

تمرین سری 10

بکتاش انصاری

بخش اول :

طبق الگوریتم نشان داده شده ما از likelihood sampling استفاده میکنیم.

قبل از توضیح این الگوریتم دلیل وجود آن را بیان کنیم.

در روش Rejection Sampling شیوه نمونه برداری به این شکل بود که ابتدا ما طبق احتمال شرطی ای که داشتیم. مقدار زیادی نمونه به صورت طبیعی ایجاد میکردیم و طبق متغیرهای احتمال شرطیمان آن نمونه هایی که مطلوب ما نبودند را حذف میکردیم (Reject).

اما این روش مشکلاتی داشت یک اینکه ممکن بود از این نمونه هایی که ما تولید میکنیم درصد زیادی مطلوب ما نباشند و Reject شوند که این باعث سربار محاسباتی برای ما میشد.

به همین دلیل روش likelihood معرفی شد که در این روش ما بجای تولید نمونه به صورت تصادفی تولید آن را fix میکنیم و طبق احتمال مورد نظر موارد نامطلوب را تولید نمیکنیم که بخواهیم مثل روش قبل آن ها را Reject کنیم. اما این روش به طور خام چون **نمونه برداری دستکاری میشود توزیع احتمال ما به هم میریزد** و جواب نهایی Consistent نمیشود. به همین دلیل به نمونه ها **وزن** میدهیم که اگر نمونه دستکاری شده تولید شد با یک ضریب توزیع احتمال مناسب باشد.

طبق الگوریتم پیش میرویم :

- ابتدا evidence های خود را مشخص میکنیم (متغیرهایی که مقدار آن ها را میدانیم).
- سپس متغیر W را برای نمونه مورد نظر با مقدار اولیه 1 تعریف میکنیم.
- حال شروع میکنیم روی تمام متغیرها پیمایش انجام میدهیم.
- اگر متغیر مورد نظر درون evidence های ما بود آن متغیر انتخاب میشود و w در احتمال انتخاب آن ضرب میشود. (w آپدیت میشود)
- در غیر این صورت به صورت تصادفی نمونه انتخاب میشود.
- در نهایت نمونه انتخاب شده و w مورد نظر بدست می آید.

طبق کد 1000 نمونه را ایجاد میکنیم.

بخش دوم :

در روش likelihood ما مشکلاتی داریم که باعث ضعیف بودن این روش میشود.

مشکل اصلی ما این است که متغیرهای بالایی (upstream) از evidence های پایینی (downstream) اثر نمیگیرند.

و چون هدف ما این است که در تعداد نمونه های محدود به جواب نسبتاً مطلوبی برسیم این روش مشکل ساز میشود.

به همین دلیل روش Gibbs Sampling معرفی شد. که در این روش ما نمونه برداری را تکرار میکنیم ولی در هر بار ایجاد نمونه برداری مقادیر غیر Evidence را تغییر میدهیم و این کار را آنقدر تکرار میکنیم تا به یک همگرایی برسیم.

در اصل در تکرار ها متغیر هارا به جز یک متغیر fix میکنیم و مقدار متغیر انتخابی رو بهینه میکنیم.

الگوریتم این روش به این شکل است که :

ابتدا ما Evidence ها خود را مانند روش قبل مشخص میکنیم.

سپس برای نمونه اول که نمونه خوبی نیست و قرار است مکرراً تغییر کند مقادیر Non-Evidence را بصورت رندوم نمونه برداری میکنیم.

