یادگیری ساختار شبکه بیزی با محدودیت‌های جانبی

Andrew C. Li ACLI@UWATERLOO.CA
Peter van Beek VANBEEK@CS.UWATERLOO.CA
Cheriton School of Computer Science
University of Waterloo

چکیده

روش‌های ترکیبی برای یادگیری ساختار شبکه بیزی که هم داده‌های مشاهده شده و هم دانش تخصصی را در بر می‌گیرد در بسیاری از زمینه‌ها مهم است. مطالعات قبلی هر دو روش دقیق و تقریبی را برای یادگیری ساختار ارائه کرده‌اند. در این مقاله، ما یک روش تقریبی مبتنی بر جستجوی محلی پیشنهاد می‌کنیم که توانایی مدیریت موثر انواع محدودیت‌های دانش قبلی را دارد، از جمله یک طبقه مهم از محدودیت‌های اجدادی غیرقابل تجزیه است که دلالت بر علیت غیر مستقیم بین متغیرهای تصادفی دارند. در آزمایش‌ها ، روش تقریبی پیشنهادی ما می‌تواند به طور قابل توجهی بهتر از روش تقریبی موجود در یافتن راه‌حل‌های عملی در زمانی که محدودیت‌های سخت اعمال می‌شوند،عمل کند. رویکرد ما قادر به یافتن شبکه‌های بهینه نزدیک در حالی است که تقریباً ۵۰ متغیر تصادفی را در برمی‌گیرد. در مقابل، روش‌های دقیق قبلی قادر به کنترل بیش از بیست متغیر تصادفی نیستند. علاوه بر این، ما نشان می‌دهیم که وقتی دانش قبلی ادغام می‌شود، ما اغلب قادر به تولید یک شبکه بسیار نزدیک‌تر به شبکه واقعی(شبکه واقعیت زمینیground truth network) هستیم، به خصوص وقتی که مقدار داده محدود باشد.

1.معرفی

شبکه بیزین (BN): یک مدل گرافیکی احتمالاتی از توزیع احتمال با کاربردها در زمینه های مختلف است.

در عمل معمولاً به طور کامل توسط یک متخصص مشخص‌شده و یا مستقیماً از داده‌های مشاهده‌شده یاد گرفته می‌شود؛ با این حال، هر دو روش دارای اشکالات عمده هستند.

متخصصین معمولاً دامنه دانش کافی برای تولید شبکه کاملرا ندارند، به خصوص که تعداد متغیرهای تصادفی افزایش یابد. از سوی دیگر، وظیفه یادگیری ساختار BN مستقیماً از داده‌ها،یک مسئلهٔ hard-NP است، حتی برای رسیدن تقریبی به یک فاکتور منطقی داده‌ها اغلب محدود یا پرهزینه هستند.

1.1.محدودیت‌های دانش تخصصی

در بسیاری از زمینه‌ها, روش‌های هیبریدی برای یادگیری ساختار BN که هم دانش و هم داده‌های تخصصی را در بر می‌گیرد برای به دست آوردن نتایج برتر اعمال شده‌است; به عنوان مثال فلورس و همکاران (2011) و antal وهمکاران (2004) همکاران (2004) و Oyenوهمکاران (2016). این مطالعات از روش‌های جستجوی تقریبی با priors آموزنده استفاده می‌کنند, که به یک متخصص اجازه می‌دهد تا سطوح اعتماد را در محدودیت‌های مختلف مشخص کند. در مقابل, ما بر استفاده از دانش قبلی به عنوان محدودیت‌های سخت تمرکز می‌کنیم. روش ما بر رویMINOBS , یک الگوریتم جستجوی محلی مبتنی بر جستجوی محلی برای یادگیری ساختاری BN توسط Lee و vanbeek (2017) ساخته شده‌است. ما قادر به رسیدگی به انواع زیر از محدودیت‌های موجود در روش خود هستیم.

(1) وجود یک یال: ادعا میکند که یک یال جهتدار وجود دارد.همچنین کاربر می‌تواند مشخص کند که یک یال بدون جهت x- y در صورتی وجود دارد که جهت شناخته نشود.
(2) نبود یال: ادعا می کند که یال x- y وجود ندارد.
(3) قید و شرط: ادعا می کند x قبل از y در برخی مرتب کردن گره‌ها در شبکه BN دیده می‌شود.
(4) محدودیت اجدادی (مثبت): ادعا می کند که مسیری جهتدار بین x و y را وجود دارد.

ما به چنین محدودیت‌هایی به عنوان **محدودیت‌های جانبی** اشاره می‌کنیم. این محدودیت‌ها به اندازه کافی گویا هستند که به طور غیر مستقیم به برخی از محدودیت‌های رایج دیگر رسیدگی کنند؛مانند نبود یال بدون جهت (نبودیال xy یا yx)، سطوح علیت یا سببی (فقط یال ها از متغیر ردیف پایین به یک متغیر ردیف بالاتر مجاز هستند) و مشخص کردن گره‌های ریشه و برگ‌ها.

چالش اصلی کار ما, ترکیب محدودیت‌های اجدادی بود که نمی‌توان به صورت محلی (با اصلاح والدین یک گره) یا با هرس فضای مرتب‌سازی انجام داد. در فرآیند یادگیری ساختار به شدت مشکل پیچیده است. با این حال, محدودیت‌های اجدادی به چند دلیل مهم هستند:

اول, اگرچه محدودیت‌های اجدادی ضعیف‌تر از محدودیت‌های ذاتی یال هستند ولی وابستگی‌های مستقیم تنها به درجه بالایی از دامنه دانش معطوف هستند لذا متخصصین احتمال بیشتری برای ارائه اطلاعات مفید دارند مخصوصا در مواقعی که محدودیت‌های خاص کمتری نیز وجود داشته باشد.

ثانیا, محدودیت‌های اجدادی می‌توانند ادعا کنند که یک متغیر علت غیرمستقیم دیگری است, که شکل فراوانی از دانش تخصصی در بسیاری از حوزه‌ها دارد.

سوم این که این شبکه‌ها برای کشف دانش و یا ارائه دانش ساخته می‌شوند و باید همه وابستگی‌های علی شناخته‌شده را کدگذاری کنند. در آزمایش‌ها ما نشان می‌دهیم که اعمال محدودیت‌های اجدادی به شبکه واقعی(شبکه واقعیت زمینیground truth network)در قالب اظهارات علی آن‌ها نزدیک‌تر است. در واقع, یادگیری ساختار BN از داده‌ها به تنهایی نمی‌تواند بین علیت و همبستگی تمایز قائل شود.

1.2.نتایج

ما یک روش یادگیری ساختار ترکیبی جدید را ارائه می‌کنیم که می‌تواند دانش کارشناسی را به عنوان محدودیت‌های سخت, از جمله محدودیت‌های اجدادی تولید کند. در ارزیابی تجربی ما نشان می‌دهیم که روش ما قادر به یافتن راه‌حل‌های با کیفیت بالا برای مشکلات محدودیت هااست, در حالی که روش‌های دقیق قبلی قادر به کنترل بیش از بیست متغیر تصادفی نیستند. علاوه براین, ما اولین کسانی هستیم که با موفقیت محدودیت‌های اجدادی را با رویکرد یادگیری ساختار BN مبتنی بر جستجو ادغام می‌کنیم.

