

گزارش تمرین برنامه نویسی (۴)

هوش مصنوعی

هلیا شمس زاده

۴۰۰۵۲۱۴۸۶

:Loading Model

ابتدا برای هر راس موجود (متغیرها) یک ایندکس تعریف می‌کنیم زیرا در ادامه کد، با ایندکس متغیرها کار می‌شود. سپس فایل `model` را باز کرده و می‌خواند. به ازای هر جدول، سه بخش داریم. بخش اول، متغیر وابسته است، بخش دوم والدهای آن متغیر و بخش سوم، احتمال درست بودن متغیر وابسته به شرط حالات مختلف متغیرهای والد. به ازای هر متغیر وابسته که خوانده می‌شود، این سه بخش تفکیک شده و `graph` و `cpts` پر می‌شود. توجه شود که به به ازای هر متغیر وابسته و حالت‌های مختلف `parent` های آن، هم احتمال مربوط به `true` بودن و هم `false` بودن آن متغیر در `cpts` قرار می‌گیرد.

:Prior Sampling

همانطور که گفته شد، با تابع `topological_sort`، گراف ساخته شده در بخش قبل مرتب شد. سپس به تعداد ۱۰۰۰۰ نمونه با کمک تابع `sample_vertex` ساخته شد. در این روش به خوب بودن نمونه به دست آمده اهمیتی داده نمی‌شود. این نمونه‌ها درون لیست `generated_samples` ذخیره می‌شوند. سپس به ازای هر نمونه موجود در این لیست، دو حالت را چک می‌کند. ابتدا تعداد نمونه‌هایی که `evidence` های آن با `evidence` های ورودی یکی می‌باشد را می‌شمارد. سپس تعداد نمونه‌هایی که هم `evidence` و هم `query` یکسان با ورودی دارند را می‌شمارد. و در آخر نسبت این دو را به عنوان خروجی برمی‌گرداند که قطعا کوچکتر مساوی یک می‌باشد.

:Rejection Sampling

در این روش، نمونه‌هایی که `evidence` های آن، با `evidence` ورودی یکی نیستند را `reject` کرده و در فرآیند نمونه‌برداری حساب نمی‌کند. در واقع نمونه‌های هوشمندانه‌تری از روش قبلی تولید می‌کند. فرقی با کد بخش قبل این است که مرحله چک کردن `evidence` ها را حین تولید نمونه انجام می‌دهد (برخلاف روش قبل که پس از تولید همه نمونه‌ها این کار را انجام می‌داد). در آخر، همانند بخش قبل، تعداد نمونه‌های `consistent` با کوئری را شمرده و نسبت را به عنوان احتمال برمی‌گرداند.

:Likelihood Sampling

در این روش متغیرهای evidence نمونه، از همان ابتدای نمونه‌برداری، مقادیر ورودی را دریافت می‌کنند. مزیت این روش این است که تمامی نمونه‌های ساخته شده با متغیرهای consistent evidence بوده و نمونه‌های غیرمفید تولید نمی‌شوند. اما از آنجایی که مقادیر برخی متغیرها در همان ابتدا ثابت شده است برای جلوگیری از خراب شدن توزیع احتمال متغیرها، به هر نمونه ضریبی داده می‌شود که برابر است با ضرب احتمال تمام متغیرهای evidence به شرط مقادیر parent هایشان.

در آخر هم به ازای هر نمونه تولید شده، چون می‌دانیم تمام نمونه‌ها evidence مشابهی با ورودی دارند، فقط کافی‌ست برابر بودن کوثری‌ها چک شود. وزن آنهایی که برابر بودند را به وزن کل نمونه‌های تولید شده تقسیم می‌کند و به عنوان خروجی برمی‌گرداند.

:Gibbs Sampling

در روش قبل، وزن برخی نمونه‌ها آنقدر کوچک بود که عملاً در نتیجه نهایی تاثیری نمی‌گذاشت. در روش Gibbs این مشکل حل شده است. مقادیر متغیرهای evidence نه فقط بر روی متغیرها و گره‌هایی که پایین درخت قرار دارند، بلکه بر روی گره‌های بالایی نیز اثر می‌گذارند. به این صورت که پس از آنکه متغیرهای evidence مقادیر ثابت خود را گرفتند و متغیرهای دیگر مقداردهی شدند، بار دیگر بر اساس sort topological متغیرها resample شده و مقدار آنها با توجه به تمامی متغیرهای دیگر بدست می‌آید و اینطوری نمونه‌های با وزن قابل توجه تولید می‌شوند. در صورتی که متغیر مورد بررسی در evidence قرار نداشت، تابع resample_vertex را صدا زده تا وزن مناسب به آن داده شود.