دانشگاه صنعتی امیرکبیر
(پلی تکنیک تهران)

گروه مستقل مهندسی رباتیک

گزارش تمرین اول درس مدل‌های احتمالاتی گرافی

استاد درس:

دکتر نیک آبادی

تدریس‌یار:

مهندس طاهرخانی

نام دانشجو:

نوید خزاعی

۹۲۱۳۵۰۰۸

اردیبهشت ۹۴

فهرست مطالب

۱ بخش نظری

۱.۱ سوال اول

قضیه زیر را اثبات کنید:

Theorem 1. Let $\mathcal{G}$ be a Bayesian Network structure over a set of random variables $\mathcal{X}$ and let P be a joint distribution over $\mathcal{X}$. If P factorizes according to $\mathcal{G}$, then $\mathcal{G}$ is an I-map for P.

پاسخ:

با توجه به این که P روی گراف $\mathcal{G}$ فاکتورایز می‌شود، پس می‌دانیم که:

$$P(X_1, \ldots, X_n) = \prod_{i=1}^{n} P(X_i \mid \mathrm{Pa}(X_i)), \tag{۱}$$

که در آن $\mathrm{Pa}(X_i)$ والدین متغیر تصادفی i م هستند. برای آن که نشان‌دهیم گراف $\mathcal{G}$ یک *I-map* برای P است، باید نشان‌دهیم $I(\mathcal{G}) \subseteq I(\mathcal{P})$. با توجه به آن‌چه در درس آمده بود، اگر مجموعه روابط استقلال $I(\mathcal{G})$ را به‌گونه‌ای با استفاده از نتایج فرمول ۱ نشان دهیم، یعنی این روابط از روابط استقلال توزیع مورد نظر استخراج شده‌اند پس زیرمجموعه‌ی آن نیز هستند. برای این کار، با توجه به این که مجموعه روابط استقلال در گراف ایجاب می‌کند که:

$$\{X_i \perp \mathrm{ND}(X_i) \mid \mathrm{Pa}(X_i); i = 1, \ldots, n\} \tag{۲}$$

که در آن $\mathrm{ND}(X_i)$ مجموعه‌ی غیرنسل[1] متغیر (X_i) است. اگر بتوانیم نشان‌دهیم که :

$$P(X_i \mid \mathrm{ND}(X_i)) = P(X_i \mid \mathrm{Pa}(X_i)) \tag{۳}$$

آن‌گاه رابطه‌ی استقلال ۲ برقرار است. برای این‌کار $P(X_i \mid \mathrm{ND}(X_i))$ را محاسبه می‌کنیم (نسل X_i با $\mathrm{D}(X_i)$ نشان داده شده‌است) :

$$\begin{aligned} P(X_i \mid \mathrm{ND}(X_i)) &= \frac{P(X_i, \mathrm{ND}(X_i))}{P(\mathrm{ND}(X_i))} = \frac{\sum_{\mathrm{D}(X_i)} P(X_1, \ldots, X_n)}{\sum_{X_i, \mathrm{D}(X_i)} P(X_1, \ldots, X_n)} \\ &= \frac{\sum_{\mathrm{D}(X_i)} \prod_{j=1}^{n} P(X_j \mid \mathrm{Pa}(X_j))}{\sum_{X_i, \mathrm{D}(X_i)} \prod_{j=1}^{n} P(X_j \mid \mathrm{Pa}(X_j))} \end{aligned} \tag{۴}$$

[1] Non-descendant

در ۴ در محاسبه‌ی سیگما، مواردی هستند که جزو نسل X_i نیستند، پس می‌توان آن‌ها را از سیگمای روی نسل آن خارج نمود. همچنین آن‌چه باقی می‌ماند:

$$\sum_{\mathrm{D}(X_i)} \prod_{X_j \in \mathrm{D}(X_i)} P(X_j \mid \mathrm{Pa}(X_j)) = ۱$$

