



دانشکده‌ی مهندسی کامپیوتر

یادگیری ماشین

بهار ۱۴۰۰

تمرین سری هشتم

مدرس: دکتر محمدحسین رهبان

زمان تحویل: ۲ و ۱۸ تیر

سوال ۱ Dimensionality Reduction

تجزیه مقدار تکین^۱ روشی است که با استفاده از آن ماتریس دلخواه $X \in \mathbb{R}^{m \times n}$ به شکل زیر تجزیه می‌شود ($m \leq n$):

$$X = USV^T$$

که S یک ماتریس قطری $n \times n$ است و ماتریس‌های U و V یکانی^۲ هستند.

(آ) با استفاده از تجزیه مقدار تکین تبدیل PCA ^۳ را پیدا کنید. (۷ نمره)

(ب) درحالتی که تعداد مشاهده‌ها بسیار کمتر از ابعاد آن است، برای انجام PCA استفاده از تجزیه مقدار تکین بهینه است یا بردار ویژه‌های ماتریس کواریانس؟ چرا؟ (۸ نمره)

سوال ۲ Clustering

(آ) ثابت کنید الگوریتم k -means همگرا می‌شود. (۴ نمره)

(ب) برای مجموعه داده‌ای با n داده و k خوشه ($k > 2$)، نیمی از داده‌ها در یک ناحیه متمرکزند و نیمی دیگر در ناحیه‌ای با چگالی کمتر قرار دارند که این دو ناحیه کم‌وبیش جدا از یکدیگرند. پس از خوشه‌بندی با در نظر گرفتن میانگین مربع‌های خطا^۴، آیا در نهایت مرکز خوشه‌ها به صورت یکنواخت بین دو ناحیه بیان شده توزیع می‌شوند یا در ناحیه‌ای چگالی بیشتری دارند؟ دلیل‌های خود را شرح دهید. (۵ نمره)

(پ) الگوریتمی برگرفته از k -means را این‌گونه در نظر بگیرید که در اولین مرحله انتخاب مرکزها، اولین مرکز به صورت تصادفی انتخاب شود ولی مرکز بعدی نقطه‌ای با بیشترین احتمال بدست آمده از فرمول زیر است. مرکزهای بعدی نیز به شیوه‌ای مشابه مشخص می‌شوند. در مرحله‌های بعدی (هنگام بروز کردن مرکزها) نیز نقطه‌ای با بیشترین احتمال بدست آمده از فرمول زیر را به عنوان مرکز جدید برمی‌گزینیم.

$$\frac{d^2(\mathbf{x})}{\sum_{\mathbf{x} \in D} d^2(\mathbf{x})}$$

^۱Singular Value Decomposition

^۲Uniform Matrix

^۳Principal Component Analysis

^۴Mean Squared Error (MSE)

که $d(x)$ بیانگر طول کوتاه‌ترین فاصله بین داده x تا نزدیکترین مرکزی است که تاکنون انتخاب شده است. D نیز نشان دهنده مجموعه تمام داده‌هاست. سرعت همگرایی و نتیجه نهایی خوشه‌بندی این الگوریتم را با نتیجه k -means مقایسه کنید. (۶ نمره)

سوال ۳ Reinforcement Learning

در دنیای شطرنجی شکل زیر، $agent$ همواره از وضعیت s_{11} شروع می‌کند. در این شکل وضعیت‌های پایانی با دایره و سایر وضعیت‌ها با مربع نشان داده شده‌اند. همچنین فضای حرکت‌ها، چهار جهت اصلی است. رسیدن به وضعیت s_{33} پاداشی برابر با ۱ و وضعیت‌های s_{13} و s_{31} پاداشی برابر با -۱ خواهند داشت. فرض کنید تمام مقدار اولیه‌های Q در وضعیت‌های غیر پایانی را به ۰.۵ و در وضعیت‌های پایانی را به صفر تغییر داده‌ایم.



آ) دو دنباله حرکت زیر را در نظر بگیرید:

$$s_{11} \rightarrow s_{12} \rightarrow s_{13}$$

$$s_{11} \rightarrow s_{12} \rightarrow s_{22} \rightarrow s_{32} \rightarrow s_{33}$$

برای هر دنباله، Q -value ها و شیوه بروزرسانی آن‌ها در هر مرحله را نشان دهید. (۶ نمره)

ب) فرض کنید دو الگوریتم Q -Learning با ضریب تخفیف‌های γ_1 و γ_2 را تا مرحله همگرایی اجرا کرده‌ایم و Q -value های زیر برای دو الگوریتم بدست آمده است:

$$Q_1(s_{11}, right) = 0.9, \quad Q_2(s_{11}, right) = 0.7$$

درباره ی نسبت مقدار های γ_1 و γ_2 چه می توان گفت؟ (۶ نمره)

پ) چه تعداد پارامتر باید برای یادگیری Q -value های مربوط به این دنیای شطرنجی یاد گرفت؟ اگر از روش های $model$ -based استفاده کنیم چطور؟ کدام روش بهینه است؟ (۴ نمره)

سوال ۴ (عملی) فایل Notebook

ی که در اختیار شما قرار داده شده را کامل کنید.

آ) در این بخش می‌خواهیم دو الگوریتم k -means و GMM را بدون استفاده از کتابخانه‌هایی که در زبان $python$ وجود دارند، پیاده‌سازی کنید.

توضیحات بیشتر در فایل پیوست ارایه شده است. (۱۵ نمره)

⁵Discount Factor

ب) کتابخانه *OpenAI Gym* شامل مجموعه‌ای از محیط‌های یادگیری تقویتی^۶ در زبان *python* است. در این سوال محیط *Mountain Car* از این مجموعه را مورد بررسی قرار می‌دهیم. *Mountain Car* مساله‌ای از یادگیری تقویتی است که هدف آن یادگیری سیاستی برای صعود ماشین از تپه‌ای شیب‌دار و رسیدن به هدف مشخص شده با پرچم است. همچنین موتور ماشین به اندازه کافی قدرتمند نیست که بتواند یکراست از تپه سمت راست صعود کند. بنابراین باید با صعود از تپه سمت چپ، شتاب کافی را کسب نماید. در این مسئله، حالت ماشین با آرایه‌ای شامل جایگاه و سرعت آن مشخص می‌شود. محدوده جایگاه و سرعت ماشین را در جدول زیر مشاهده می‌کنید:

<i>Max</i>	<i>Min</i>	<i>Observation</i>	<i>Num</i>
<i>0.6</i>	<i>-1.2</i>	<i>position</i>	<i>0</i>
<i>0.07</i>	<i>-0.07</i>	<i>velocity</i>	<i>1</i>

عامل هوشمند در هر مرحله، مجاز به انجام سه حرکت *push right*، *push left* و *no push* است. حرکت عامل به محیط داده شده و محیط حالت بعد را به همراه پاداش حرکت برمی‌گرداند. برای هر گامی که ماشین به هدف نمی‌رسد، هزینه ۱- در نظر گرفته شده است. با استفاده از *Q-learning*، سیاست بهینه را در هر حالت بیابید. برای انجام اینکار باید تابع‌های موجود در فایل پیوست را کامل کنید. در صورتی که با بهینه کردن پیاده‌سازی خود، نتیجه‌ها را بهبود داده و دست‌کم به مقدار ۱۲۰- برای خروجی تابع *Score* برسید، نمره امتیازی‌ای متناسب با کیفیت پیاده‌سازی برای شما در نظر گرفته می‌شود. (۱۵ نمره + ۱۰ نمره امتیازی)

پاینده باشید

^۶Reinforcement Learning