X_i\}$ به طور کلی برای هر $Y \in \mathbf{X}$، $N_Y \perp\!\!\!\perp \mathbf{ND}_Y$.

Algorithm 1 Regression with subsequent independence test (RESIT)

1: **Input:** i.i.d. samples of a p-dimensional distribution on $(X_1, \dots, X_p)$
2: $S := \{1, \dots, p\}, \pi := [\,]$
3: PHASE 1: Determine causal order.
4: **repeat**
5:   **for** $k \in S$ **do**
6:     Regress X_k on $\{X_i\}_{i \in S \setminus \{k\}}$.
7:     Measure dependence between residuals and $\{X_i\}_{i \in S \setminus \{k\}}$.
8:   **end for**
9:   Let k^* be the k with the weakest dependence.
10:   $S := S \setminus \{k^*\}$
11:   $\mathrm{pa}(k^*) := S$
12:   $\pi := [k^*, \pi]$ (π will be the causal order, its last component being a sink)
13: **until** $\#S = 1$
14: PHASE 2: Remove superfluous edges.
15: **for** $k \in \{2, \dots, p\}$ **do**
16:   **for** $\ell \in \mathrm{pa}(\pi(k))$ **do**
17:     Regress $X_{\pi(k)}$ on $\{X_i\}_{i \in \mathrm{pa}(\pi(k)) \setminus \{\ell\}}$.
18:     **if** residuals are independent of $\{X_i\}_{i \in \{\pi(1), \dots, \pi(k-1)\}}$ **then**
19:       $\mathrm{pa}(\pi(k)) := \mathrm{pa}(\pi(k)) \setminus \{\ell\}$
20:     **end if**
21:   **end for**
22: **end for**
23: **Output:** $(\mathrm{pa}(1), \dots, \mathrm{pa}(p))$

الگوریتم RESIT در هر مرحله یک Sink Node را تشخیص داده و حذف می‌کند. برای تشخیص یک Sink نیز از ویژگی $N_i \perp\!\!\!\perp \mathbf{X} \setminus \{X_i\}$ استفاده می‌کند.

الگوریتم RESIT دو فاز دارد. در فاز اول (خط ۳ تا ۱۳)، یک Causal Order پیدا می‌شود. با رگرس کردن هر متغیر روی بقیه‌ی متغیرهای گراف هر مرحله، متغیری که باقی مانده‌ی رگرسیون مربوط به از دیگر متغیرها مستقل‌تر(مثلاً با معیار p-value آزمون HSIC) باشد را به عنوان یک sink در نظر می‌گیریم. با حذف این راس، مجدداً یک DAG دیگر به وجود می‌آید که در آن همین روند را روی آن تکرار می‌کنیم. با این کار می‌توان به یک Causal Order برای متغیرها رسید. در فاز دوم، برای شروع فرض می‌شود که اگر $\pi(i) < \pi(j)$، از i به j یک یال وجود دارد . از این گراف شروع کرده. هر بار یک متغیر، $X_{\pi(k)}$، را در نظر گرفته و آن را بر روی parent هایش به جز یک parent، X_l، رگرس می‌کنیم به نحوی که هر parent یک بار از رگرسیون کنار گذاشته شود. در هر رگرسیون، اگر باقی‌مانده رگرسیون از متغیرهایی که در Causal Order بالاتر از $X_{\pi(k)}$ هستند مستقل شد، ارتباط X_l و $X_{\pi(k)}$ را حذف می‌کنیم.

الگوریتم RESIT در مرحله‌ی اول خود $O(n^2)$ تست آماری انجام می‌دهد و در مرحله‌ی دوم نیز تعداد تست های آماری $O(n)$ است. چند جمله‌ای بودن این الگوریتم بسیار عجیب است زیرا مسائل معمول در Bayesian Network Learning اکثراً NP-Hard هستند. با این وجود الگوریتم RESIT برای n های بزرگ قابل استفاده نیست زیرا در صورتی که در انجام تست آماری دچار خطا شویم، خطا به شدت در مراحل بعد منتشر شده و باعث می‌شود به طور قابل ملاحظه‌ای از گراف اصلی دور شویم.

۲.۴ روش Brute Force و GDS

یک دسته‌ی دیگر از الگوریتم‌های یادگیری در مدل‌های نویز جمعی، الگوریتم‌های مبتنی بر score هستند.
برای یادگیری ساختار مدل‌های نویز جمعی محدود شده، می‌توان تمام DAG ها را enumerate کرد و بررسی کرد که آیا استقلال‌هایی که از داده‌ها استخراج می‌شوند در گراف نیز وجود دارند یا خیر. یک ایراد این روش این است که این روش لزوما یک گراف Causal Minimal به ما نمی‌دهد. برای حل این مشکل یک penalized independence score تعریف کرده و آن را برای گراف‌ها محاسبه می‌کنیم و این معیار را مبنای انتخاب گراف قرار می‌دهیم.