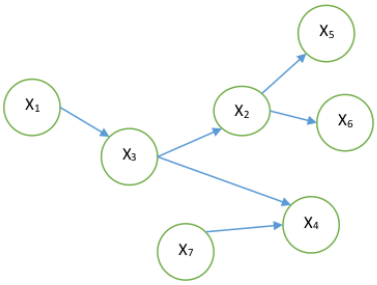
مشابه همین استدلال برای مخرج نیز پاسخ‌گو است، لذا خواهیم داشت:

$$\begin{aligned} &= \frac{\prod_{X_j \in (\mathrm{ND}(X_i) \cup X_i)} P(X_j \mid \mathrm{Pa}(X_j)) \times ۱}{\prod_{X_j \in \mathrm{ND}(X_i)} P(X_j \mid \mathrm{Pa}(X_j)) \times ۱} \\ &= P(X_i \mid \mathrm{Pa}(X_i)). \end{aligned} \tag{۵}$$

پس نشان‌دادیم ۳ و در نتیجه ۲ برقرار است، پس می‌توان نتیجه گرفت که $I(\mathcal{G}) \subseteq I(\mathcal{P})$، و حکم اثبات می‌شود.

۲.۱ سوال دوم

با توجه به شبکه‌ی شکل ۱ به سوالات پاسخ داده شده‌است.



شکل ۱: شکل سوال دوم

۱.۲.۱ توزیع توام

$$P(X_۱, \ldots, X_۷) = P(X_۱)P(X_۳ \mid X_۱)P(X_۲ \mid X_۳)P(X_۵ \mid X_۲)P(X_۶ \mid X_۲)P(X_۴ \mid X_۳, X_۷)P(X_۷)$$

۲.۲.۱ درستی و نادرستی

- $X_۱ \perp X_۵ \mid X_۲$: درست است. اگر $X_۲$ را بدانیم مسیر فعال بین $X_۱$ و $X_۵$ غیر فعال می‌شود و مستقل می‌شوند.

- $X_2 \perp X_7 \mid X_4$: نادرست است. در ساختار وی‌ـشکل بین X_7 و X_3 با دانستن X_4 وابستگی ایجاد می‌شود، و چون X_2 به پدرش X_3 وابسته است، با X_7 نیز در این شرایط وابسته است.

- $X_1 \perp X_5 \mid X_2$: درست است، چرا که دانستن X_3 ، استقلال بین X_2 و X_4 را ایجاد می‌کند و بنابر این از فرزندان X_2 یعنی X_5 نیز مستقل می‌شویم.

Markov Blanket for X_3 ۳.۲.۱

برای این کار ابتدا جهت‌ها را حذف می‌کنیم و سپس فساد[1] ها را حذف می‌کنیم. یعنی نباید گره‌ای باشد که والد یکی از فرزندان همین گره باشد، در صورت وجود به آن گره وصل می‌کنیم تا فساد از بین برود. به تمامی اتصالات موجود نیز وصل می‌کنیم، بنا بر این $MB(X_3) = \{X_1, X_2, X_4, X_7\}$ خواهد بود.

۲ بخش پیاده‌سازی

۱.۲ مقدمه

در این بخش باید با استفاده از پایگاه داده‌ی معرفی شده، چند مدل بیزین ارایه کنیم تا قادر به تشخیص بیماری قلبی باشد.

برای پیاده‌سازی این بخش، تصمیم به توسعه‌ی ابزار جدیدی برای کار با مدل‌های احتمالاتی گرافیکی گرفتیم، که کاملا با ابزارهای تحت خط فرمان سیستم‌عامل لینوکس و پوسته‌ی Bash کار می‌کند. این‌گونه ابزارها، معمولا برای استفاده در مواردی که فایل‌های حجیمی نیاز به پردازش دارند و می‌خواهیم درگیر مسایل مدیریت حافظه و Caching نشویم کاربرد دارند. همچنین، دید مبتنی بر فایل در بسیاری موارد به کمک توسعه‌دهنده می‌آید و توسعه‌ی ابزارها تحت قالب Shell Script، انعطاف بسیار زیادی را برای برنامه‌نویس فراهم می‌کند. این گونه ابزارها معمولا در فازهای یادگیری به دلیل نیاز به نوشتن مفرط در فایل کند هستند، به‌گونه‌ای که ابزار تولید شده در این تمرین عملا کارایی برای آموزش روی داده‌ها ندارد که در تمرینات آتی به رفع این ایرادها می‌پردازیم. در عوض، از آن‌جا که