1.3.کارهای مرتبط

ما به طور خلاصه برخی کارهای مرتبط با الگوریتم‌های یادگیری ساختارشبکه های بیزین را مورد بحث قرار می‌دهیم.

در یک مقاله کلاسیک توسط Tsamardinos و همکارانش (۲۰۰۶)، از رویکرد greedy hill-climbing به طور موثر ودقیق برای یادگیری ساختارشبکه های بیزین استاندارد استفاده شد. یک الگوریتم جستجوی محلی دیگر توسط De و Castellano (۲۰۰۷) توانست دقیقاً به محدودیت‌های (۱) - (۳) رسیدگی کند. با این حال، این روش جستجوی محلی آنها مبتنی بر فضای DAGs است در حالی که روش پیشنهادی ما عمدتاً بر روی فضای Topologicalیا موضعی جستجو می‌کند.

یک رویکرد دقیق جدید توسط Chen وهمکارانش (2016) به عنوان **درخت EC** به طور خاص برای رسیدگی به محدودیت‌های اجدادی طراحی شده بود. درخت EC نیز می‌تواند محدودیت‌هایی را بر عدم وجود مسیر هدایت کند و راه‌حل‌هایی را با بهینگی تضمین‌شده پیدا کند که روش پیشنهادی ما نمی‌تواند.با این حال, روش آن‌ها در مقایسه با۲۰ متغیر تصادفی از لحاظ زمانی از امتیاز BDeu نیست.

یک روش دیگر GOBNILP توسط Bartlett و Cussens (2017)است که یک رویکرد دقیق مورد استفاده گسترده و قادر به رسیدگی به انواع زیادی از محدودیت‌ها است. Chen و همکارانش (2016) ثابت کردند که GOBNILP را میتوان برای رسیدگی به محدودیت‌های اجدادی هم انجام داد, اما نیازمند مرتبه بیشتری از درخت EC است.

روش CaMML که کشف تصادفی از طریق mml است وتوسط Korb و Nicholson(۲۰۱۰) ارائه شده ازجستجوی تصادفی براساس زنجیره مارکوف مونت کارلو استفاده می‌کند. این روش قادر به کنترل تمام محدودیت‌های (۱) - (۴) به جز محدودیت نبود یال است. همچنین می‌توانست ادعا کند که دو متغیر همبسته اند و تمامی محدودیت‌ها را میتوان به عنوان یک محدودیت نرم با تنظیم سطوح اعتماد مشخص نمود. ما CaMML را با روش پیشنهادی خود در ارزیابی تجربی وآزمایشات مقایسه می‌کنیم.

2.پیش نیاز ها

ما با مرور جزئیات مقدماتی در شبکه‌های بیز شروع می‌کنیم.یک شبکه های بیزین شامل n نود $V=\{v_1 ,\ldots, v_n\}$ که هر گره متغیر تصادفی است و یک توزیع احتمالی شرطی $P(v_i | par(v_i))$ برای هر متغیر تصادفی v_i است که در آن $par(v_i)$احتمال مجموعه والدین v_i در G است. یال‌های G از یک‌گره به دیگری نشان‌دهنده وابستگی‌های مستقیم هستند و ساختار کامل G تمام وابستگی‌های شرطی بین متغیرها را کد می‌کند.

در این مقاله ما بر روی روش امتیازوجستجو (score-search) برای یادگیری ساختار شبکه های بیزین تمرکز می‌کنیم. یک تابع امتیازدهی σ(G) انتخاب می‌شود که در آن هر DAG مانند G برروی n متغیررا براساس میزان تناسب با داده‌های مشاهده شده تعیین امتیازمی‌کند(ما فرض می‌کنیم یک امتیاز پایین‌تر بهتر است).یکی ازویژگی‌های معمول اکثرتوابع امتیازدهی مانند BIC/MDL و Deu، خاصیت تجزیه پذیری است به این معنی که امتیاز هر DAG مجموع نمرات محلی خود در هر گره است:

این روش معمول است که نمرات پیش فرض را برای هر متغیر تصادفی v_i و هر کدام از مجموعه نامزد های والد $p \subseteq 2^{V/\{v_i\}}$ محاسبه می کنند.شناسایی مجموعه پدر و مادر شامل هرس مجموعه پدر و مادر که هرگز ظاهر در راه حلهای بهینه ساز نمی شود(بیشتر در بخش 3 بحث شده است).ساختار بهینه سازی ،موظف به اختصاص یک پدر و مادر برای هر گره برای رسیدن به حداقل امتیاز شبکه بدون دور است.

به منظور گنجاندن دانش قبلی، ما می‌خواهیم مشکل یادگیری ساختار شبکه های بیزین را با محدودیت‌های بیشتر حل کنیم. ما رسماً این مشکل را مطرح می‌کنیم که بر روی این مقاله تمرکز کنیم.

تعریف ۱ (مشکل یادگیری ساختار شبکه بیزین) :

با استفاده از یک مجموعه داده $D = \{d_1, . . . , d_N\}$ ، که در آن هر d_i یک بردار از مقادیر گسسته از ویژگی‌ها و متغیرهای تصادفی Vاست، یک تابع امتیازدهی σ نشان می‌دهد چگونه یک ساختار کاندیدا توسط داده‌های مشاهده شده D پشتیبانی می‌شود، و محدودیت‌های بسیاری از اشکال زیر را می سنجد:

1. x→y (وجود یال جهت دار) یا x−y (وجود یال بدون جهت)
2. x↛y (نبود یال جهتدار)
3. x<y (محدودیت مرتب‌سازی موضعی Topological)
4. x⤳y (محدودیت اجدادی)

مساله یادگیری ساختار شبکه بیزی محدود معطوف به یافتن گراف جهتدار و بدون دور G برروی Vکه با توجه به موارد زیر امتیاز σ(G) را به حداقل برساند :

- برای هر گونه محدودیت x→y ، یال xy در G است.
- برای هر محدودیتx−y , یکی از یال‌ها ی xy یا yx در G است.
- برای هر محدودیت x↛y ، یال xy در G نیست.
- ترتیبی چون O وجود دارد که برای تمام محدودیت هایx<y که x قبل از y در O است وO یک مرتب سازی موضعی یا توپولوژیک برای G است.
- برای هر گونه محدودیت x⤳y ، یک مسیرجهتدار از x به y در G وجود دارد.