$$\widehat{G} = \mathrm{argmin}_G \sum_{i=1}^{n} \mathrm{DM}(res_i^{G,\mathrm{RM}}, res_{-i}^{G,\mathrm{RM}}) + \lambda \#\mathrm{edges}$$

در آن RM روش رگرسیون ما و DM یک معیار استقلال است. res_i مقدار باقی مانده‌ی رگرسیون X_i است وقتی آن را بر روی تمام parent هایش رگرس می‌کنیم و res_{-i} باقی‌مانده‌ی رگرسیون مابقی متغیر ها به جز X_i است. واضح است که این الگوریتم برای گراف‌های بزرگ به هیچ وجه قابل استفاده نخواهد بود.
یک راه معقول برای کاهش پیچیدگی محاسباتی الگوریتم فوق، استفاده از روش‌های حریصانه است [۶]. دو گراف را مجاور می‌گوییم اگر تنها با یک تغییر جهت، افزودن یال یا کاستن یال به یکدیگر تبدیل شوند. در الگوریتم حریصانه، در هر مرحله score گراف آن مرحله را حساب کرده و با score گراف‌های مجاور مقایسه می‌شود. هرجا امتیاز یکی از این گراف‌های مجاور از گراف اصلی بالاتر بود، گراف اصلی را برابر گراف مجاور می‌گذاریم و الگوریتم را ادامه می‌دهیم. برای اینکه استپ‌ها کمی بهتر شوند، به صورت تصادفی امتیاز همسایه‌ها را محاسبه نمی‌کنیم بلکه از همسایه‌ای شروع می‌کنیم که باقی‌مانده آن نسبت به باقی مانده‌ی دیگر راس‌ها کمتر مستقل است.
تضمینی وجود ندارد که این روش به بهترین گراف برسد [۶]. نویسندگان مقاله حتی ادعای مبنی بر این موضوع نیز انجام نمی‌دهند.

۳.۴ مقایسه‌ی الگوریتم‌های RESIT، Brute Force و GDS با سایر الگوریتم‌ها

مقاله‌ی [۶] به صورت مفصل الگوریتم‌های RESIT، Brute Force، GDS، PC GES و LiNGAM را مقایسه می‌کند.
در ادامه به بررسی این مقایسه و نقد آن می‌پردازیم. **ایراد بزرگ این مقایسه این است که بر روی داده‌هایی انجام شده که به صورت مصنوعی تولید شده‌اند.** معیار مورد استفاده‌ی ما برای مقایسه‌ی الگوریتم‌ها **Structural Hamming Distance (SHD)** است. همان‌طور که از اسم این معیار بر می‌آید، این معیار تعداد یال‌های اشتباه را می‌شمارد. وجود هر یال اضافه یا جهت‌دار به دست‌آوردن یال بی‌جهت نیز یک خطا در نظر گرفته می‌شود. در جدول‌های مقایسه‌ای که در بخش‌های بعد خواهند آمد، ردیف DAG مقایسه‌ی DAG واقعی با گراف به دست آمده و ردیف CDAG مقایسه‌ی CDAG و گراف حاصل است.

۱.۳.۴ حالت SCM خطی

در حالت خطی، ضرایب β_{jk} به صورت تصادفی با توزیع یکنواخت $[-2,-0.1] \cup [0.1,2]$ انتخاب می‌شوند. نویز های برونی مستقل هستند و با توزیع $K_j \cdot \mathrm{sign}(M_j) \cdot |M_j|^{\alpha_j}$ تولید می‌شوند که در آن $M_j \sim N(0,1)$، $K_j \sim U([0.1,0.5])$ و $\alpha_j \sim U([2,4])$. جدول (۲) مقایسه‌ی SHD این الگوریتم‌ها برای گراف‌ها با تعداد راس مختلف است. در این جدول، روش RAND

	GDS	BF	RESIT	LiNGAM	PC	CPC	GES	RAND
				$p=4, n=100$				
DAG	0.7 ± 0.9	0.6 ± 0.8	1.2 ± 1.3	1.9 ± 1.2	3.5 ± 1.5	3.6 ± 1.4	3.1 ± 1.7	4.4 ± 1.0
CPDAG	1.1 ± 1.5	0.9 ± 1.4	1.5 ± 1.7	2.4 ± 1.5	2.4 ± 1.7	2.3 ± 1.6	2.0 ± 2.0	4.3 ± 1.4
				$p=4, n=500$				
DAG	0.2 ± 0.6	0.1 ± 0.3	0.6 ± 0.8	0.5 ± 0.8	3.1 ± 1.4	3.2 ± 1.4	2.9 ± 1.6	4.1 ± 1.2
CPDAG	0.3 ± 0.9	0.2 ± 0.5	0.9 ± 1.3	0.8 ± 1.2	1.9 ± 1.8	1.6 ± 1.7	1.6 ± 1.9	3.9 ± 1.4
				$p=15, n=100$				
DAG	12.2 ± 5.3	–	25.2 ± 8.3	11.1 ± 3.7	13.0 ± 3.6	13.7 ± 3.7	12.7 ± 4.2	57.4 ± 26.4
CPDAG	13.2 ± 5.4	–	27.0 ± 8.5	12.4 ± 3.9	10.7 ± 3.5	10.8 ± 3.8	12.4 ± 4.9	58.5 ± 27.1
				$p=15, n=500$				
DAG	6.1 ± 6.4	–	51.2 ± 17.8	3.4 ± 2.8	10.2 ± 3.8	10.8 ± 4.2	8.7 ± 4.6	57.6 ± 24.2
CPDAG	6.8 ± 6.9	–	54.5 ± 18.5	4.5 ± 3.8	8.2 ± 4.6	7.5 ± 4.4	7.1 ± 5.6	58.9 ± 25.0