[1] Immorality

به کل مساله‌ی آموزش و تست به دید مساله‌ی پردازش متن نگاه کرده‌ایم، در فاز تست سرعت پاسخ‌گویی بسیار بالا خواهد بود چرا که ابزارهای پروژه‌ی GNU مانند grep که برای جست‌وجو استفاده می‌شوند و یا awk، sed و cut که به دفعات بسیاری از آن‌ها استفاده نموده‌ایم، کارایی به شدت بالایی در پردازش فایل‌های عظیم دارند. به این ترتیب در پردازش پایگاه‌های داده‌ی بیش از حد بزرگ و زمانی که CPD ها بزرگ می‌شوند و نیاز به جست‌وجوی موثر داریم، کارا خواهند بود. همچنین انعطاف در تعیین نوع داده‌ای ویژگی‌ها و متغیرها از جمله مزیات دیگر این ابزار است و ورودی و خروجی‌های تولید شده به آسانی با دیگر ابزارهای تحت خط فرمان لینوکس قابل استفاده هستند.

در ادامه پس از توضیح چگونگی استفاده از پایگاه داده، به بررسی ابزار توسعه داده‌شده و پاسخ سوالات تمرین می‌پردازیم. ☹

۲.۲ پایگاه داده و پردازش ها

با دریافت پایگاه داده‌ی شهر کلیولوند[۱] از سایت معرفی شده [۲] ، به بررسی ستون‌های ویژگی و پردازش‌های مورد نیاز پرداختیم. آزمایش‌ها بر روی نسخه‌ی تجدیدنظر شده نیز انجام گرفتند [۳] . از جمله گسسته‌سازی هر ویژگی با توجه به مفاهیم آن انجام گرفت، در زیر توضیحات گسسته‌سازی آورده شده است:

۱. **سن**: این متغیر با مقادیر گسسته‌ی ۲،۳،۴،۵،۶ و ۷ نمایش داده‌شد، تا نشان‌دهنده‌ی رقم دهگان نمونه‌ی مورد نظر باشد.

۲. **کلسترول**: مقادیر کمتر از ۲۰۰ (سالم) ، بین ۲۰۰ تا ۲۴۰ (مرز سلامتی) و بالاتر از ۲۴۰ (خطرناک) به ترتیب با اعداد ۱، ۲ و ۳ نشان داده شدند[۴].

۳. **فشار خون در استراحت**: به ترتیب در بازه‌های ۱۱۹_۹۰، ۱۳۹_۱۲۰، ۱۵۹_۱۴۰، ۱۷۹_۱۶۰ و بیش از ۱۸۰ در نظر گرفته شدند و اعداد یک تا پنج به آن‌ها اختصاص داده‌شد [۵] .

---

[۱] Cleveland
[۲] http://archive.ics.uci.edu/ml/datasets/Heart+Disease
[۳] http://archive.ics.uci.edu/ml/machine-learning-databases/statlog/heart/
[۴] http://en.wikipedia.org/wiki/Cholesterol
[۵] http://en.wikipedia.org/wiki/Blood_pressure

۴. **Old Peak**: با توجه به بازه‌های موجود در پایگاه، بازه‌های صفر تا یک، یک تا دو، دو تا سه، سه تا پنج و پنج تا ۷ در نظر گرفته شدند.