MINOBS جستجوی مبتنی بر مرتب سازی (Teyssier و Koller، ۲۰۰۵) را به یادگیری ساختار شبکه های بیزین اعمال می‌کند. با توجه به یک مرتب‌سازی موضعی یا Topologicalبرای مرتب کردن گره‌ها، پارامترهای بهینه سازگار با O را میتوان در زمان چند جمله‌ای پیدا کرد، با فرض اینکه مجموعه‌های والد اغلب محدود به یک اندازه ثابت در k هستند. این رویکرد در مرتبه زمانی O(n!) بر رویn متغیر جستجو را انجام می دهد، جاییکه در آن امتیاز یک مرتب سازی حداقل امتیاز یک شبکه های بیزین سازگار با این مرتب سازی است.

بقیه مقاله بر روی روش‌های مورد استفاده برای حل مساله یادگیری ساختاربا محدودیت متمرکز است.

3.شناسایی مجموعه والد

رویکردهای امتیازوجستجو برای یادگیری ساختار شبکه های بیزین روی استراتژی‌های هرس موثر تکیه می‌کنند تا تعداد مجموعه‌های والد کاندیدا برای هر متغیر کاهش یابد. مشکل شناسایی مجموعه والد، تعیین این است که کدام مجموعه والد در راه‌حل‌های بهینه ظاهر نمی‌شوند و ازاین رو لازم نیست بیشتر مورد توجه قرار گیرند. متاسفانه، قوانین هرس فعلی راه‌حل بهینه را تحت این فرض حفظ می‌کنند که ما در حال حل مساله با هیچ محدودیت جانبی روبرو نیستیم، بلکه می‌تواند در حضور محدودیت‌های کناری اشتباه باشد. برای مثال، استفاده گسترده از قانون pruning یا هرس کردن را در نظر بگیرید.

لم 2(Teyssier and Koller (2005) :

فرض کنید x_iیک نود یا گره باشد و Π‘وΠ را پتانسیل مجموعه والد های xi *که Π⊂ Π’ و* $\sigma(x_i,\Pi) < \sigma(x_i,\Pi')$ آنگاه *Π’ مجموعه والد* x_i *در DAG بهینه *G نیست.*

در مساله یادگیری ساختار شبکه های بیزین (نامحدود)، این قانون اجازه می دهد Π' ،که یک مجموعه والد برای x_iدر DAG G'است،هرس شود اگر Π مجموعه والد x_iبا امتیازکمتری از DAG G موجود باشد وDAG G رانیز به عنوان DAGبهینه انتخاب کند. با این حال، هنگامی که محدودیت‌ها در نظر گرفته می‌شوند، این قاعده دیگر برقرار نیست، چرا که ممکن استG' وجود یال یا محدودیت‌های اجدادی را برآورده کند که G انجام نمی‌دهد.بنابراین این تکنیک هرس نباید مورد استفاده قرار گیرد.

روش ما از رویکردهای جدیدی به شناسایی مجموعه والد برای مساله یادگیری ساختاری محدود شبکه های بیزین استفاده می‌کند. برای مثال، قوانین زیر از محدودیت‌ها برای حذف راه‌حل‌های غیرعملی استفاده می‌کنند.

(P1) ***اگر*** $x \to y$ ***آنگاه میتوان هر مجموعه والد y که شامل x نباشد را هرس کرد؛***

(P2) ***اگر*** $x \nleftrightarrow y$ ***آنگاه میتوان تمام مجموعه والدهای y که شامل x باشد را هرس کرد؛***

(P3) ***اگر*** $x < y$ ***آنگاه میتوان مجموعه والد x که شامل y باشد را هرس کرد؛***

(P4) ***اگر*** $x \nsim y$ ***و Π یک مجموعه والد y که تمام اعضای آن*** $\Pi_i < x$***،آنگاه میتوان Π را هرس کرد؛***

با استفاده از این قواعد برای هرس مجموعه‌های والد کاندیدا، هر DAG که از مجموعه والد باقی مانده ساخته شده‌باشد، باید تمام محدودیت‌های شکل $x \to y$ و $x \nleftrightarrow y$ را برآورده کند. اگر می‌خواهیم فقط DAGهای سازگار با برخی مرتب سازی ها را در نظر بگیریم، می‌توانیم به راحتی با استفاده از ***(P1)*** محدودیت x-y را به عنوان جهت یال از O درنظر بگیریم. قاعده بعدی نادرست است اما می‌تواند تعداد مجموعه‌های والد کاندید را به شدت کاهش دهد.

(P5) اگرΠوΠ' مجموعه های والد x_i که $\Pi \subset \Pi'$ و $\lambda \geq 1$یک ثابت که $\lambda\sigma(x_i,\Pi) < \sigma(x_i,\Pi')$ آنگاه میتوان Π' را هرس کرد.

توجه داشته باشید که این قاعده هرس،معادل با لم ۲ است اگر λ = ۱ انتخاب شود.ما مجموعه والد ***Π’*** را مورد هرس قرار می‌دهیم اگر امتیاز آن به طور قابل‌توجهی بدتر از یکی از زیرمجموعه‌های آن مانند ***Π*** نباشد،چرا که ‘Π بعید به نظر می‌رسد که در راه‌حل‌های بهینه برای مشکل یادگیری ساختاری شبکه های بیزین محدود ظاهرشود.انتخاب λ نیز مهم است؛چرا که هرس بیش از حد (Overpruning)می‌تواند برای کیفیت اثر منفی در کیفیت روش بگذارد در حالی که کم هرس کردن (Underpruning) نیز ناکارآمد است. ما یک مقدار اولیه از λ با استفاده از فرمول محاسبه می‌کنیم،

که در آن N تعداد متغیرهای تصادفی ، N تعداد نقاط داده، m_{anc} تعداد محدودیت‌های اجدادی ، و ω یک ثابت است.

این فرمول یک کلید اصلی را برآورده می‌کند. فرض کنید Gو H را ئاشته باشیم که DAGهایی با امتیاز بهینه و بدون محدودیت اجدادی باشد. انتظار داریم که G و H در ساختار و امتیاز نزدیک باشند اگر:

(i) **تعداد نقاط داده بزرگ باشد (هر دو رویکرد شبکه واقعی یا یک شبکه معادل داشته باشند) ویا**
(ii) **تعداد محدودیت‌های نیاکانی ناچیز باشد.**

بنابراین, در این موارد می‌خواهیم یک λ کوچک برای هرس مجموعه والد با امتیازات ضعیف ولی سخاوتمندانه داشته باشیم.

در رویکرد خود، ما جستجو را با مقدار بالاتر از λ مجدداًآغاز می‌کنیم در صورتی که راه‌حل‌های عملی را نمی‌توان یافت. ما همچنین مقدار ω را در روش تجربی خود مشخص می‌کنیم.

4.رویکرد Hill Climbing براساس محدودیت

در این بخش، ما در مورد نحوه جستجو برای یک راه‌حل بهینه با یک دستوریا مرتب سازی داده شده مانند O صحبت می‌کنیم که تمام محدودیت‌ها را برآورده می‌کند. به یاد می‌آورید که x→y ، x↛y ، و x–y همگی می‌توانند در شناسایی مجموعه والد به کار گرفته شوند.