شکل ۲: SHD الگوریتم‌های مختلف به ازای گراف‌هایی با سایزهای مختلف - مدل خطی

به این نحو است که به صورت کاملاً تصادفی یک گراف انتخاب کرده و به داده‌ها نسبت می‌دهیم.

○ همان‌طور که انتظار می‌رفت، در گراف‌های کوچک، بررسی تمام گراف‌ها بهترین نتیجه را داشته ولی برای گراف‌های بزرگ، امکان اجرای این الگوریتم نبوده است.

○ در گراف‌های بزرگ‌تر روش LiNGAM بهترین نتیجه را دارد. نکته‌ی جالب این است که روش DAG نیز حاصلی شبیه به روش LiNGAM دارد که نشان می‌دهد روش GAS در کمینه‌های موضعی گیر نیفتاده است.

○ مطابق انتظار، الگوریتم REST در حالت خطی بسیار ناموفق عمل کرده است. این عدم موفقیت، به خصوص در گراف‌هایی با تعداد رأس زیاد بسیار شدید است. دلیل این امر این است که این الگوریتم، در این حالت در فاز اول خود یال‌های اضافی زیادی نگه می‌دارد که در فاز دوم قابل حذف نیستند.

۲.۳.۴ حالت SCM غیرخطی

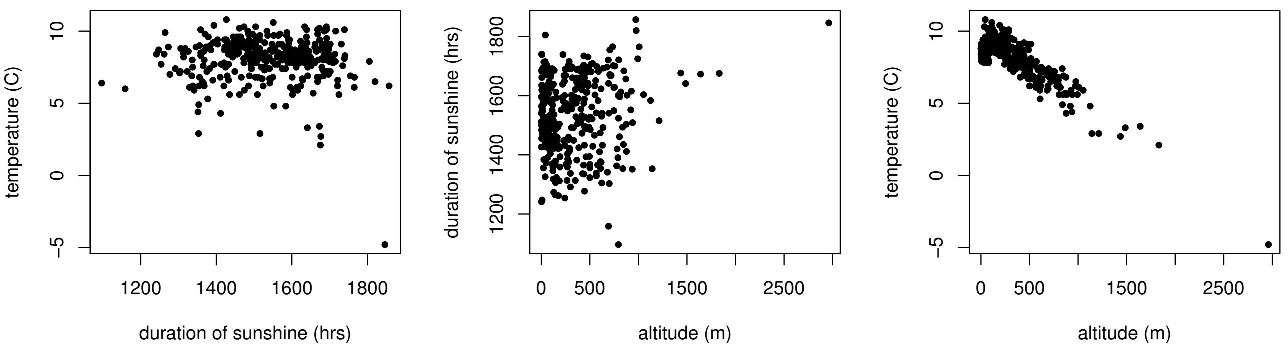
در این حالت، تابع به صورت تصادفی از یک فرآیند گاوسی با پهنای باند ۱ انتخاب شده و نویزها گاوسی هستند و واریانس آن تصادفی انتخاب شده است. در حالت غیرخطی نیز مشابه حالت خطی، در گراف‌های کوچک روش Brute Force بهترین نتیجه را می‌دهد. روش GDS نیز مشابه روش Brute Force عمل می‌کند. با بزرگتر شدن سایز گراف، دیگر روش Brute Force قابل استفاده نیست و روش GDS هم کارایی خود را از دست داده و گرفتار کمینه‌های موضعی می‌شود. به نظر می‌رسد که در گراف‌های بزرگتر الگوریتم RESIT موفق‌تر از سایر روش‌ها عمل می‌کند. بهتر بود که این مقاله برای گراف‌های بزرگتر نیز این الگوریتم‌ها را تست می‌کرد تا بتوانیم با اطمینان بیشتری گزاره‌ی بهتر بودن روش RESIT از سایر روش‌ها در گراف‌های بزرگ‌تر را مطرح کنیم.