۵. **بیشینه ضربان قلب**: برای محاسبه‌ی این ویژگی، سن طبق فرمول‌های زیادی از جمله :

$$۲۰۸ - (۰/۷ * \text{سن})$$

تاثیر دارد [۱] . با محاسبه‌ی این اعداد برای سنین ۲۰ تا هفتاد سالگی، به اعداد نرمال برای این ویژگی رسیدیم و بازه‌های زیر را در نظر گرفتیم :

کمتر از ۱۵۹، تا ۱۶۶، تا ۱۷۳، تا ۱۸۰، تا ۱۸۷، تا ۱۹۴، و بالاتر از ۱۹۴ که هفت دسته را شامل می‌شود.

۶. **حضور بیماری**: در پایگاه‌داده‌ی قدیمی‌تر، مقادیر غیر صفر و ناشناخته را دو در نظر گرفتیم که نشان وجود بیماری است و یک عدم وجود آن.

به جای مقادیر ناشناخته، مقادیر میانگین هر ستون را قرار دادیم، این اعمال با دو اسکریپت توسعه داده شده تحت نام‌های changeUnknownWithMean.sh و quantizeQuery.sh (که برای یک خط کار می‌کند و روی کل خطوط فایل داده فراخوانی می‌شود) انجام می‌گیرد. همچنین برای اطمینان از این که سطرهای پایگاه داده هم طول هستند می‌توانید از اسکریپت checkLines.sh استفاده کنید.

۳.۲ ابزارهای توسعه‌داده شده و ساختار

در این ابزار، ابتدا نیاز دارید با دو اسکریپت معرفی شده در بخش قبل فایل مورد نظر را بسازید، از این پس به نام پایگاه داده از این فایل گسسته‌سازی شده یاد می‌کنیم و ابزارهای توسعه داده شده‌ی مرتبط نیز در بخش قبل معرفی شد. سپس باید متغیرهای تصادفی خود را و این که هر کدام در چه ستونی از پایگاه داده هستند در یک فایل مانند vars.define مانند زیر با دو نقطه و در هر خط معرفی کنید:

age:1

sex:2

chest:3

[۱] http://en.wikipedia.org/wiki/Heart_rate#Maximum_heart_rate

RBP:4

chol:5

FBS:6

RC:7

MHR:8

EIA:9

oldPak:10

slope:11

vessels:12

thal:13

HD:14

دقت کنید که این فایل باید به ترتیب عددی ستون‌ها از بالا به پایین مرتب شده باشد در غیر این صورت الگوریتم‌های آتی دچار مشکل خواهد شد.

رعایت تمامی نکات دستوری بالا و آنچه در ادامه خواهد آمد لازم است چرا که تمامی پردازش‌ها با این فرض انجام شده است.

سپس نیاز دارید بازه‌های داده‌ها را استخراج کنید که در اصل مقادیر مجاز برای داده‌ها می‌باشند. این کار را می‌توانید با اسکریپت توسعه‌داده شده با نام extractRanges.sh انجام دهید، مانند:

./extractRanges.sh ../Dataset/cleveland-bin.quantized ./Sample/vars.define ./Sample/newVars/

که پارامتر اول پایگاه داده، دوم فایل توصیف متغیرها و سومی مکانی برای ساخت بازه‌های متغیرها است. این نحوه آدرس‌دهی را در اکثر اسکریپت‌های توسعه‌داده شده خواهید دید. پس از اجرا، در آدرسی که برای ساخت بازه‌های متغیرها دادیم، فایل‌هایی با پسوند var. ساخته می‌شود که در آن مقادیر مجاز با کاما از هم جدا شده‌اند. به نفع کاربر است که در مقادیر متغیرها از فاصله یا wildcharها مانند ستاره استفاده نکند چرا که در عملیات دستوری ممکن است خطاهای ناخواسته ایجاد شود.