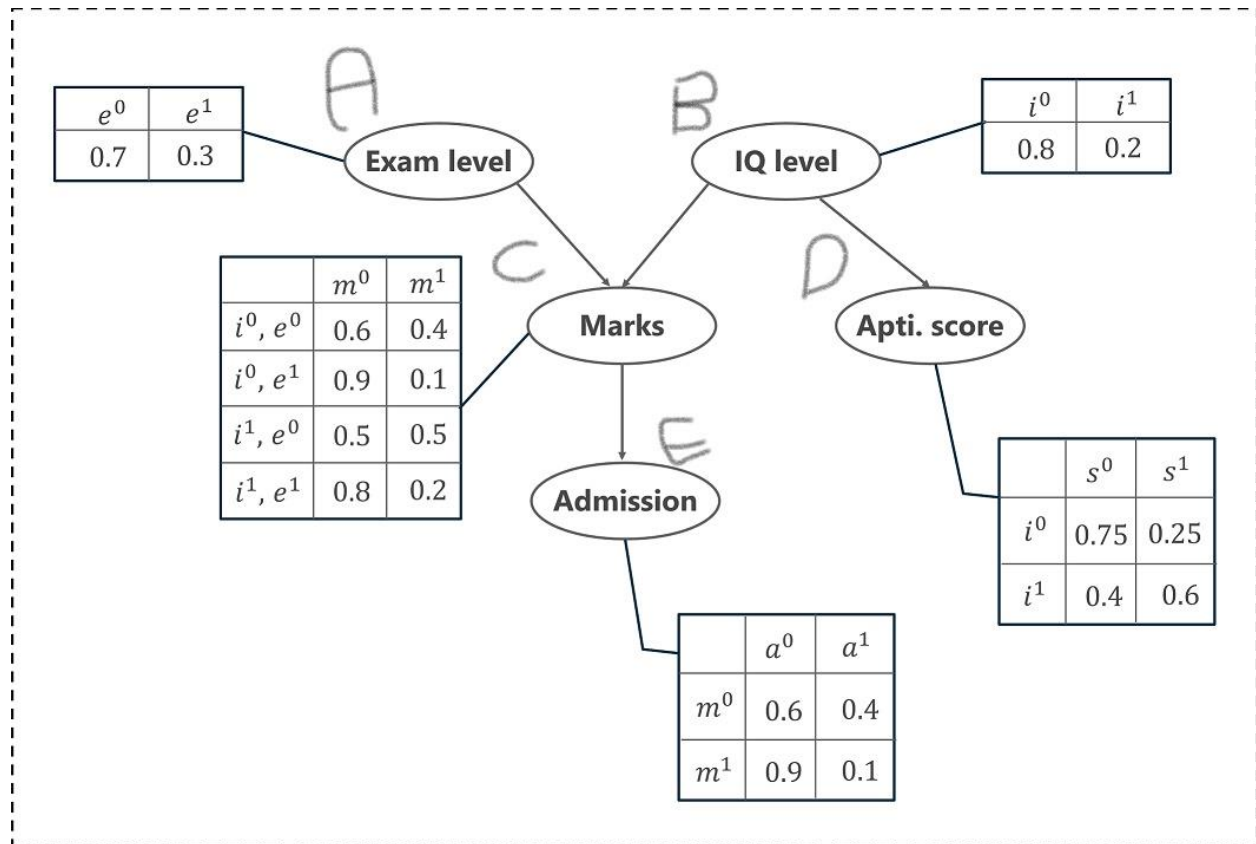
حال شروع به تکرار میکنیم و از بین non-evidence ها یک متغیر را انتخاب میکنیم و بقیه متغیر هارا با مقادیری که دارند fix میکنیم و آن متغیر را بهینه میکنیم و در تکرار بعدی از fix شده همین مقدار برای این متغیر استفاده میکنیم.

در نهایت با توجه به کد این تکرار هارا 1000 بار انجام میدهیم تا به یک مقدار نسبتاً مطلوب برسیم.

بخش سوم :

ابتدا یک bayesnet طراحی میکنیم و 7 کوئری میسازیم.