ما فرض می‌کنیم که O با تمام محدودیت‌هایی که در بخش بعدی به کار می‌رود سازگار است. در اینجا ما نشان می‌دهیم که چگونه محدودیت‌های اجدادی را کنترل کنیم. برای راحتی، ما به ϕ(G) اشاره می‌کنیم که تعداد محدودیت‌های اجدادی برآورده شده توسط DAG G را نشان می دهد.

تعریف (همسایه والدینی):

دو DAG مانند G و G’ با گره‌های یکسان , همسایه والدینی منتسب هستند, اگر و تنها اگر مجموعه والد یک گره تنها بین G و G’ متفاوت باشد.

ابتدا در الگوریتم 2 برای مرتب سازی O وDAG داده شده ،که شامل محدودیت ها است،کمترین امتیاز را محاسبه شده وبه عنوان DAG G_0 همراه با مرتب سازی O به الگوریتم 1 بازگردانده می شودهنگامی که ما روش را در الگوریت 1 برروی G_0 اعمال می کنیم در هر مرحله از این رویکرد ما یک همسایه منتسب بهترمانندG’ را برای G انتخاب کرده و سپس G’ را با G جایگزین می کنیم.لذاگوییم *G’ بهبود یافته G است اگر وتنها اگر* $\varphi(G) \geq \varphi(G_0)$ *و* $\sigma(G) < \sigma(G_0)$.

*در اینجا هدف اولیه برآورده کردن محدودیت اجدادی وسپس بهینه سازی نمره است.*سپس یک لیست ازمجموعه والدهای کاندیدا بامرتب سازی صعودی نمره مرتب می شود.

سپس از روش Hill Climbing استفاده می‌کنیم. لیست مجموعه‌های والد کاندید با مرتب‌سازی صعودی مرتب می‌شود. برای هر مجموعه والد کاندیدای انتخابی p برای گره v, ما G’ را به عنوان همسایه والدینی G در نظر می‌گیریم.همانطور که تعداد مجموعه‌های والد کاندیدا می‌تواند بزرگ باشد, ما یک استراتژی انتخاب همسایه را اتخاذ می‌کنیم.با انجام یک جستجوی عمق-اول برای بررسی هر محدودیت اجدادی x⤳y در بدترین حالت، متاسفانه محاسبه ی $\varphi(G_0)$ می‌تواند بسیار پرهزینه باشد. با این حال, یک محدودیت x⤳yکه در G برآورده می‌شود باید در G_0 نیزبرآورده شود مگر اینکه v یکی از نوادگان x و یک جد y در G باشد. به علاوه, اگر یک محدودیت x⤳y در G برآورده نشده باشد,باید در G_0 برآورده شود در صورتی که vیک نسل از xو یکی از اجداد y باشد.به عبارت دیگر یک محدودیت اجدادی x⤳y که درG برآورده نشود درG’ برآورده میشود اگروتنها اگر نود v یک جد *y*ویک نسل از *x* باشد.از این رو, تعداد مشاهدات عمق اول را می‌توان با شرایطی کاهش داد که محدودیت‌های اجدادی برآورده شوند و نوادگان و اجداد گره‌های درگیر در محدودیت‌های اجدادی را در بر گیرد.

یک مشکل در واگذاری همسایگی والدینی این است که کوچک بوده و الگوریتم Hill Climbing می‌تواند در مینیمم محلی با کیفیت پایین درآن گیر کند. ما این مساله را با معرفی پیاده‌روی تصادفی (walkProb)و فهرست tabu حل می‌کنیم. وقتی Hill Climbing به یک حداقل محلی می‌رسد, با احتمال کمی به یک راه‌حل تصادفی ساده حرکت می‌کند. گره ای که مجموعه والد آن تحت‌تاثیر قرار می‌گیرند, پس از سه تکرار بعدی Hill Climbing ممنوع اعلام می‌شود, بنابراین تغییر بلافاصله برگردانده نمی‌شود. (در حالی که یک‌گره ممنوع باشد نمی‌تواند والد خود را تخصیص دهد)

Algorithm 1: hillClimbDAG(G, $\mathcal{O}$)

```
allParents ← allParentSets() ;            /* sorted by increasing score */
while true do
    for p in allParents do
        x ← child(p) ;
        if not tabu(x) and feasible(p, O) then
            G' ← G;
            G'.parentOf(x) ← p ;
            if G' improves G then
                G ← G' ;
                break ;
            end if
        end if
    end for
    if no improvement found then
        if random(0, 1) < walkProb then
            p ← random feasible parent set;
            x ← child(p) ;
            G.parentOf(x) ← p ;
            tabu x for 3 iterations ;
        else
            break ;
        end if
    end if
end while
return G ;
```

Algorithm 2: bestDAGForOrdering($\mathcal{O}$)

```
G0 ← initialDAG(O) ;
return hillClimbDAG(G0, O) ;
```

الگوریتم 1: شبه کد Hill Climbing

- ورودی : DAGبهینهٔ G_0 دارای بهترین امتیاز به همراه مرتب سازی ***Ordering (O)***
- ***عملیات : حذف مجموعه والدهای کاندیدای مضاعف(بخش اول)بر اساس امتیاز و آزمون درستی بهینه سراسری(بخش دوم)***
- *خروجی :* **DAG** بهینهٔ **G** دارای مجموعه والدهای بهینه

الگوریتم 2:.محاسبهٔ بهترین امتیازمرتب سازی

- *ورودی : DAG G**به همراه محدودیت ها و مرتب سازی*** **Orderring(O)**
- *عملیات*: محاسبه ی امتیاز تمام مجموعه های والد و **DAG**های آنها
- *خروجی*: **DAG**های بهینهٔ G_0دارای بهترین امتیازبه همراه مرتب سازی ***Ordering (O)***

5.جستجو براساس مرتب سازی *Ordering-based Search*

در این بخش ما بر روی Hill Climbing بر روی فضایمرتب سازی topological متمرکز می‌شویم. امتیاز یک مرتب سازی O حداقل مقدار σ (G) برایDAGهای G سازگار بامرتب سازی O و تمامی محدودیت‌ها است. همانطور که پیدا کردن حداقل امتیاز دشواراست، ما بهترین مقدار یافت‌شده توسط الگوریتم ۲ را به عنوان امتیازمرتب سازی O درنظرمی گیریم.

ما تقریباً همان استراتژی جستجو و همسایگی را که در MINOBS (لی و ون beek، ۲۰۱۷) که از همسایگی وارد شده و همسایگی مجاور آن استفاده می‌کنند، دنبال می‌کنیم. به بیان ساده، 'O یک *همسایه ورودی*(*insert neighbours*) از O است اگر از عناصرومرتب سازی O باشد و آن را به یک شاخص جدید اضافه کند.

به عنوان مثال { x_1 , x_2 , x_3 , x_4 } و { x_1 , x_4 , x_2 , x_3 } *همسایه ورودی* هستند چراکه یک همسایه مجاور ازیک مرتب سازی O ناشی از مبادله یک عنصر با عنصر قبلی آن است.به عنوان مثال دیگر { x_1 , x_2 , x_3 , x_4 }و { x_1 , x_2 , x_4 , x_3 }.