پس از این، نوبت به ساخت معماری شبکه‌ی مورد نظر می‌رسد. برای این کار ساختار

ساده‌ای در نظر گرفتیم، در یک فایل به ازای هر یال پدر به فرزند، یک زوج متغیر با فاصله از هم جدا می‌شوند، مانند زیر که نمونه یک شبکه‌ی Naïve Bayes برای ۹ متغیر ویژگی و یک متغیر کلاس بیماری است که با نام myBn.bn ذخیره شده است. دقت کنید پسوند فایل‌های معرفی شده مهم نمی‌باشد.

HD age

HD sex

HD chest

HD RBP

HD chol

HD FBS

HD RC

HD MHR

HD EIA

HD oldPak

HD slope

HD vessels

HD thal

برای آموزش شبکه‌ی معرفی شده، اسکریپت train.sh را به صورت زیر صدا کنید (اسامی برای مثال آورده شده‌اند):

./train.sh Sample/vars.define ../Dataset/cleveland-bin.quantized ./Sample/naiveBayes.bn ./Sample/varsNaive/

این اسکریپت، از ماژول‌های زیرین کوچک‌تری استفاده می‌کند که هر کدام وظایف اتمیک در مدل‌های احتمالاتی را دارند، برای نمونه ساخت جداول CPD یکی از این ماژول‌ها است و یا با اسکریپت probQuery.h می‌توان مستقیما احتمال یک پیشامد را با نام متغیرها و مقادیر آن‌ها در جدول‌های احتمال توام جست‌وجو کرد. همچنین پس از ساخت CPD ها می‌توان یک احتمال شرطی را با اسکریپت cpdQuery.sh از پایگاه سوال کرد.

پس از اجرای اسکریپت train.sh دو پوشه در مسیر جاری به نام‌های joints و CPD

ساخته می‌شوند. فایل‌های پوشه‌ی اول جداول توام ساخته شده از روی پایگاه هستند که پسوند joint. دارند و در پوشه دوم، CPD ها با پسوند visualized قابل دسترسی هستند، هرچند سیستم طراحی شده از جداول با پسوند cpd. استفاده می‌کند که به دلیل ذخیره‌سازی خطی، برای خواندن در خط فرمان و جست‌وجو ساده‌تر هستند. یکی از ابزارهای دیگر، ابزاری سریع برای تولید انواع ترکیب‌های متغیرهای ورودی است که combineVars.sh نام دارد و برای ساخت مقادیر ممکن برای سطرهای جداول توام و شرطی به کار می‌رود.

اسکریپت leaveOneOut.sh نیز از اسکریپت makeNthQuery.sh که عمل یک خط خاص را جدا کردن و کوئری ساختن از آن را برایمان انجام می‌دهد، و fullObserveQuery.sh یک کوئری ساده برای کلاس به شرط مشاهدات (همه متغیرها) را انجام می‌دهد. این کار برای نایبو بیز به سادگی از فرمول :

$$P(Y = y \mid E = e) = \frac{P(E \mid Y) \cdot P(Y)}{\sum_y P(E \mid Y_y) P(Y_y)} \qquad (۶)$$

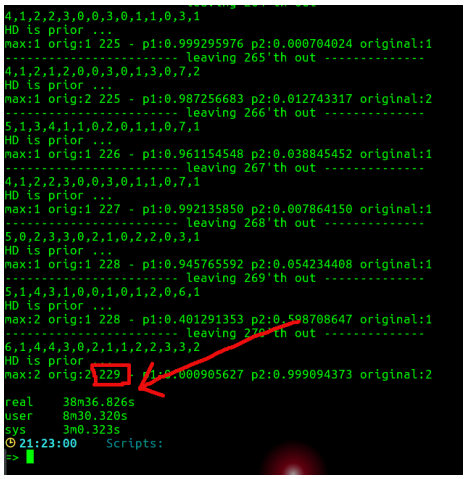
محاسبه‌ی این مقدار با توجه به مستقل بودن به شرط دانستن Y در نایبو بیز، در اسکریپت fullObserveQuery.sh پیاده‌سازی شده‌است.

