

گزارش پروژه‌ی چهارم مبانی و کاربردهای هوش مصنوعی

بخش ۲)

سوال: چرا باید تابع `normalize` را در این بخش فراخوانی کنیم؟ لزوم استفاده از آن را شرح دهید.

`Normalize` کردن جهت کسب اطمینان از این است که جمع احتمال‌ها برابر ۱ خواهد شد و از آنجا که در بیزین و استنتاج از احتمالات استفاده می‌کنیم، انجام این کار ضروری است. با فراخوانی تابع `normalize` از این امر و همچنین اینکه باورها به درستی توزیع شده‌اند و مقادیرشان معتبر است، مطمئن می‌شویم. در اینجا ما تابع `normalize` را فرامی‌خوانیم تا مطمئن شویم جمع مقادیر درون `beliefs` برابر ۱ خواهد شد.

در اینجا ما `beliefs` را که نشان‌دهنده احتمال حضور روح در یک موقعیت خاص است را نرمال می‌کنیم. طبیعتاً جمع این احتمال‌ها باید ۱ شود. علت نرمال سازی هم آن است که توزیع باورها با توجه به مشاهدات انجام شده، بروز شده است پس نرمال‌شان می‌کنیم تا مطمئن شویم که معتبر بوده و جمعشان برابر ۱ خواهد شد.

سوال: توضیح دهید مقدار `beliefs` در گذر زمان چگونه تغییر می‌کند؟

با گذشت زمان مقدار `beliefs` به مقدار واقعی‌تر نزدیک‌تر می‌شود و همگرا می‌شود. علت این امر نیز آن است که در هر مرحله از نتایج قبلی و مشاهده انجام شده، در حال استفاده هستیم. در ابتدا با ابر بزرگی از `belief` روبرو هستیم که به مرور و با انجام مشاهدات، کاهش می‌یابد. (همانطور که در اجرا مشاهده می‌شود به مرور زمان برخی نقاط تیره و تیره‌تر می‌شوند که نشان‌دهنده کاهش احتمال حضور روح در آن نقاط و همگرا شدن به مقدار واقعی است.)

برای هر موقعیت روی نقشه، `belief` نشان‌دهنده احتمال حضور روح در آن نقطه است. با انجام هر مشاهده (`observe`) توسط سنسورها، مقادیر `beliefs` در تابع `observeUpdate` به کمک `getObservationProb` که احتمال مشاهده انجام شده بر اساس موقعیت پکمن، موقعیت روح و موقعیت زندان است، بروزرسانی می‌شود. این بروزرسانی با ضرب مقدار `beliefs` در آن موقعیت (احتمال حضور روح در آن موقعیت) در مقدار حاصل از `getObservationProb` و ذخیره آن به عنوان مقدار جدید در `beliefs` انجام می‌شود. این ضرب بر اساس قانون بیز انجام می‌شود یعنی:

$$P(A|B) = \frac{P(B|A) \times P(A)}{P(B)}$$

در اینجا A معادل beliefs و B معادل مشاهده است. مقدار $P(A|B)$ نشان‌دهنده احتمال باور با توجه به مشاهده انجام شده است.

در نهایت نیز برای آنکه جمع احتمال‌ها برابر ۱ شوند باید normalize کنیم.

با توجه به توضیحات بالا، با انجام هر مشاهده، در حال بروز کردن مقدار احتمال حضور روح در یک موقعیت خاص هستیم (belief) و از آنجا که از مشاهدات و نتایج قسمت‌های قبل استفاده می‌کنیم پس به مرور زمان این مقدار همگرا شده و به مقدار درستش نزدیک می‌شود.

بخش ۳)

سوال: کاربرد کلاس DiscreteDistribution را به همراه متدهای آن توضیح دهید.

همانطور که در تعریف کلاس نیز مشخص است، این کلاس از dict ارث‌بری می‌کند و در واقع یک دیکشنری است. از این کلاس برای مدل کردن توزیع احتمال‌ها بر روی مجموعه‌ای گسسته از کلیدها استفاده می‌شود. در این پروژه از آن برای مدل کردن توزیع باورها استفاده شده است. تمامی state‌ها را به عنوان key برای آن تعریف کرده و value متناظر با هر کلید می‌شود احتمال مربوط به آن حالت.

کلید: حالت

مقدار: احتمال

Key: State

Value: Probability

متودهای این کلاس:

__getitem__(self, key):

مقدار متناظر با کلید را برگرداند. اگر کلید در توزیع وجود نداشته باشد مقدار ۰ را برای آن ست کرده و برمی‌گرداند.

`:copy(self)`

یک کپی از توزیع را برمی گرداند.

`:argMax(self)`

کلیدی (حالتی) که بیشترین مقدار (بیشترین احتمال) را در توزیع دارد، برمی گرداند.

`:total(self)`

مجموع مقادیر تمام کلیدها (احتمالها) را برمی گرداند.

`:normalize(self)`

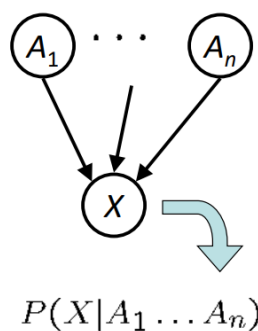
توزیع را نرمال می کند تا جمع احتمالها (مقادیر تمام کلیدها) برابر ۱ شوند. نسبت مقادیر میان تمام کلیدها ثابت می ماند. اگر جمع احتمالها (مقادیر) صفر باشد، کاری نمی کند.

`:sample(self)`

یک نمونه تصادفی از توزیع می گیرد و کلید (حالت) متناظر را برمی گرداند. این نمونه گیری با توجه به مقداری (احتمال) که هر کلید (حالت) دارد، وزن دار است.

سوال: چگونه ساختار شبکه بیزین در پروژه وابستگی های بین متغیرها را نشان می دهد؟

شبکه بیزین یک گراف جهت دار بدون دور است و جداول احتمال شرطی برای هر متغیر به صورت زیر است.



به کمک این جداول شرطی (و گراف) می توانیم وابستگی و ارتباط میان متغیرها را نمایش دهیم.

در شبکه بیزین متغیرهای تصادفی با گره‌ها و ارتباط و وابستگی میان آنها با یال میان گره‌ها نمایش داده می‌شود. همچنین برای بدست آوردن احتمال یک assignment کامل از رابطه زیر استفاده می‌شود:

$$P(x_1, x_2, \dots, x_n) = \prod_{i=1}^n P(x_i | \text{parents}(X_i))$$

شبکه بیزین کمک می‌کند تا وابستگی‌ها و استقلال‌های (شرطی و غیرشرطی) میان متغیرها را مدل کنیم. به کمک شبکه بیزین می‌توان روابط میان متغیرها را مدل کرد و در استنتاج و روزرسانی مقادیر از این وابستگی‌ها و شبکه بیزین استفاده کرد. ما با استفاده از احتمالات شرطی، قانون بیز، مشاهدات و روابط و وابستگی‌ها و استقلالی که میان متغیرها وجود دارد می‌توانیم استنتاج کنیم و به جواب مدنظر برسیم.

در این پروژه ما متغیرهایی میان موقعیت روح، موقعیت پکمن، موقعیت زندان و ... را داریم.

ما با استفاده از شبکه بیزین و انجام محاسبات و احتمالات بر اساس مشاهدات، می‌توانیم در نهایت توزیع احتمال روح در صفحه بازی را بفهمیم و متوجه شویم که محتمل ترین جایی که روح در آن حضور دارد کجاست و سعی کنیم به آن سمت برای شکارش برویم و بهینه عمل کنیم.