همسایه ورودی را می‌توان با استفاده از مجموعه‌ای از حرکت‌های مجاور پیمود.هرحرکت تبادل-مجاور(swap-adjacent) تنها بر انتخاب مجموعه والد برای دو عنصر تعویض شده در زمانی که هیچ محدودیت اضافی وجود ندارد، تاثیر می‌گذارد، اما هنگام وجود محدودیت‌های جانبی اثر گذارنیست.لذا ما استراتژی زیر را برای موثر بودن پیدا کردیم.

فرض کنید مرتب سازی $O=\{x_1,\ldots,x_i,x_{i+1},\ldots,x_n\}$ *داده شده و G به عنوان بهترین DAG* به همرا محدودیت هابدست آمده باشد.حال فرض کنید که عنصر i با عنصر i + ۱ جابجا شود و ما می‌خواهیم بهترین حالت ممکن با $O'=\{x_1,\ldots,x_{i+1},x_i,\ldots,x_n\}$ را بیابیم. ما دریافتیم که استفاده از G به عنوان نقطه شروع اولیه برای Hill Climbing بر روی *O'* زمان اجرا رابهبود می‌بخشد، چراکه G همه محدودیت‌های اجدادی را برآورده می‌کند واحتمالاً دارای کیفیت بالااست. اگرمجموعه والد های x_{i+1}و در G حاوی x_i باشد، G تحت *O'* پذیرفتنی نیست، اما ما این را با جایگزین کردن والدین x_{i+1} با مجموعه والد های با پایین‌ترین امتیاز ممکن تحت *O'* جایگزین می‌کنیم.

6.ارزیابی تجربی

ما در آزمایش‌ها خود روش پیشنهادی خود را با نشان دادن MINOBSx در برابر CaMML مقایسه می‌کنیم. CaMML برای مقایسه انتخاب شد چون به طور گسترده در زمینه‌های کاربردی استفاده می‌شود (مثلا فلورس و همکارانش(۲۰۱۱)، Kennett و همکارانش (۲۰۱۳)، Sesen و همکارانش (۲۰۱۳)از آن استفاده کردند)وهمچنین مبتنی بر رویکرد تصادفی امتیازوجستجو است. توجه داشتن به این مسعله مهم است که مساله یادگیری بدون محدودیت‌های اجدادی آسان‌تر است (اضافه کردن هر کدام از محدودیت‌های دیگر به طور کلی عملکرد را بهبود می‌بخشد.).بنابراین ما CaMML را برای مقایسه MINOBSx در برابر روش‌هایی که نمی‌توانستند محدودیت‌های اجدادی را کنترل کنند،انتخاب کردیم.متاسفانه کد منبع برای درخت EC دردسترس نبود.

نمونه‌های آزمایشی ما شامل همه شبکه‌های کوچک و متوسط در مخزن شبکه Bayesian است که بزرگ‌ترین آن‌ها barley (۴۸نود) است. ما به طور تصادفی ۶ مجموعه داده کوچک و ۶ مجموعه داده بزرگ را از توزیع‌های احتمالی مشترک شبکه‌های واقعیت زمینی این داده ها نمونه‌برداری کردیم.دو مجموعه از آزمایش‌ها اجرا شدند: (۱) تنها اعمال محدودیت‌های اجدادی و (۲) اعمال محدودیت‌های مختلف.

برای هر مجموعه‌ای از آزمایش‌ها، ما ۴ درصد ثابت را انتخاب کردیم و ۵ مجموعه از محدودیت را در آن درصد از شبکه واقعی(ground truth network)نمونه گیری کردیم. برای درصد p ما از محدودیت‌های موجود در روش زیر نمونه برداری کردیم:

- ما p% ازهمه یال‌های جهتدار‌در شبکه واقعی را انتخاب می‌کنیم. برای هر یال، x−y احتمال ½ را در نظر می‌گیریم، درغیر این صورت y - x را گزارش می‌کنیم.
- ما p% از تمام جفت‌ها (y,x) را که در شبکه واقعی هیچ یال‌های جهتداری بین آنها وجود ندارد، انتخاب وx↛y را گزارش می‌کنیم.
- ما p% از همه زوج‌های (y،x) را که یک مسیر جهتدار بین xوy در شبکه واقعی وجود دارد را انتخاب وx⤳y را گزارش می‌کنیم.
- ما از مرتب سازی توپولوژیک برای بازیابی نظم دلخواه از شبکه ی واقعی استفاده کرده و p% ازتمام $\binom{n}{2}$ محدودیت مرتب سازهای ممکنرا انتخابو x<y را برای O گزارش می‌کنیم.

وقتی MINOBSx و CaMML را مقایسه می‌کنیم، محدودیت‌های نبود یال x↛y را حذف می‌کنیم چون CaMML این محدودیت‌ها را قبول نمی‌کند. برای هر مجموعه داده و هر مجموعه محدودیت، ما هردو روش MINOBSx و CaMML را پیموده و مجموعه داده‌ها و محدودیت ها را به عنوان ورودی ارایه کردیم.

6.1 تنظیمات

تمام آزمایش‌ها بر روی یک هسته واحد از پردازنده اینتل Intel “Broadwell” E5-2683 v4 CPU @ 2.1 Gh با یک محدوده حافظه پردازشی ۵۱۲ مگابایت برای شبکه‌های کوچک و ۴ گیگابایت برای شبکه‌های متوسط اجرا شدند.

با توجه به محدود بودن فضای دیسک دخیره سازی, امتیاز مجموعه والدین با استفاده از فرمول کوکران و روش GOBNILP تا سه والدمحاسبه شد.زمان اجرای GOBNILP در زمان اجرای MINOBSx درنتایج آن محاسبه شده‌است. این روش برای 50 دوره در شبکه‌های کوچک و 10 دوره روی شبکه‌های متوسط اجرا شد. پارامتری ω نیزبا استفاده از نمونه‌های barle، child وسرطان تنظیم شده‌است.

CaMML تحت تنظیمات پیش‌فرض خود در جاوا اجرا شد. در حالی که MINOBSx در C++ اجرا می‌شود.تمامی محدودیت‌ها مربوط به CaMML با سطح اعتماد1.0 (محدودیت سخت) به جز محدودیت‌های اجدادی مشخص شدند، که به طور شگفت‌انگیز، هنگامی که سطح اعتماد برای محدودیت‌های اجدادی با تغییر اعتماد به 0.999 برای محدودیت‌های اجدادی مشخص شد، CaMML توانست برخی نمونه‌ها را کاملاً حل کند که در غیر این صورت قادر به یافتن راه‌حلی برای آن نبود.

6.2 نتایج

روش MINOBSx توانست راه حل های پذیرفتنی شامل تمام محدودیت ها را برای تمامی نمونه ها بدست آورد.

روش CaMML قادر بود به طور مداوم بیش‌ترین محدودیت را بر روی شبکه‌های کوچک داشته باشد, اما در همه موارد بسیار ضعیف عمل کرد. نتایج به طور صریح در جدول نشان‌داده شده‌است.