متاسفانه این ابزار هنوز برای همه‌ی مدل‌ها پاسخ‌گو نیست چرا که نیاز به پیاده‌سازی روش‌های استنتاج دارد و پیچیدگی‌های فنی بسیار زیاد Shell Script روند کار را به شدت کند می‌کند. برای این تمرین در حد آموزش و تست نایبو بیز نتایج درست است ولی در شبکه‌های دیگر درست کار نمی‌کند که از آوردن نتایج آن خودداری کرده‌ایم، چرا که تا توسعه نیافتن ابزار استنتاج ارزشی نخواهد داشت. از مزیت‌های این ابزار انجام محاسبات تا ۲۱ رقم اعشار است که در زیاد بودن تعداد پارامترها به ما کمک می‌کند، هر چند برای سهولت بیشتر در خواندن اعداد، این ارقام را تا نه رقم اعشار چاپ کرده‌ایم.

۴.۲ نتایج

نتایج اجرای Leave one out برای دو پایگاه داده شهر کلیولند بررسی شدند. در ابتدا هر دو پایگاه در یک شبکه نایبو بیز با ۱۴ پارامتر تست شدند که نتایج در زیر آورده شده است. تعداد تشخیص‌های درست را در تصویر مشاهده می‌کنید.

مشاهده می‌کنید که به دلیل استفاده از حلقه‌های متعدد در اسکریپت‌ها، زمان اجرای



شکل ۲: نتایج برای پایگاه‌داده نسخه جدید با ۱۴ ویژگی ـ نایبو بیز

Leave one out عملا غیر قابل کاربرد است (نزدیک به ۳۰ دقیقه) که ضعف اصلی ابزار توسعه داده شده می‌باشد. هرچند انتظار می‌رود با استفاده از یک فایل سیستم tmpfs که از حافظه اصلی RAM برای ذخیره‌سازی استفاده می‌کند کمی بهبود یابد اما امیدی به کارایی بالا در استنتاج‌های پیچیده‌تر نیست.

برای پایگاه داده‌ی نسخه‌ی جدید، با ۱۴ متغیر، دقت ۰٫۸۴۸۱۴۸۱۴۸۱۴۸۱۴۸۱۴۸۱۴ و با پایگاه قدیمی که مقادیر ناقص آن با میانگین ستون‌ها جاگزین شد دقت ۰٫۹۲۵۹۲۵۹۲۵۹۲۵۹۲۵۹۲ حاصل شد. برای آزمون، با توجه به جداول CPD در همین مدل‌ها، آن دسته ویژگی‌هایی که احتمال‌هایشان یک‌نواخت تر توزیع شده بود و اعدادشان برای دسته‌های متفاوت نزدیک به هم بود، کنار گذاشته شد (متغیرهای RBP,FBS,RC,oldPeak) و دوباره تست انجام گرفت. دقت‌ها به ترتیب برای پایگاه جدید با اندکی کاهش به ۰٫۸۴۰۷۴۰۷۴۰۷۴۰۷۴۰۷۴۰۷۴ رسید ولی برای پایگاه قدیمی ثابت ماند. بنابراین استفاده از پایگاه داده‌ی قدیمی دقت بیشتری به دست داد. هرچند روش درست حل این مساله، ترتیب اثر دادن ماتریس هزینه برای تشخیص غلط بیماری است که باید به جای بیشترین احتمال، مجموع هزینه (احتمال ضرب در هزینه) روی همه‌ی پیش‌بینی‌های موجود محاسبه شود و آن پیش‌بینی که این هزینه را کمینه می‌کند انتخاب شود که متاسفانه در این فاز از تمرین به انجام نرسید. با فرض دانستن slope=2 تغییری که در جدول‌های احتمال شرطی ایجاد می‌شود، حذف سطرهایی است که در آن‌ها مقدار ویژگی برابر با ۲ است که این کار نیز به آسانی با یک عمل grep بر روی جداول به‌دست

آمده قابل انجام است.

اسکریپت‌های توسعه داده شده از مسیر

https://github.com/nkhdiscovery/PGM-Homeworks

قابل دریافت هستند.