تمرین کامپیوتری دوم ـ استنتاج علّی

بهراد منیری
۹۵۱۰۹۵۶۴
bemoniri@live.com

دانشکده‌ی مهندسی برق ـ دانشگاه صنعتی شریف

۱ بخش اول ـ پیاده‌سازی الگوریتم pc

در این بخش الگوریتم PC ر ا با آزمون فرض استقلال شرطی مبتنی بر Partial Correlation پیاده سازی می‌کنیم. برای پیاده‌سازی از زبان پایتون استفاده کرده و گراف را به کمک کتابخانه‌ی networkx می‌سازیم. برای محاسبات نیز از کتابخانه‌های numpy و sklearn استفاده می‌کنیم.

۱.۱ پیاده‌سازی آزمون فرض

برای پیاده‌سازی آزمون فرضیه، تابع test را نوشتیم: test (data, x, y, s)
این تابع، ماتریس داده‌ها و اندیس x ،y و مجموعه‌ای از اندیس‌ها، S، را در ورودی گرفته و آزمون فرضیه‌ی

$$X \perp\!\!\!\perp Y|S$$

را انجام می‌دهد و مقدار p-value را در خروجی بر می‌گرداند. هر اندیس نماینده‌ی ستون مورد نظر در ماتریس داده‌ها، یعنی یک متغیر است.

۲.۱ پیاده‌سازی الگوریتم pc

تابع pc پیاده‌سازی الگوریتم pc است که شبه‌کد آن ارائه شده.

pc (da, alpha)

این تابع، ماتریس داده‌ها که هر ستون آن یک متغیر است و همچنین مقدار significance level یعنی α را در ورودی دریافت کرده و اسکلت گراف مولد داده‌ها را تخمین می‌زند.

۳.۱ بررسی عملکرد الگوریتم با دو Toy Example

برای بررسی عملکرد الگوریتم، چند SCM گاوسی‌ـخطی ساخته و الگوریتم را روی آنها اجرا می‌کنیم. از هر متغیر یک میلیون سمپل می‌سازیم.

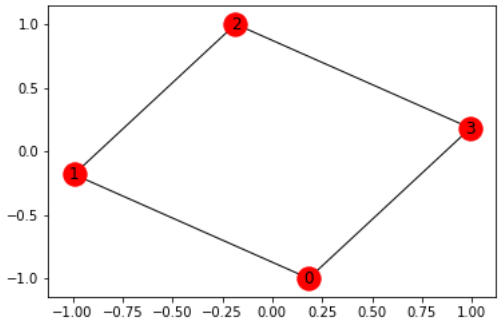
۱.۳.۱ مثال اول

$$\begin{cases} x_۰ = N_۰ \\ x_۱ = x_۰ + N_y \\ x_۲ = x_۱ + N_z \\ x_۳ = ۲x_۲ + ۳x_۰ + N_w \end{cases} \qquad N_۱, N_۲, N_۳, N_۴ \overset{iid}{\sim} Normal(۰, ۱) \tag{۱}$$

برنامه‌ی ما، دو عبارت استقلال زیر را از داده استخراج می‌کند.

$$\begin{cases} x_۰ \perp\!\!\!\perp x_۲ | x_۱ \\ x_۱ \perp\!\!\!\perp x_۳ | x_۰, x_۲ \end{cases} \tag{۲}$$

و دو یال متناظر را حذف می‌کند. شکل (۱) گراف تخمین زده شده است. **تمام یال‌ها به درستی انتخاب شدند.**