ما همچنین کیفیت شبکه‌های یادگیری(learn) شده توسط روش MINOBSx را دربرابر شبکه‌های واقعی زمین مقایسه می‌کنیم. ما کیفیت شبکه‌های یادگیری(learn) شده توسط روش CaMML را دربرابر شبکه‌های واقعی مقایسه نمی‌کنیم.چراکه این دو روش از توابع امتیازدهی مختلف استفاده می‌کنند، که می‌تواند تاثیر زیادی بر روی نتایج داشته باشد. برای مواردی که تنها محدودیت‌های اجدادی بود، ما به طور مستقیم نتایج تولید شده در آزمایش‌های گروه قبلی را تجزیه و تحلیل کردیم؛ با این حال، برای مواردی با محدودیت‌های مختلف، ما روش MINOBSx را بر روی مجموعه‌های محدودیت اجرا کردیم که شامل محدودیت نبود یال بودند. ما تعداد یال های گمشده،اضافی و معکوس(missing/extra/reverse)و فاصله مداخله سازه‌ای (SID) بین شبکه‌های واقعیت زمینی و شبکه یادگیری شذه را گزارش می‌کنیم (که تفاوت در علیت را اندازه‌گیری می‌کند).

instance	N	%	MINOBSx			CaMML		
			% feasible	% sat	t (seconds)	% feasible	% sat	t (seconds)
asia 8 variables 18 params	250	10 / 5	100 / 100	100 / 100	1.1 / 0.5	100 / 100	100 / 100	5.8 / 5.4
		25 / 10	100 / 100	100 / 100	1.3 / 0.5	100 / 100	100 / 100	6.3 / 5.5
		50 / 15	100 / 100	100 / 100	0.9 / 0.3	100 / 100	100 / 100	6.8 / 5.9
		100 / 20	100 / 100	100 / 100	0.5 / 0.2	100 / 100	100 / 100	8.3 / 5.7
	1000	10 / 5	100 / 100	100 / 100	0.9 / 0.4	100 / 100	100 / 100	5.5 / 5.0
		25 / 10	100 / 100	100 / 100	1.1 / 0.4	100 / 100	100 / 100	6.0 / 5.3
		50 / 15	100 / 100	100 / 100	0.7 / 0.3	100 / 96.7	100 / 99.7	6.3 / 5.6
		100 / 20	100 / 100	100 / 100	0.4 / 0.2	100 / 100	100 / 100	7.3 / 5.6
insurance 27 variables 984 params	500	10 / 5	100 / 100	100 / 100	180.5 / 104.9	50.0 / 70.0	95.9 / 99.0	439.5 / 325.3
		25 / 10	100 / 100	100 / 100	318.9 / 56.5	56.7 / 50.0	98.0 / 98.9	723.9 / 385.0
		50 / 15	100 / 100	100 / 100	328.7 / 52.8	30.0 / 33.3	98.5 / 98.1	1165.6 / 485.2
		100 / 20	100 / 100	100 / 100	292.5 / 37.4	0 / 53.3	98.1 / 99.6	2052.3 / 571.7
	2000	10 / 5	100 / 100	100 / 100	124.0 / 88.3	0 / 53.3	78.8 / 98.1	438.5 / 309.8
		25 / 10	100 / 100	100 / 100	236.3 / 49.6	16.7 / 23.3	92.4 / 96.8	748.6 / 393.2
		50 / 15	100 / 100	100 / 100	251.5 / 48.2	3.3 / 10.0	94.4 / 96.6	1175.2 / 487.9
		100 / 20	100 / 100	100 / 100	233.1 / 33.4	0 / 10.0	95.6 / 98.8	1956.8 / 571.1
barley 48 variables 114005 params	2000	10 / 5	100 / 100	100 / 100	2321.4 / 5866.8	0 / 0	70.3 / 89.3	19824.8 / 11666.1
		25 / 10	100 / 100	100 / 100	4228.9 / 2941.3	0 / 0	74.4 / 93.6	37034.3 / 14864.9
		50 / 15	100 / 100	100 / 100	7163.0 / 3518.9	0 / 0	73.9 / 95.0	70433.0 / 20339.2
		100 / 20	100 / 100	100 / 100	7246.6 / 1806.3	0 / 0	80.9 / 96.3	114036.7 / 22366.9
	8000	10 / 5	100 / 100	100 / 100	4761.1 / 6032.7	0 / 0	44.0 / 82.8	18092.6 / 10759.6
		25 / 10	100 / 100	100 / 100	4063.8 / 3620.0	0 / 0	52.5 / 91.8	35308.3 / 14477.5
		50 / 15	100 / 100	100 / 100	5137.8 / 3022.8	0 / 0	54.0 / 95.1	64893.5 / 18142.0
		100 / 20	100 / 100	100 / 100	5675.6 / 1638.8	0 / 0	53.4 / 95.1	111338.4 / 21184.4

جدول 1: نتایج عملکرد برای محدودیت‌های اجدادی تنها (اولین عدد در هر جفت) و با محدودیت‌های مختلف (عدد دوم در هر جفت)

- % :ثابت درصد مورد استفاده برای محدودیت نمونه ها،درصد مواردی که راه حل رضایت همه محدودیت های تحمیل شده
- feasible %:درصد قابل اجرا،درصد مواردی که راه حل آن توسط رضایت همه محدودیت ها تحمیل شده
- sat %:درصد محدودیت های برآورده شده توسط محدودیت های تحمیل شده
- t: زمان در حال اجرا و مورد نیاز برنامه
- N: تعداد مشاهدات
- سلول های برجسته نمایانگربرآورده نشدن تمامی محدودیت ها

