دانشکده‌ی مهندسی کامپیوتر

یادگیری ماشین تابستان ۱۴۰۰

پاسخ تمرین سری هشتم

مدرس: دکتر محمّدحسین رهبان

پاسخ سوال ۱ *Dimensionality Reduction*

آ) اینجا را نگاه کنید.

ب) استفاده از *SVD* راحت‌تر است چرا که تعداد نمونه‌ها از تعداد ویژگی‌ها کمتر است. در نتیجه تجزیه یک ماتریس با ابعاد "تعداد نمونه‌ها * تعداد ویژگی‌ها" از نظر محاسباتی به صرفه‌تر از یک ماتریس با ابعاد "تعداد ویژگی‌ها * تعداد ویژگی‌ها" است.

پاسخ سوال ۲ *Clustering*

آ) برای N داده با k خوشه، در کل k^N حالت خوشه‌بندی می‌توانیم داشته باشیم؛ پس دامنه تعداد خوشه‌ها محدود است و نمی‌تواند تا بی‌نهایت ادامه داشته باشد مگر اینکه در حلقه بیفتد (پس از چند مرحله، دوباره مرکزهای قبلی انتخاب شوند). ولی چون در هر مرحله الگوریتم *k-means* ، خوشه‌بندی جدید با استفاده از آخرین خوشه‌بندی ساخته می‌شود و تنها زمانی الگوریتم ادامه می‌یابد که مرکز خوشه‌های جدید متفاوت با مرکز خوشه‌های قبلی باشند (یا بیشتر از ϵ با هم فرق داشته باشند) و *MSE* کمتری داشته باشند، که چون *MSE* مرکزهای جدید کمتر از مرکزهای مرحله قبل است، با افتادن در حلقه *MSE* مرحله‌ای کمتر از *MSE* خودش می‌شود که قبلا انتخاب شده بود، که این امکان پذیر نیست. پس در حلقه نیز نمی‌افتد. لذا در تعداد مراحل محدودی *k-means* خاتمه می‌یابد.

ب) در ناحیه‌ای با تمرکز کم داده، مرکزهای بیشتری در نهایت تجمیع می‌شوند. چون در این الگوریتم می‌خواهیم میانگین فاصله‌ها در خوشه‌ها را کمینه کنیم، پس در ناحیه با تمرکز کم داده که فضا بزرگتر است نیاز به تعداد مرکز بیشتری داریم.

پ) با این معیار انتخاب مرکز در اولین مرحله، دورترین نقطه به مرکزهای قبلی انتخاب می‌شود. چون مرکزها در نزدیکی کم انتخاب نمی‌شوند، تعداد مرحله‌های اجرای الگوریتم کاهش یافته و زودتر به پاسخ نهایی می‌رسیم ولی احتمال دارد به یک بهینه محلی برسیم.

پاسخ سوال ۳ *Reinforcement Learning*

آ) نرخ یادگیری را برابر با α و ضریب تخفیف را برابر با γ می‌گیریم. پاداش در هر دور رفتن از هر حالتی (غیر از حالت‌های پایانی) صفر بوده و برای حالت‌های پایانی نیز با توجه به موقعیت آن‌ها -1 یا 1 خواهد بود. به این ترتیب در دور اول داریم:

$$Q(s_{11}, right) = (1-\alpha)\, Q(s_{11}, right) + \alpha \left(r + \gamma \max_{a \in \mathbb{R}} Q(s_{12}, a)\right) = 0.5\,(1-\alpha) + 0.5\,\alpha\gamma$$

$$Q(s_{12}, right) = (1-\alpha)\, Q(s_{12}, right) + \alpha \left(r + \gamma \max_{a \in \mathbb{R}} Q(s_{13}, a)\right) = 0.5 - 1.5\,\alpha$$

و در دور دوم داریم:

$$Q(s_{11}, right) = (1-\alpha)\, Q(s_{11}, right) + \alpha\big(r + \gamma \max_{a\in\mathbb{R}} Q(s_{12}, a)\big) = 0.5\,(1-\alpha)^2 + 0.5\,\alpha\gamma\,(2-\alpha)$$

$$Q(s_{12}, down) = (1-\alpha)\, Q(s_{12}, down) + \alpha\big(r + \gamma \max_{a\in\mathbb{R}} Q(s_{22}, a)\big) = 0.5\,(1-\alpha) + 0.5\,\alpha\gamma$$

$$Q(s_{22}, down) = (1-\alpha)\, Q(s_{22}, down) + \alpha\big(r + \gamma \max_{a\in\mathbb{R}} Q(s_{32}, a)\big) = 0.5\,(1-\alpha) + 0.5\,\alpha\gamma$$

$$Q(s_{32}, right) = (1-\alpha)\, Q(s_{32}, right) + \alpha\big(r + \gamma \max_{a\in\mathbb{R}} Q(s_{33}, a)\big) = 0.5\,(1+\alpha)$$

ب) در حالت کلی تنها می‌توان گفت $\gamma_1 > \gamma_2$. دلیل آن این است که اگر با ضریب تخفیف γ_1، همان سیاست γ_2 بازی شود مقادیر V اکیدا زیاد می‌شوند؛ بنابراین Q بیشتر متناظر با γ بیشتر است. اگر بازی قطعی باشد، می‌توان تحلیل دقیق‌تری انجام داد. پس از انجام اکشن *right* در خانه s_{11} به خانه s_{12} می‌رسیم و از اینجا به بعد سیاست بهینه این است که پس از سه حرکت دیگر به s_{33} برسیم. بنابراین قبل از دریافت پاداش 1، سه حرکت انجام می‌دهیم یعنی:

$$Q_1(s_{11}, right) = \gamma_1^3, \quad Q_2(s_{11}, right) = \gamma_2^3 \implies \gamma_1 = (9/7)^{(1/3)} \times \gamma_2$$

پ) به ازای هرکدام از حالت‌های غیر پایانی به جز s_{22} و s_{11} سه اکشن، به ازای s_{11} دو اکشن و به ازای s_{22} چهار اکشن داریم؛ بنابراین در مجموع تعداد *Q-value*هایی که باید یاد گرفته شوند 18 تاست. در نتیجه الگوریتم *Q-learning* نیاز به یادگیری 18 پارامتر دارد.

اگر از روش‌های *model-based* استفاده کنیم باید به ازای هر زوج حالت و اکشن، احتمالات و پاداش‌های رفتن به هر حالت همسایه را یاد بگیریم. یعنی فرض می‌کنیم پاداش‌ها را از قبل نمی‌دانیم. همچنین فرض می‌کنیم می‌دانیم احتمال رفتن از هر خانه به یک خانه غیر همسایه صفر است. به این ترتیب به ازای هر حالت غیر پایانی به جز s_{22} و s_{11} باید ۱۸ پارامتر، به ازای s_{11}، ۸ پارامتر و به ازای حالت s_{22} باید ۳۲ پارامتر را یاد بگیریم. بنابراین در روش *model-based* در مجموع باید ۱۱۲ پارامتر را یاد بگیریم. بنابراین تعداد پارامترهای روش *Q-learning* کمتر بوده و این روش بهینه است.

پاینده باشید