	GDS	BF	RESIT	LiNGAM	PC	CPC	GES	RAND
				$p=4, n=100$				
DAG	1.5 ± 1.4	1.0 ± 1.0	1.7 ± 1.3	3.5 ± 1.2	3.5 ± 1.5	3.8 ± 1.4	3.5 ± 1.3	4.0 ± 1.3
CPDAG	1.7 ± 1.7	1.2 ± 1.4	2.0 ± 1.6	3.0 ± 1.4	2.9 ± 1.5	2.7 ± 1.4	3.4 ± 1.7	3.9 ± 1.4
				$p=4, n=500$				
DAG	0.5 ± 0.9	0.3 ± 0.5	0.8 ± 0.9	3.7 ± 1.2	3.5 ± 1.5	3.8 ± 1.5	3.3 ± 1.5	4.1 ± 1.2
CPDAG	0.6 ± 1.1	0.6 ± 1.0	1.0 ± 1.3	3.0 ± 1.7	3.1 ± 1.9	2.8 ± 1.8	3.4 ± 1.9	3.8 ± 1.6
				$p=15, n=100$				
DAG	14.3 ± 4.9	–	15.4 ± 5.7	15.4 ± 3.6	14.2 ± 3.5	15.5 ± 3.6	24.8 ± 6.3	56.8 ± 24.1
CPDAG	15.1 ± 5.4	–	16.5 ± 5.9	15.3 ± 4.0	13.3 ± 3.6	13.3 ± 4.0	26.4 ± 6.5	58.0 ± 24.7
				$p=15, n=500$				
DAG	13.0 ± 8.4	–	10.1 ± 5.7	21.4 ± 6.9	13.9 ± 4.5	15.1 ± 4.8	26.8 ± 8.5	56.1 ± 26.8
CPDAG	14.2 ± 9.2	–	11.3 ± 6.3	21.1 ± 7.3	13.7 ± 4.9	13.4 ± 5.1	28.6 ± 8.8	57.0 ± 27.3

شکل ۳: SHD الگوریتم‌های مختلف به ازای گراف‌هایی با سایزهای مختلف - مدل غیرخطی

۳.۳.۴ بررسی داده‌های واقعی؟!

در ادامه این مقاله تلاشی مذبوحانه(!) برای اثبات کارآمدی الگوریتم RESIT انجام می‌دهد.
داده‌ی مورد استفاده، سه متغیر مشاهده شده دارد: دمای شهر، ارتفاع شهر و مدت زمان تابش خورشید. این داده از ۳۴۹ مرکز هواشناسی آلمان به دست آمده است.



شکل ۴: نمودار پراکنش داده‌های مسئله

Method	Graph
LiNGAM	$T \to A$
PC	$T \to A \leftarrow DS$
CPC	$T \to A \leftarrow DS$
GES	Fully Connected
GDS	$T \leftarrow A \to DS$
BF	$T \leftarrow A \to DS$
RESIT	$T \leftarrow A \to DS$

شکل ۵: گراف پیشنهادی الگوریتم‌های مختلف

دیده می‌شود که الگوریتم‌های LiNGAM، (C)PC، GES گراف‌هایی را پیشنهاد می‌کنند که به وضوح غلطند! زیرا Intervention برو روی دمای شهر و یا زمان تابش خورشید در یک شهر، تاثیری بر روی ارتفاع آن نخواهد داشت. سه روش GES، BF و RESIT جواب‌های مشابهی می‌دهند.

به نظر من این گراف‌ها غلط هستند زیرا علاوه بر دو یالی که RESIT می‌یابد، چنین یالی نیز در گراف موجود باشد: $T \leftarrow DS$. نویسنده‌ی مقاله بیان می‌کند که گرافی که هر سه یال را دارد، در الگوریتم های GES و BF دومین بالاترین score را می‌یابد و ادعا می‌کند که این اشتباه به دلیل وجود confounder هایی از جنس متغیرهای جغرافیایی است.

به نظر من این تحلیل به هیچ عنوان تحلیل جامع و دقیقی برای مقایسه‌ی الگوریتم‌ها نیست. اولاً این مطالعه تنها بر روی یک دیتاست انجام شده است، حتی برای بدترین لگوریتم‌ها نیز می‌توان یک دیتاست یافت که خروجی آن الگوریتم جواب مناسبی باشد. دوماً جواب این الگوریتم خیلی جواب خوبی نیز نیست! سوماً اینکه **تعداد متغیرهای این دیتاست بسیار محدود است و برای آزمودن الگوریتمی که ادعای چندمتغیره بودن دارد و با پشتوانه‌ی داده‌های مصنوعی، در مورد آن این ادعا شده است که در گراف‌هایی با تعداد متغیر بالا از الگوریتم‌های گذشته بهتر عمل می‌کند، کافی نیست.**

۵ متغیرهای پنهان

تا کنون کار زیادی بر روی یافت متغیرهای پنهان در مدل‌های نویز جمعی انجام نشده است. یکی از معدود مقالات این حوزه، مقاله‌ی [۳] است. در این مقاله، الگوریتمی به نام ICAN معرفی می‌شود که در حالت دو بعدی، می‌تواند وجود یک متغیر پنهان را تشخیص دهد.

۱.۵ روش ساده‌لوحانه

در وهله‌ی برای یافتن متغیر پنهان، روش زیر به ذهن می رسد.