instance	N	%	Missing	Extra	Reversed	SID	Score (BDeu)
asia	250	0*	1.5	1.7	1.0	12.2	0%
		10 / 5	1.4 / 1.4	1.6 / 1.6	0.7 / 0.9	10.3 / 11.1	0.0% / 0.0%
		25 / 10	1.2 / 1.4	1.8 / 1.6	0.5 / 0.6	7.2 / 8.9	0.1% / 0.1%
		50 / 15	1.0 / 0.8	2.0 / 1.4	0.3 / 0.4	4.3 / 4.4	0.2% / 0.3%
		100 / 20	0.5 / 0.7	1.7 / 1.1	0.0 / 0.2	1.8 / 3.9	0.3% / 0.3%
8 variables	1000	0*	0.8	0.3	1	9.0	0%
18 parameters		10 / 5	0.7 / 0.8	0.4 / 0.4	0.9 / 1.1	7.3 / 9.0	0.0% / 0.0%
		25 / 10	0.4 / 0.5	0.5 / 0.3	0.4 / 0.8	4.2 / 6.4	0.0% / 0.0%
		50 / 15	0.2 / 0.3	0.4 / 0.4	0.0 / 0.5	0.4 / 3.6	0.0% / 0.0%
		100 / 20	0.0 / 0.2	0.3 / 0.3	0.0 / 0.4	0 / 3.1	0.0% / 0.0%
child	500	0*	5.3	1.0	3.0	115.7	0%
		10 / 5	4.8 / 4.8	1.1 / 0.8	1.9 / 1.3	91.4 / 82.6	0.1% / 0.2%
		25 / 10	4.7 / 4.3	1.9 / 1.6	1.6 / 1.8	93.9 / 79.4	0.2% / 0.3%
		50 / 15	3.9 / 4.2	1.4 / 1.2	1.7 / 0.7	76.1 / 69.9	0.3% / 0.4%
		100 / 20	2.2 / 3.6	2.2 / 0.8	0.0 / 0.3	35.8 / 57.6	0.9% / 0.4%
20 variables	2000	0*	1.7	0.2	3.5	79.2	0%
230 parameters		10 / 5	1.5 / 1.3	0.2 / 0.2	0.5 / 0.1	26.0 / 20.8	0.0% / 0.0%
		25 / 10	0.7 / 1.0	0.3 / 0.3	0.3 / 0.5	12.1 / 18.0	0.0% / 0.0%
		50 / 15	0.7 / 1.0	0.3 / 0.3	0.5 / 0.0	12.2 / 16.2	0.0% / 0.0%
		100 / 20	0.2 / 1.0	0.3 / 0.1	0.0 / 0.0	2.5 / 16.6	0.1% / 0.0%
alarm	1000	0*	2.2	5.8	1.3	45.7	0%
		10 / 5	2.0 / 1.6	6.2 / 5.5	1.1 / 1.6	34.7 / 46.4	0.0% / 0.1%
		25 / 10	2.0 / 1.8	6.3 / 5.4	0.7 / 0.8	27.7 / 32.2	0.1% / 0.1%
		50 / 15	2.0 / 1.5	6.1 / 5.0	0.3 / 0.6	22.4 / 27.7	0.1% / 0.2%
		100 / 20	2.0 / 1.5	6.2 / 4.4	0.0 / 0.1	18.0 / 18.3	0.1% / 0.2%
37 variables	4000	0*	2.0	3.2	1.8	39.5	0%
509 parameters		10 / 5	2.0 / 1.6	4.6 / 4.5	0.6 / 1.2	24.9 / 38.8	0.0% / 0.0%
		25 / 10	2.0 / 1.8	4.5 / 4.3	0.3 / 0.6	20.1 / 28.0	0.0% / 0.0%
		50 / 15	2.0 / 1.5	4.1 / 4.2	0.0 / 0.4	18.0 / 22.3	0.0% / 0.1%
		100 / 20	1.7 / 1.4	4.2 / 3.7	0.0 / 0.3	12.3 / 20.4	0.0% / 0.1%
barley	2000	0*	32.3	8.2	9.7	949.5	0%
		10 / 5	33.3 / 31.7	14.4 / 12.2	5.1 / 4.4	792.9 / 756.7	0.6% / 1.4%
		25 / 10	31.9 / 31.0	15.6 / 13.8	5.3 / 4.8	802.6 / 741.5	0.9% / 1.7%
		50 / 15	31.0 / 27.5	17.4 / 11.3	3.0 / 3.2	699.4 / 666.9	1.4% / 2.9%
		100 / 20	30.2 / 26.4	19.5 / 11.8	0.0 / 1.5	619.3 / 628.1	2.4% / 3.9%
48 variables	8000	0*	25.5	3.7	9.7	794.7	0%
114005 parameters		10 / 5	25.6 / 24.6	8.9 / 8.0	5.3 / 5.2	636.6 / 625.4	0.3% / 0.8%
		25 / 10	25.0 / 24.0	10.4 / 8.8	5.5 / 4.3	644.4 / 584.7	0.4% / 0.9%
		50 / 15	22.1 / 20.5	9.2 / 7.6	3.3 / 1.9	541.7 / 487.4	0.6% / 1.4%
		100 / 20	20.8 / 19.3	12.0 / 7.0	0.0 / 1.6	457.2 / 507.2	0.9% / 1.8%

جدول 2: نتایج عملکرد برای محدودیت‌های اجدادی تنها (اولین عدد در هر جفت) و با محدودیت‌های مختلف (عدد دوم در هر جفت)

- نتایج عملکرد برای محدودیت های اجدادی (تعداد اول درهرجفت) و با محدودیت های مختلف(شماره دوم در هر جفت)
- ردیف *0 نشان داده شده زیر ستون٪ نمایانگر راه حل های مطلوب تولید شده توسط GOBNILP بدون محدودیت های جانبی

ما همچنین درصد اختلاف بین امتیاز BDeu شبکه‌های یادگیری(learn) شده و امتیاز بهینه تولید شده توسط GOBNILP در نمونه بدون محدودیت جانبی را گزارش می‌کنیم. نتایج برای ۴ شبکه از اندازه‌های مختلف در جدول ۲ نشان‌داده شده‌اند. با این حال, شبکه‌های دیگر نتایج مشابهی را ایجاد کرده‌اند.

از این نتایج, به نظر می‌رسد که به ویژه با افزایش تعداد متغیرها,روش MINOBSx بسیار قویتر از روش CaMML است. حتی وقتی روش CaMML به طور قابل‌توجهی زمان بیشتری را نسبت به زمان روش MINOBSx صرف می‌کند، اما اغلب قادر به یافتن راه‌حل‌های پذیرفتنی نیست. همچنین درصد اختلاف بین امتیازات BDeu در مساله همراه با محدودیت و مساله بدون محدودیت کوچک است و هرگز بیش از 4% نیست. از این رو در بدترین حالت, جواب‌های تولید شده توسط MINOBSx در محدوده خطای 4% قرار دارند.

هنگامی که ما تاثیر محدودیت‌های اجدادی بر کیفیت راه‌حل را تجزیه و تحلیل می‌کنیم، می‌بینیم که تعدادیال های گم‌شده و معکوس تمایل به کاهش دارد در حالی که تعداد یال های اضافی تغییر نمی‌کند یا بدتر می‌شود. با این حال، در برخی موارد تعداد یال های از دست رفته بهبود نیافته است، حتی هنگامی که ۱۰۰ % از محدودیت‌های اجدادی نیز اضافه شده‌است. به نظر می‌رسد که تعداد کمی از محدودیت‌های مختلف بهبود قابل‌توجه و پایداری را برای کم کردن یال های بیشتر ارایه می‌دهند. دلیل احتمالی این است که یال وجود دارد. و محدودیت‌های نبود یال می‌توانند مستقیماً یک یال یا یال اضافی را ثابت کنند در حالی که محدودیت‌های اجدادی مبهم هستند. اگر یک مسیر نادرست برای برآورده سازی محدودیت اجدادی مورد استفاده قرار گیرد, ممکن است باعث ایجاد یال‌های اضافی شود در حالی که تعداد یال‌های گم شده را بهبود نمی دهد. از سوی دیگر به نظر می‌رسد که محدودیت‌های اجدادی به تنهایی به طور قابل‌توجهی (SID) را بهبود می‌بخشد.درحالی که محدودیت‌های اجدادی منجر به دلایل غیر مستقیم می‌شوند و (SID) تفاوت در اظهارات علی را اندازه‌گیری می‌کند،این انتظار را نیز میتوان داشت.