شکل ۱: گراف تخمین‌زده‌شده برای مثال اول

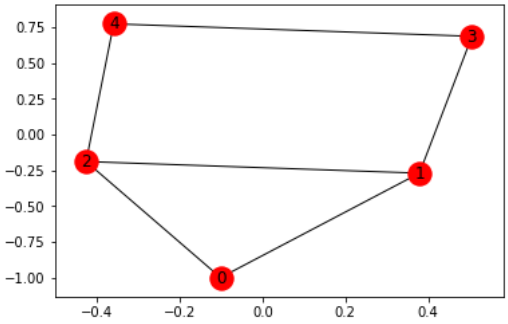
۲.۳.۱ مثال دوم

$$\begin{cases} x_۰ = N_۰ \\ x_۱ = x_۰ + N_۱ \\ x_۲ = ۲x_۰ + N_۲ \\ x_۳ = ۳x_۱ + N_۳ \\ x_۴ = ۴x_۲ + N_۴ \end{cases} \qquad N_۰, N_۱, N_۲, N_۳, N_۴ \overset{ind.}{\sim} Normal(۰, \sigma_i) \tag{۳}$$

برنامه‌ی ما، دو عبارت استقلال زیر را از داده استخراج می‌کند.

$$\begin{cases} x_۱ \perp\!\!\!\perp x_۴ | x_۲, x_۳ \\ x_۰ \perp\!\!\!\perp x_۴ | x_۱, x_۲ \\ x_۲ \perp\!\!\!\perp x_۳ | x_۱ \\ x_۰ \perp\!\!\!\perp x_۳ | x_۱ \end{cases} \tag{۴}$$

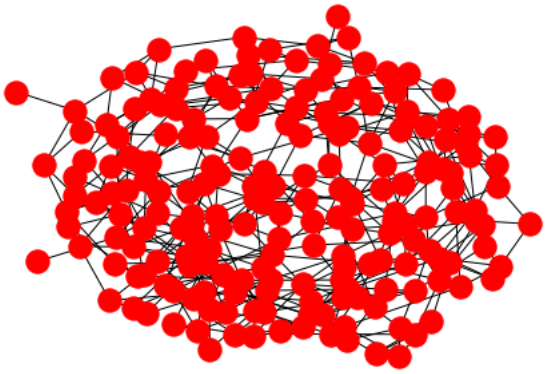
و یال‌های متناظر را حذف می‌کند. شکل (۲) گراف تخمین زده شده است. **تمام یال‌ها به درستی انتخاب شدند.**



شکل ۲: گراف تخمین‌زده‌شده برای مثال دوم

۲ بخش دوم ـ تخمین گراف مولد داده‌های ضمیمه

به کمک تابع pc اسکلت گراف را تخمین می‌زنیم. این الگوریتم بعد از سطح $l = ۶$ متوقف شده و ۴۱۵ یال در این گراف باقی می‌ماند.



شکل ۳: خروجی الگوریتم

۳ بخش سوم ـ پیاده‌سازی الگوریتم pc پایدار

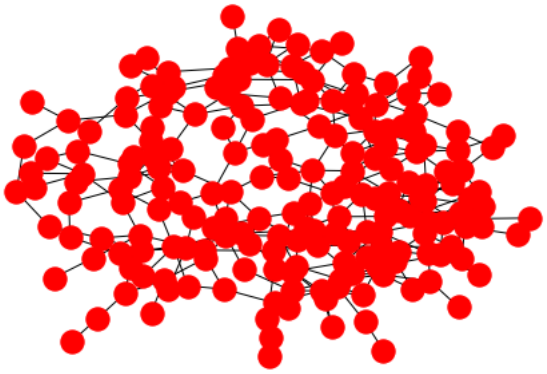
با تغییری کوچک در الگوریتم pc می‌توان به الگوریتم pc-stable رسید. در الگوریتم پایدار، بعد از اینکه تصمیم به قطع یک یال گرفتیم، یال مورد نظر را ذخیره می‌کنیم و در آخر هر سطح، تمام یال‌هایی که باید حذف شوند را یک‌جا حذف می‌کنیم. در الگوریتم اصلی این امکان وجود دارد که به دلیل حذف اشتباه یک یال، برخی از آزمون فرض‌ها را دیگر انجام ندهید و به ای دلیل خطا به شدت گسترش پیدا کند. در روش پایدار تا حدی این اثر کم‌رنگ می‌شود.