فرض کنید

$$\begin{cases} X = f(T) + N_X \\ Y = g(T) + N_Y \end{cases}$$

می‌توانیم به روشی مشابه روش‌های Dimension Reduction عمل کنیم. فرض کنید (X_k, Y_k) داده‌هایی باشند که در اختیار ما هستند. برای یک خم دلخواه $\mathbf{s}(t)$ تعریف کنید

$$\widehat{T}_k = \text{argmin}_{t\in[0,1]} ||(X_k, Y_k) - \mathbf{s}(t)||_2$$

قصد داریم $\widehat{\mathbf{s}}$ را بیابیم که $\sum_{k=1}^{n} ||(X_k, Y_k) - \mathbf{s}(\widehat{T}_k)||_2$. بعد از یافتن این خم، بررسی کنیم که آیا $(\widehat{T}_k)$ و

$$(N_x, N_y) = (X, Y) - \mathbf{s}(\widehat{T})$$

مستقل هستند یا خیر. این روش با وجود معقول بودن، در بسیاری از موارد ساده نیز بسیار بد عمل می‌کند.

Algorithm 1 Identifying Confounders using Additive Noise Models (ICAN)

Input: $(X_1, Y_1), \ldots, (X_n, Y_n)$ (normalized)

Initialization:
Fit a curve $\hat{\mathbf{s}}$ to the data that minimizes ℓ_2 distance: $\hat{\mathbf{s}} := \operatorname{argmin}_{\mathbf{s}\in\mathcal{S}} \sum_{k=1}^{n} \operatorname{dist}\big(\mathbf{s}, (X_k, Y_k)\big)$.

repeat
  Projection:
  $\hat{T} := \operatorname{argmin}_T \operatorname{DEP}(\hat{N}_X, \hat{N}_Y) + \operatorname{DEP}(\hat{N}_X, T) + \operatorname{DEP}(\hat{N}_Y, T)$ with $(\hat{N}_{X,k}, \hat{N}_{Y,k}) = (X_k, Y_k) - \hat{\mathbf{s}}(T_k)$
  if $\hat{N}_X \perp\!\!\!\perp \hat{N}_Y$ and $\hat{N}_X \perp\!\!\!\perp \hat{T}$ and $\hat{N}_Y \perp\!\!\!\perp \hat{T}$ **then**
    Output: $(\hat{T}_1, \ldots, \hat{T}_n)$, $\hat{u} = \hat{\mathbf{s}}_1$, $\hat{v} = \hat{\mathbf{s}}_2$, and $\frac{\mathbb{V}\mathrm{ar}\,\hat{N}_X}{\mathbb{V}\mathrm{ar}\,\hat{N}_Y}$.
    Break.
  end if
  Regression:
  Estimate $\hat{\mathbf{s}}$ by regression $(X, Y) = \hat{\mathbf{s}}(\hat{T}) + \hat{\mathbf{N}}$. Set $\hat{u} = \hat{\mathbf{s}}_1$, $\hat{v} = \hat{\mathbf{s}}_2$.
until K iterations

Output: Data cannot be fitted by a CAN model.

۲.۵ الگوریتم ICAN

برای رفع مشکلات الگوریتم ساده‌لوحانه، الگوریتم ICAN ارائه شده است. در این الگوریتم از خم فوق به عنوان یک نقطه‌ی شروع استفاده می‌کند ولی به دنبالی خمی می‌گردد که نویزها تا حد امکان از همدیگر مستقل شوند.

اگر جهت علّی از X به Y باشد، به راحتی دیده می‌شود که خمی که این الگوریتم باز می‌گرداند منجر به این نتیجه می‌شود که $\mathbf{var}\{\widehat{N}_x\} << \mathbf{var}\{\widehat{N}_y\}$ از این ویژگی برای این استفاده خواهیم کرد که وجود یک متغیر پنهان را تشخیص دهیم. شبه کد این الگوریتم در صفحه‌ی بعد آورده شده است.

مقاله اثباتی برای قابل‌شناسایی بودن، identifiability برای این الگوریتم ارائه می‌کند. نکته‌ی اصلی این مقاله این است که این قضیه تنها برای رژیمی که در آن واریانس نویز بسیار کم است ارائه شده و اثبات آن به شدت بر این فرض استوار است. داده‌های تجربی نشان می دهد که این روش می‌تواند در حالتی که واریانس نویز بزرگ است نیز موفق عمل کند، در نتیجه به نظر می‌رسد که قید کوچک بودن واریانس، تنها به دلیل ضعف اثبات است و انتظار می‌رود بتوان قضیه‌ی identifiability کلی‌ای، مشابه قضیه‌ی ارائه شده در [۲] برای آن ارائه داد. این تلاش تا کنون ناموفق باقی مانده است. از آن‌جا که اثبات تا حدی طولانی است، از آوردن آن خودداری کردیم.