یک دام یا نقص که ما یادداشت می‌کنیم این است که به دلیل حد پایین والد ۳، MINOBSx قادر به بازیابی شبکه واقعی(ground truth network) در برخی موارد نیست، در حالی کهCaMML صریحاً یک حد برای تعداد والدها را تعیین نمی‌کند. این امر در نتایج alarm که شامل یک نود با ۴ والد است، مشاهده می‌شود. با این حال، این به طور کلی مهم نیست چراکه مجموعه‌های والد بزرگ منجر به مدل‌های پیچیده می‌شوند که بعید به نطر می رسد که در راه‌حل‌های بهینه ظاهر شوند، به خصوص زمانی که مقدار داده کوچک باشد.

7.نتیجه‌گیری

ما یک روش جدید را برای ترکیب محدودیت‌های دانش قبلی (prior) به یک الگوریتم جستجوی محلی برای یادگیری ساختار شبکه بیزی, شامل محدودیت‌های مصنوعی غیر همگن ارائه می‌کنیم. در حالی که روش‌های دقیق قبلی می‌توانند به بیست متغیر تصادفی رسیدگی کنند, نشان می‌دهیم که روش جستجوی تصادفی ما قادر است تا به حدود پنجاه متغیر تصادفی در حین تولید شبکه‌های با کیفیت بالا،رسیدگی کند. در مقایسه با الگوریتم های مورد استفاده, CaMML یک نرم‌افزاراست که به طور گسترده توسط محققان در زمینه‌های کاربردی مورد استفاده قرار می‌گیرد.روش پیشنهادی ما عملکرد بسیار قوی و سازگار در شبکه ها،حتی با بیش از 20گره، را نشان می دهد.

برای کارهای آینده، تکنیک‌های مشابهی که در این مقاله ارائه شده‌اند،برای رسیدگی براستفاده موثر از محدودیت‌های غیرقابل تجزیه ،ممکن است مفید واقع شوند. یکی از این نمونه‌ها، d-جدایی (d-separation) است، که می‌تواند برای تضمین و اثبات روابط استقلال شرطی بین متغیرها استفاده شود.

7.1.سپاس گذاری ها

این تحقیق در بخشی از WestGrid ، Compute Canada پشتیبانی وبا یک جایزه NSERCUSRA حمایت شد.

منابع

مجموعه مقالات وتحقیقات یادگیری ماشین جلد ۷۲، ۲۲۵ - ۲۳۶، ۲۰۱۸ – ۲۰۱۸

P. Antal, G. Fannes, D. Timmerman, Y. Moreau, and B. De Moor. Using literature and data to learn BN as clinical models of ovarian tumors. Artificial Intelligence in Medicine, 30 (3):257–281, 2004.

M. Bartlett and J. Cussens. Integer linear programming for the BN structure learning problem. Artificial Intelligence, 244:258–271, 2017.

E. Y.-J.Chen,Y. Shen, A. Choi,and A.Darwiche. Learning BNs with ancestral constraints.In Advances in Neural Information Processing Systems, pages 2325–2333, 2016.

T.-T. Chen and S.-S. Leu. Fall risk assessment of cantilever bridge projects using BN.Safety science, 70:161–171, 2014.

D. M. Chickering, C. Meek, and D. Heckerman. Large-sample learning of BNs is NP-hard.
In Proceedings of the 19th Conference on Uncertainty in Artificial Intelligence, pages124–133, 2003.

S. Dasgupta. Learning polytrees. In Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence, pages 134–141, 1999.

L. M.de Campos and J.G.Castellano. BN learning using structural restrictions. International Journal of Approximate Reasoning, 45:233–254,2007.

M. J. Flores, A. E. Nicholson, A. Brunskill, K. B. Korb, and S. Mascaro. Incorporating expert knowledge when learning BN structure: a medical case study. Artificial intelligence in medicine, 53(3):181–204, 2011.

N. Friedman, M. Linial, I. Nachman, and D. Pe’er. Using BNs to analyze expression data. Journal of computational biology, 7(3-4):601–620, 2000.

R. Giordano, D. D’Agostino, C. Apollonio, N. Lamaddalena, and M. Vurro. Bayesian belief network to support conflict analysis for groundwater protection: the case of the Apulia region. Journal of environmental management, 115:136–146, 2013.

D. Heckerman, D. Geiger, and D. M. Chickering.The combination of knowledge and statistical data. Machine Learning, 20:197–243, 1995.

R. J. Kennett, K. B. Korb, and A. E. Nicholson. Seabreeze prediction using BNs. In Pacific-Asia Conference on Knowledge Discovery and Data Mining, pages 148–153.Springer,2001.

K. B. Korb and A. E. Nicholson. Bayesian Artificial Intelligence. CRC press, 2010.

C. Lee and P. van Beek. Metaheuristics for score-and-search BNs structure learning.In Proceedings of the 30th Canadian Conference on Artificial Intelligence, pages 129–141, 2017.Available as: LNCS 10233.

S.-S. Leu and C.-M. Chang. BN-based safety risk assessment for steel construction projects. Accident Analysis & Prevention, 54:122–133, 2013.

T.-Y. Ma, J. Y. Chow, and J. Xu. Causal structure learning for travel mode choice using structural restrictions and model averaging algorithm. Transportmetrica A: Transport Science, 13(4):299–325, 2017.

D. Oyen, B. Anderson, and C. M. Anderson-Cook. BNs with prior knowledge for malware phylogenetics. In AAAI Workshop: Artificial Intelligence for Cyber Security, 2016.

J. Peters and P. Buhlmann. Structural intervention distance for evaluating causal graphs.Neural computation, 27(3):771–799, 2015.

C. A. Pollino, O. Woodberry, A. Nicholson, K. Korb, and B. T. Hart. Parameterisation and evaluation of a BN for use in an ecological risk assessment. Environmental Modelling &Software, 22(8):1140–1152, 2007.

S¸. O. S¸ ahin, F. Ulengin, and B. Ulengin. A Bayesian causal map for inflation analysis: The case of Turkey. European Journal of Operational Research, 175(2):1268–1284, 2006.

G. Schwarz. Estimating the dimension of a model. The Annals of Statistics, 6:461–464, 1978.

M.B.Sesen, A.E.Nicholson, R.Banares-Alcantara, T.Kadir, and M.Brady.BNs clinical decision support in lung cancer care.PloS one,8(12):e82349, 2013.

M. Teyssier and D. Koller. Ordering-based search: A simple and effective algorithm for learning BNs. In Proceedings of the 21st Conference on Uncertainty in Artificial Intelligence, pages 548–549, 2005.

I. Tsamardinos, L. E. Brown, and C. F. Aliferis. The max-min hill-climbing BN structure learning. Machine learning, 65(1):31–78, 2006.