بسمه تعالی

استنتاج علّی — نیمسال اول ۹۸ــ۹۹
تمرین کامپیوتری سری سوم — مدرس: صابر صالح

سوال ۱

در این تمرین به شما دو سری زمانی داده میشود و از شما خواسته میشود ماتریس مجاورت متناظر با هرکدام از این دو سری زمانی را با الگوریتم های Granger Causality و PC و FCI و LinGAM بدست آورید. توجه کنید که ملاحظات لازم برای استفاده از برخی از الگوریتم های ذکر شده در سری های زمانی را حتما درنظر بگیرید.

الف

در فایل تمرین فایل مربوط به `model01.txt` یک سری زمانی است. با توجه به این حداکثر تاخیر در این سری زمانی ۱ میباشد، ماتریس های مجاورت متناظر با هر الگوریتم را درنظر بگیرید. ماتریس های خود را `graphModel01.txt` مقایسه کنید. برای مقایسه میتوانید از معیار Pearson Correlation استفاده نمایید.

ب

کدام الگوریتم ها در این مورد خاص از الگوریتم های دیگر برتری دارند؟ پاسخ خود را توجیه کنید.

پ(امتیازی)

برای فایل `model02.txt` که به همان روش سری زمانی فایل `model01.txt` شبیه سازی شده است، با هر روش دلخواه یک و تنها یک ماتریس مجاورت بدست بیاورید و این ماتریس را در یک فایل `tex`. ذخیره کرده و به همراه باقی تمرین آپلود کنید.

سوال ۲

در این قسمت قصد داریم تا تعداد گراف جهت دار بدون دور عضو یک کلاس هم ارزي مارکوف (MEC) را بشماریم. در ابتدا نیاز است برخی تعاریف را مرور کنیم. یک گراف زنجیره اي (chain graph) است اگر راس هاي آن را به گونه اي دسته بندي کنیم که تمام یال هاي بین راس هاي بین اعضاي هر دسته، مولفه زنجیره (chain component) ، بدون جهت باشند و یال هاي بین اعضاي دو مولفه زنجیره جهت دار باشند. براي مثال گراف کاملا بدون جهت یک مولفه زنجیره دارد که شامل همه ي رئوس گراف می شود و در گراف کاملا جهت دار هر گره یک مولفه ي زنجیر است. گراف اساسی یک گراف جهت دار بدون دور D یک گراف جزئی جهت دار است که اسکلتی مشابه گراف D دارد و یک یال در آن جهت دار است اگر و تنها اگر در تمام گراف هاي جهت دار هم ارز D جهت یکسانی داشته باشند. بنابراین هر کلاس هم ارزي مارکوف را می توان با یک گراف اساسی بیان کرد. می توان نشان داد که هر مولفه ي زنجیره در گراف اساسی یک گراف بدون جهت و همبند وتري (UCCG - undirected and connected chordal graph) است. اندازه ي یک MEC برابر با حاصلضرب اندازه ي کلاس هاي هم ارزي هر کدام از مولفه هاي زنجیره ي گراف اساسی است. پس براي شمردن اعضاي کلاس هم ارزي کافی است راه حلی براي به دست آوردن اندازه ي کلاس هم ارزي در UCCG ها ارائه کنیم:

$$SizeMEC(C) = \prod_{C_{\tau i} \in ChainComp(C)} SizeMEC(C_{\tau i})$$

روش پیشنهادي به صورت بازگشتی عمل می کند تا به یکی از ۵ گراف پایه اي برسد. اندازه ي کلاس هم ارزي یک UCCG با تعداد p راس و n یال براي این ۵ گراف پایه را می توان این گونه مشخص کرد:

- $n = p - 1 \rightarrow SizeMEC(U_p; n) = p$
- $n = p \rightarrow SizeMEC(U_p; n) = 2p$
- $n = p(p-1)/2 - 2 \rightarrow SizeMEC(U_p; n) = (p2 - p - 4)(p-3)!$
- $n = p(p-1)/2 - 1 \rightarrow SizeMEC(U_p; n) = 2(p-1)! - (p-2)!$
- $n = p(p-1)/2 \rightarrow SizeMEC(U_p; n) = p!$

در غیر از شرایط بالا اندازه ي کلاس هم ارزي را می توان به صورت بازگشتی حساب کرد.

$$SizeMEC(U) = \sum_{i=1}^{p} SizeMEC(U^{v_i})$$

که در عبارت بالا U^{v_i} گراف اساسی ریشه دار شده ي گراف U است. به این معنی که تمام یال هاي متصل به راس v_i به صورت خروجی هستند و جهت از v_i به سمت همسایه ها است. براي این قسمت نیاز به پیاده سازي تابع SizeMEC است که ماتریس مجاورت یک گراف اساسی را به عنوان ورودي می گیرد و اندازه ي کلاس هم ارزي متناظر با آن را نتیجه می دهد.

Algorithm 2: $SizeMEC(\mathcal{U})$

Input: $\mathcal{U}$: a UCCG.
Output: the size of Markov equivalence classes represented by $\mathcal{U}$

Let p and n be the numbers of vertices and edges in $\mathcal{U}$;
switch n **do**
  case $p-1$ **return** p;
  case p **return** $2p$;
  case $p(p-1)/2-2$ **return** $(p^2-p-4)(p-3)!$;
  case $p(p-1)/2-1$ **return** $2(p-1)!-(p-2)!$;
  case $p(p-1)/2$ **return** $p!$;
for $j \leftarrow 1$ **to** p **do**
  $\{\mathcal{U}_1, \cdots, \mathcal{U}_{l_j}\} \leftarrow ChainCom(\mathcal{U}, v_j)$;
  $s_j \leftarrow \prod_{i=1}^{l_j} SizeMEC(\mathcal{U}_i)$
return $\sum_{i=1}^{p} s_i$

Algorithm 1: $ChainCom(\mathcal{U}, v)$

Input: $\mathcal{U}$, a UCCG; v, a vertex of $\mathcal{U}$.
Output: $v-$rooted essential graph of $\mathcal{U}$ and all of its chain components.

Set $A = \{v\}$, $B = \tau \setminus v$, $\mathcal{G} = \mathcal{U}$ and $\mathcal{O} = \emptyset$
while *B is not empty* **do**
  Set $T = \{w : w$ in B and adjacent to $A\}$;
  Orient all edges between A and T as $c \rightarrow t$ in $\mathcal{G}$, where $c \in A, t \in T$;
  repeat
    for *each edge $y - z$ in the vertex-induced subgraph $\mathcal{G}_T$* **do**
      if *$x \rightarrow y - z$ in $\mathcal{G}$ and x and z are not adjacent in $\mathcal{G}$* **then**
        Orient $y - z$ to $y \rightarrow z$ in $\mathcal{G}$
  until *no more undirected edges in the vertex-induced subgraph $\mathcal{G}_T$ can be oriented*;
  Set $A = T$ and $B = B \setminus T$;
  Append all isolated undirected graphs in $\mathcal{G}_T$ to $\mathcal{O}$;
return *$\mathcal{G}$ and $\mathcal{O}$*