۳.۵ بررسی تجربی الگوریتم

۱.۳.۵ داده‌ی مصنوعی

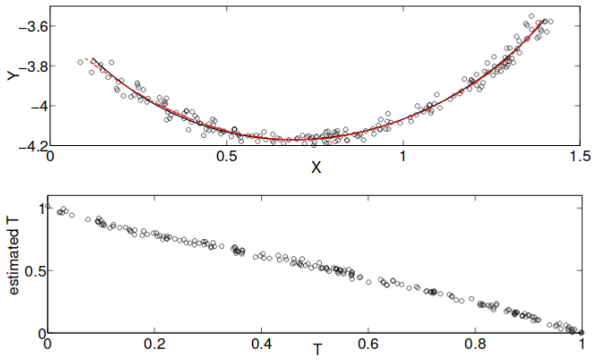
یک بررسی الگوریتم که در مقاله مطرح شده، به این شیوه بوده است: مدل

$$\begin{cases} X = f(T) + N_X \\ Y = g(T) + N_Y \end{cases}$$

را در نظر گرفته و توابع u و v را به صورت تصادفی به صورت ترکیب خطی از توابع گاوسی انتخاب می‌کند و ۲۰۰ نقطه با کمکم آن تولید می‌کند. توزیع نویز $U[0.035, 0.035]$ است. آزمون فرضیه‌ی استقلال مورد استفاده، آزمون HSIC است. نتیجه‌ی این آزمایش به شدت امیدوارکننده است. دیده می‌شود که الگوریتم به درستی قادر به تشخیص مقدار confounder است و تنها آن را در -1 ضرب کرده، یک پارامتری‌کردن دیگر خم. شکل (۶) بررسی نتایج این الگوریتم بر داده‌ی یادشده است.

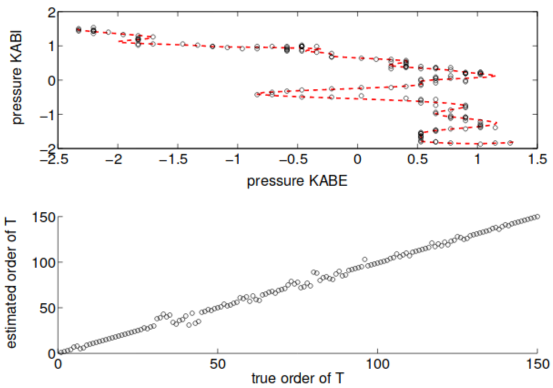
۲.۳.۵ داده‌ی واقعی

برای بررسی الگوریتم روی داده‌های واقعی، از داده‌ی معروف The Automated Surface Observations Systems (ASOS) استفاده شده است. این دیتاست متشکل از داده‌های ارسالی چند ایستگاه است که هر دقیقه داده‌های آب و هوا را



شکل ۶: بررسی مقدار متغیر پنهان در داده‌های مصنوعی که توسط الگوریتم تشخیص داده شده است و مقایسه‌ی آن با مقدار واقعی.

مخابره می‌کنند. از داده‌های فشار هوا که از ایستگاه‌های KABE و KABI در سال ۲۰۰۰ گرفته شده استفاده شده است. انتظار بر این است زمان یک confounder برای آن‌ها باشد. الگوریتم نیز یک confounder تشخیص می‌دهد و نتایج آن به طور کامل با این فرض هم‌خوانی دارند.



شکل ۷: بررسی مقدار متغیر پنهان در دیتاست واقعی که توسط الگوریتم تشخیص داده شده است و مقایسه‌ی آن با مقدار واقعی.

۶ سری‌های زمانی

یکی از مدل‌های موفق و مبتنی بر محدود کردن مجموعه‌ی توابع، در سری‌های زمانی، الگوریتم‌های TiMINo است.

تعریف ۱.۶. یک سری زمانی در نظر بگیرید. $X_t = (X_t^i)_{i\in V}$، به نحوی که توزیع‌های متناهی‌بعد آن تابع چگالی احتمال دارند. می‌گوییم این سری زمانی یک *TiMINo* است اگر وجود داشته باشد $p > 0$ و $\forall i \in V$ مجموعه‌هایی وجود دارد که $\mathbf{PA}i^0 \subseteq X^{V\setminus\{i\}}, \mathbf{PA}_i^k \subseteq X^V$، به نحوی که $\forall t$

$$X_t^i = f_i\big((\mathbf{PA}_i^p)_{t-p}, \ldots, (\mathbf{PA}_i^1)_{t-1}, (\mathbf{PA}_i^0)_t, N_t^i\big)\,, \qquad (۱۳)$$

که در آن N_t^i (مشترکاً) مستقل هستند و برای هر i، N_t^i در زمان *identically distributed* است. همچنین فرض بر این است که *Full Time Graph* دور ندارد.

این مدل می‌تواند به راحتی به یک مدل نویز جمع شونده تبدیل شود. می‌توان با اندکی تلاش، از قضیه‌ی اثبات شده در [۲] برای دو متغیر و تعمیم آن در [۶] به یک قضیه‌ی identifiability برای سری‌های زمانی TiMINo با فرض نویز جمعی رسید.
مقاله‌ی [۵] الگوریتمی ساده برای یافتن جهت‌های علّی در TiMINo ارائه می‌کند. این الگوریتم شباهت زیادی به الگوریتم RESIT دارد.

