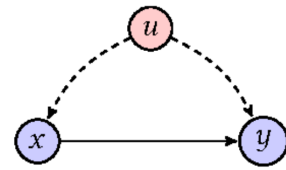


به نام خدا

علی فتحی ۹۴۱۰۹۲۰۵



تکلیف کامپیوتری سری ۲ استنتاج علی – دکتر صالح



**درباره داده‌ها:** آن طور که بنظر می‌آید، تعداد ۲۰۰ عدد متغیر در فایل pc-data.csv موجود است (متناظر با تعداد ستون‌ها). همچنین فایل، ۵۰۰۰ خط دارد که بیانگر آن است از این ۲۰۰ متغیر، ۵۰۰۰ داده آزمایش در دست داریم.

**درباره خروجی:** کد بخش اول و دوم، PC ساده، با نام PCalgNew.py ذخیره شده است. در فولدر تمرین یک کد دیگر با نام PCalg.py نیز آمده است که همان عملکرد را دارد و آن را با استفاده از کلاس بندی در پایتون نوشتیم؛ اما سرعت آن پایین است (زیرا تمام تست‌های استقلال را ذخیره می‌کند و نیز مجموعه جدا کننده رؤس نیز برای تشخیص جهت ذخیره می‌گردد) و با محدودتر کردن عملکرد آن، از PCalgNew.py استفاده می‌کنم. برای قسمت دوم (Stable PC) نیز کد PCalgStable.py نوشته شده است. خروجی ماتریس مجاورت‌ها نیز در دو فایل به نام‌های adjacent\_pc.csv و adjacent\_pc\_stable.csv ذخیره شده است که در فولدر سوال موجود است.

**درباره کد:** برای مشاهده زمان اجرا، از کتابخانه tqdm استفاده شده است:

```
l= 0
100%|██████████| 39800/39800 [00:00<00:00, 93846.28it/s]
l= 1
100%|██████████| 38640/38640 [00:28<00:00, 1365.81it/s]
l= 2
13%|███| 828/6412 [00:03<00:19, 289.09it/s]
```

خروجی نیز علاوه بر ذخیره، به صورت زیر نیز چاپ می‌شود (ماتریس مجاورت boolean است):

```
Finally, DAG Edges Are:
( 0 , 28 )
( 0 , 65 )
( 0 , 128 )
( 0 , 136 )
( 0 , 185 )
( 1 , 99 )
( 1 , 114 )
( 1 , 138 )
( 1 , 159 )
( 2 , 58 )
( 2 , 187 )
( 2 , 195 )
( 3 , 33 )
( 3 , 88 )
( 3 , 89 )
```

زمان اجرای PC ساده حدود ۱ دقیقه و زمان اجرای Stable PC حدود ۱ دقیقه و ۳۰ ثانیه است.

### بخش د: مقایسه

برای این بخش، چون به استفاده مکرر از PC و Stable PC نیاز است، این دو را در یک module به نام PCfuncs ریختم و آن در ابتدای کد این قسمت، به نام PCEval، import کرده‌ام.

نتایج این قسمت به صورت زیر است:

```
100%|██████████| 200/200 [00:47<00:00, 3.87it/s] For Accept Level = 0.25% :
Simple PC [TP, TN, FP, FN]= [ 6.62625 2.8625 7.03875 83.4725 ] Percent
Stable PC [TP, TN, FP, FN]= [ 6.56375 2.925 6.85125 83.66 ] Percent

100%|██████████| 200/200 [00:49<00:00, 3.67it/s]
0%|██████████| 0/200 [00:00<?, ?it/s] For Accept Level = 0.5% :
Simple PC [TP, TN, FP, FN]= [ 6.65 2.74 7.155 83.455 ] Percent
Stable PC [TP, TN, FP, FN]= [ 6.61125 2.77875 6.94125 83.66875 ] Percent
100%|██████████| 200/200 [00:58<00:00, 3.90it/s]
For Accept Level = 1% :
0%|██████████| 0/200 [00:00<?, ?it/s] Simple PC [TP, TN, FP, FN]= [ 6.85375 2.815 7.34125 82.99 ] Percent
Stable PC [TP, TN, FP, FN]= [ 6.815 2.85375 7.17 83.16125 ] Percent
100%|██████████| 199/200 [01:04<00:00, 3.30it/s] For Accept Level = 2% :
Simple PC [TP, TN, FP, FN]= [ 6.9675 2.50125 7.64 82.89125 ] Percent
Stable PC [TP, TN, FP, FN]= [ 6.93375 2.535 7.42625 83.105 ] Percent
100%|██████████| 200/200 [01:04<00:00, 3.11it/s]
```

که در نمایش ها، TP به معنی یال‌هایی اند که بوده‌اند و تشخیص داده شده اند، TN، یال‌هایی اند که بوده‌اند ولی حذف شده اند، FP، یال‌هایی اند که وجود ندارند ولی حذف نشده اند، و FN، یال‌هایی اند که وجود ندارند و به درستی حذف شده اند که درصد بیشتری را تشکیل می‌دهند. ملاحظه می‌شود حدود ۱۰ درصد خطا داریم (حذف غلط یا نگه داشتن غلط) و ۹۰ درصد صحت داریم.

در مورد رابطه با سطح اطمینان نیز می‌توان گفت که هر چه این سطح بیشتر باشد (مثلا ۲ درصد نسبت به ۰.۲۵) حذف غلط یال‌ها کمتر می‌شود (مثلا در اینجا حدود ۳ درصد یال‌ها در سطح اطمینان ۰.۲۵ درصد به اشتباه حذف شده اند، و این مقدار به ۲.۵ درصد در سطح اطمینان ۲ درصد کاهش یافته است) که دلیل آن هم مشخص است؛ چون زیرا هرچه درجه اطمینان بیشتر و wide تر باشد، یال‌های کمتری خط می‌خورند. اما در عوض نیز یال‌هایی که به اشتباه خط نخورده اند هم بیشتر می‌شوند. پس، در اینجا یک بده بستانی وجود دارد؛ و اگر حذف یال موجود از حذف نکردن یال ناموجود برای ما مهمل‌تر است، خوب است سطح اطمینان را عدد بیشتری بگیریم.

\*\*\*\*\* پایان گزارش \*\*\*\*\*