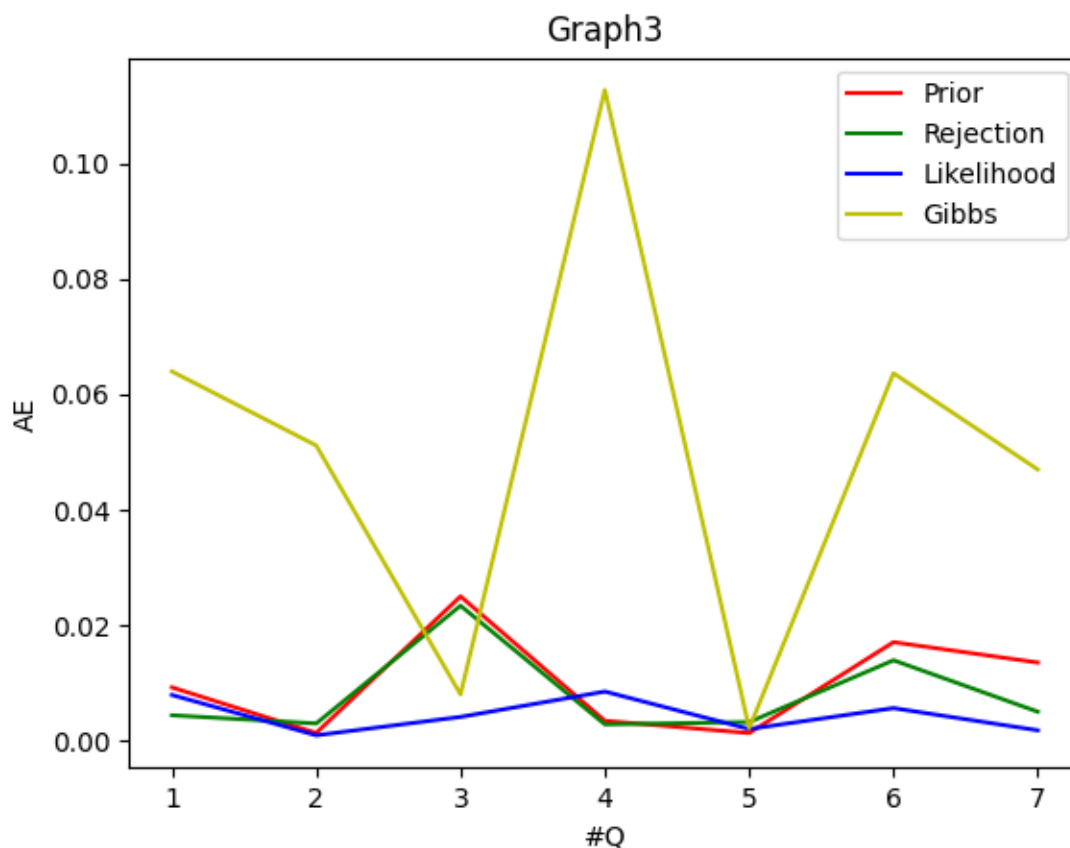
بیژنت به شکل زیر است :



و کوئری های مورد نظر نیز به شکل زیر هستند :

1. $P(A|E)$
2. $P(A', B|D')$
3. $P(D|A', B)$
4. $P(B, C|E', D)$
5. $P(E)$
6. $P(A, D|C', B)$
7. $P(B|C)$

و نمودار مورد نظر نیز به شکل زیر بدست میآید :



طبق نمودار بدست آمده و نمودار های قبلی میتوان به نتایجی رسید

در حالت کلی در روش Gibbs اگر کوئری مورد نظر از پایین به بالا باشد (Downstream به Upstream) این روش درست عمل نمیکند. (در حالت کلی کد پیاده سازی نیز برای این روش کامل نبود).

آنالیز کردن روش Prior نیز منطقی نیست زیرا که این روش کاملا بصورت تصادفی عمل میکند. در روش likelihood یکی از مشکلات اصلی که باعث جایگزین کردن روش Gibbs میشود این است که متغیر های بالایی (upstream) از evidence های پایینی (downstream) اثر نمیگیرند. که اینجا هم در کوئری ها مشاهده میشود.

فایل های تولیدی در فایل زیپ قرار داده میشود.