ایده‌ی این الگوریتم استفاده از یک روش رگرسیون برای fit کردن یک TiMINo بر داده‌ها است با این فرض که می‌خواهیم کمترین تعداد والد ممکن برای رسیدن به نویز مستقل را داشته باشیم. برای استفاده از این الگوریتم باید یک روش رگرسیون را مشخص کنیم. به طور خاص در این مقاله از روش‌های gam و GP استفاده شده است.

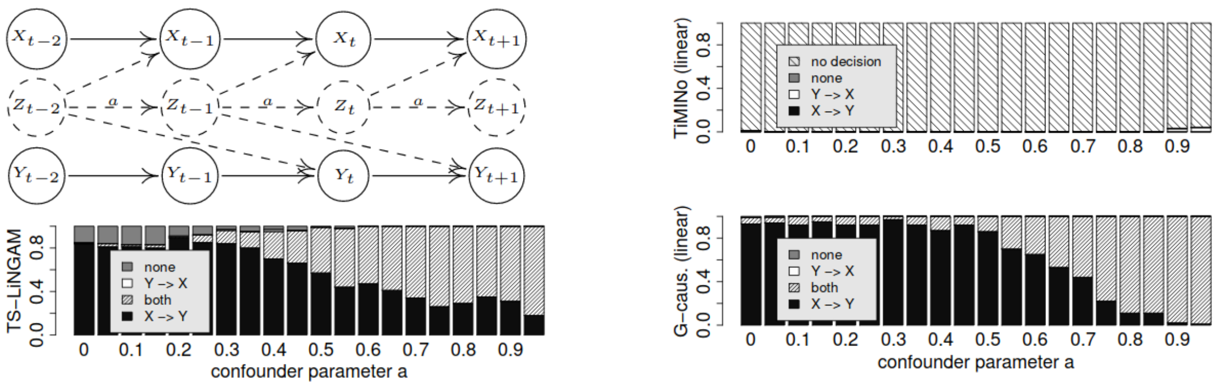
Algorithm 2 TiMINo causality

Input: Samples from a d-dimensional time series of length T: $(_1, \ldots, _T)$, maximal order p
$S := (1, \ldots, d)$
repeat
  for k in S **do**
    Fit TiMINo for X_t^k using $X_{t-p}^k, \ldots, X_{t-1}^k, X_{t-p}^i, \ldots, X_{t-1}^i, X_t^i$ for $i \in S \setminus \{k\}$
    Test if residuals are indep. of $X^i, i \in S$.
  end for
  Choose k^* to be the k with the weakest dependence. (If there is no k with independence, break and output: “I do not know - bad model fit”).
  $S := S \setminus \{k^*\}$
  $\mathrm{pa}(k^*) := S$
until length(S)=1
For all k remove all unnecessary parents.
Output: (pa(1), . . . , pa(d))

۱.۶ مزایای TiMINo بر سایر روش‌ها

TiMINo مزیت‌هایی بر سایر روش‌های تشخیص جهت علّی در سری‌های زمانی، به خصوص Granger Causality دارد که برخی از آنها را در ادامه خواهیم دید و مثال‌هایی از عملکرد روش های مختلف خواهیم زد.

- TiMINo به وجود متغیر های مخفی نسبت به Granger Causality مقاوم‌تر است. علّت این مقاومت این است که در صورتی که موفق به مستقل کردن نویزها نشود، بیان می کند که قادر به تشخیص جهت علّی نیست. به عنوان مثال به مثال زیر توجه کنید. در این گراف، z یک متغیر پنهان است. در این مثال با تغییر پارامتر a در گراف مشخص شده، نشان داده شده



شکل ۸: تاثیر متغیرهای پنهان در عملکرد الگوریتم‌های مختلف

است که الگوریتم‌های مختلف در چه کسری از مواقع، summary graph اشتباهی را بازگرداند. مشاهده می‌شود با افزایش a، Granger Causality و TS-LiNGAM هر دو جهت را به عنوان جهت درست علّی معرفی می‌کنند ولی TiMINo همواره بیان می‌کند که جهت قابل شناسایی نبود.

- TiMINo نیازی به فرض عدم وجود instantaneous effect ندارد.

به عنوان مثال سری‌های زمانی زیر را در نظر بگیرید. $X_t = A_1 \cdot X_{t-1} + N_{X,t}, W_t = A_2 \cdot W_{t-1} + A_3 \cdot X_t + N_{W,t}, Y_t = A_4 \cdot Y_{t-1} + A_5 \cdot W_{t-1} + N_{Y,t}, Z_t = A_6 \cdot Z_{t-1} + A_7 \cdot W_t + A_8 \cdot Y_{t-1} + N_{Z,t}$

$N_{\cdot,t} \sim 0.4 \cdot \mathcal{N}(0,1)$ و A_i از توزیع $\mathcal{U}([-0.8,-0.2] \cup [0.2,0.8])$ آمده است. در این بررسی، گراف $X \to W \to Z$ و $Y \to Z$ را به عنوان گراف صحیح در نظر گرفته‌ایم. موفقیت الگوریتم‌های مختلف برای تشخیص جهت علّی بدین صورت است.