۱.۳ بررسی عملکرد الگوریتم پایدار با دو Toy Example

در این بخش الگوریتم pc-stable را بر روی داده‌های تولید شده توسط همان دو SCM بخش (۳.۱) اجرا می‌کنیم. مجدداً الگوریتم هر دو اسکلت را به درستی تخمین می‌زند.

۲.۳ اعمال الگوریتم پایدار بر داده‌های ضمیمه‌شده

با الگوریتم pc-stable اسکلت گراف مولد داده‌ها را تخمین می‌زنیم. این الگوریتم زمان بیشتری نسبت به الگوریتم pc می‌گیرد و در نهایت در آخر سطح ۶ متوقف شده و ۳۱۳ یال باقی می‌ماند.



شکل ۴: خروجی الگوریتم پایدار

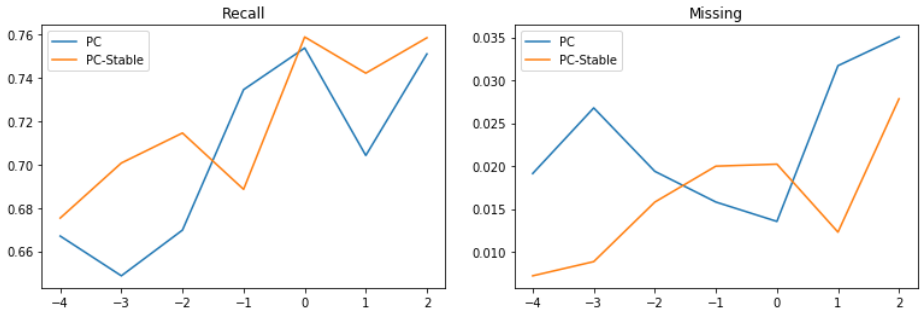
۴ بخش چهارم ـ بررسی اثر درصد اطمینان

در این بخش درصد اطمینان را در بازه‌ی $٪۲^{-۴}$ تا ٪۲ تغییر می‌دهیم و برای هر بازه‌ی اطمینان، ۲۰۰ بار الگوریتم مدنظر را اجرا می‌کنیم. در هر اجرا دو کمیت زیر را محاسبه می‌کنیم.

۱. Recall نسبت تعداد یال‌های گراف واقعی که درست تشخیص داده‌شده به تعداد کل یال‌های گراف واقعی

۲. Missing نسبت تعداد یال‌هایی که ما تشخیص دادیم درحالی که در گراف اصلی نبوده به تعداد کل یال‌های گراف واقعی

با افزایش ترشهولد، مشاهده می‌شود که Recall و Missing افزایش می‌یابد. این موضوع بدیهی است زیرا با زیاد کردن ترشهولد، با سخت‌گیری کمتری یال‌ها را حذف می‌کنیم پس درکل تعداد یال‌های حذف شده زیاد می‌شود و می‌توان به یال‌های باقی‌مانده اطمینان بیشتری داشت. دو نمودار زیر، نتایج این بخش هستند:



شکل ۵: مقایسه‌ی دو الگوریتم

الگوریتم پایدار عملکرد بهتری داشته زیرا در الگوریتم pc معمولی چندین مشکل وجود دارد. اول اینکه با تغییر دادن ترتیب آزمون‌ها، نتایج متفاوت خواهد بود. دوم اینکه با در صورت یک خطا در آزمون، تعدادی آزمون دیگر انجام نخواهند شد. این دو مشکل تا حد زیادی در pc-stable حل شده است. به همین دلیل تعداد یال‌های حذف‌شده در روش پایدار بیشتر از روش معمولی است. مشاهده می‌شود در هر دو الگوریتم، با کاهش تعداد رئوس، نتایج پیشرفت قابل ملاحظه‌ای می‌کنند. دلیل این امر این است که در گراف‌های کوچک، خطا فرصت انتشار ندارد. این بهبود تا جایی است که برای گراف‌هایی با تعداد راس کمتر از ۱۵، Recall تقریباً برابر یک است و Missing مساوی صفر.