DAG	lin. Granger	TiMINo-lin	TS-LiNGAM
correct	13%	83%	19%
wrong	87%	7%	81%
no dec.	0%	10%	0%

شکل ۹: تاثیر اثرات آنی بر عملکرد الگوریتم‌های مختلف

مشاهده می‌شود که TiMINo مطابق انتظارات، توانسته از دو الگوریتم دیگر بسیار بهتر عمل کند.

- ادعای بزرگ دیگری که در این مقاله مطرح شده اما به صورت تجربی آزمایش نشده این است که فرض کنید سیگنال‌ها از سورس‌های مختلف به ما می‌رسند و در نتیجه تاخیرهای زمانی متفاوتی در دریافت آن‌ها داریم. به دلیل این تاخیرها که از وجود آنها اطلاع نداریم، ممکن است جهت علّی معکوس در زمان نیز داشته باشیم. ادعا شده است که حتی در این حالت نیز TiMINo قادر به تشخیص summary graph است.

به طور کلی، مقاله‌ی [۵]، بررسی تجربی بسیار دقیقی روی TiMINo انجام می‌دهد. این مقاله، در مثال‌های واقعی زیادی الگوریتم TiMINo را آزمایش می‌کند. از جمله‌ی این آزمایش‌ها می‌توان به بررسی قیمت روزانه‌ی پنیر و کره پرداخت. انتظار این است که این دو سری زمانی، یک confounder مثل قیمت شیر داشته باشند. TiMINo در این حالت جهت علّی تشخیص نمی‌دهد ولی دو روش TS-LiNGAM و Granger Causality به اشتباه جهت‌هایی علّی ارائه می‌کنند.

۷ پیشنهاد برای کارهای پژوهشی آتی

به نظر من، در اولین قدم باید تلاشی برای یافتن یک قضیه‌ی identifiability کامل و بدون فرض کوچک بودن نویزها، که در [۳] انجام شده است، انجام شود. این کار می‌تواند سنگ‌بنای کارهای بزرگتری، مانند تعمیم نتایج آن و الگوریتم ICAN، به بعدهای بالاتر باشد.

مقاله‌ی [۶] بررسی تجربی دقیقی بر الگوریتم RESIT انجام نداده است. بررسی دقیق عملکرد این الگوریتم و مشاهده‌ی pithole های آن می‌تواند شهود بسیار بهتری برای ارائه‌های الگوریتم‌های یادگیری ساختار علّی دیگر به ما بدهد.

الگوریتم TiMINo به دلیل توانایی هندل کردن تاخیرهای زمانی در سیگنال‌هایی که قصد یافتن جهت‌های علّی آن‌را داریم، می‌تواند در بررسی‌های Fumctional Connectivity در مغز بسیار مفید باشد. به طور خاص در داده‌های fMRI مشکل وجود تاخیرهای مختلف زمانی مشاهده شده است.

الگوریتم یادگیری مناسب و بهینه‌ای برای Post Non Linear Models وجود ندارد در حالی که به نظر می‌رسد این مدل‌ها تا حد زیادی مطابق اندازه‌گیری‌های ما از طبیعت باشند، زیرا تابع بیرونی می‌تواند اندازه‌گیری ما را مدل کند. به نظر من این مدل‌ها، پتانسیل این را دارند توصیف‌گر دقیق‌تری از طبیعت باشند. همچنین تعمیم این مدل به مدل

$$\begin{cases} X = N_X \\ Y = f(g(X) + N_y) + N_* \end{cases}$$

با فرض استقلال N_x, N_y, N_* که در آن N_* مدل‌کننده نویز اضافه شده بر دیتای اندازه‌گیری شده است و ارائه‌ی یک قضیه‌ی identifiability احتمالی برای این مدل‌ها خالی از لطف نیست.

مراجع

[1] Chickering, D. M. A transformational characterization of equivalent bayesian network structures. in *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence* (1995), UAI'95, pp. 87–98.

[2] Hoyer, P. O., Janzing, D., Mooij, J. M., Peters, J., and Schölkopf, B. Nonlinear causal discovery with additive noise models. in *Advances in Neural Information Processing Systems 21*. 2009, pp. 689–696.

[3] Janzing, D., Peters, J., Mooij, J. M., and Schölkopf, B. Identifying confounders using additive noise models. in *UAI* (2009).

[4] Nowzohour, C., and Bühlmann, P. Score-based causal learning in additive noise models. *Statistics 50*, 3 (2016), 471–485.

[5] Peters, J., Janzing, D., and Schölkopf, B. Causal inference on time series using restricted structural equation models. in *Advances in Neural Information Processing Systems 26*. 2013, pp. 154–162.

[6] Peters, J., Mooij, J. M., Janzing, D., and Schölkopf, B. Causal discovery with continuous additive noise models. *J. Mach. Learn. Res. 15*, 1 (2014), 2009–2053.

[7] Zhang, K., and Hyvärinen, A. On the identifiability of the post-nonlinear causal model. in *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence* (2009), UAI '09, pp. 647–655.