به نام خدا

درس یادگیری ماشین
نیم‌سال دوم ۰۳-۰۴
استاد: ربیعی_ دهقانیان

دانشکده مهندسی کامپیوتر

آزمونک پنجم - ۱۴۰۴/۳/۱۱ زمان آزمون: ۴۰ دقیقه

۱. (۲۵ نمره) فرض کنید ماتریس داده زیر را داریم که نمایانگر چهار نقطهٔ نمونه X_i در $\mathbb{R}^2$ است:

$$X = \begin{pmatrix} 4 & 1 \\ 2 & 3 \\ 5 & 4 \\ 1 & 2 \end{pmatrix}$$

ما می‌خواهیم داده‌ها را تنها در یک بُعد نمایش دهیم، بنابراین از روش تحلیل مولفه‌های اصلی (PCA) استفاده می‌کنیم.

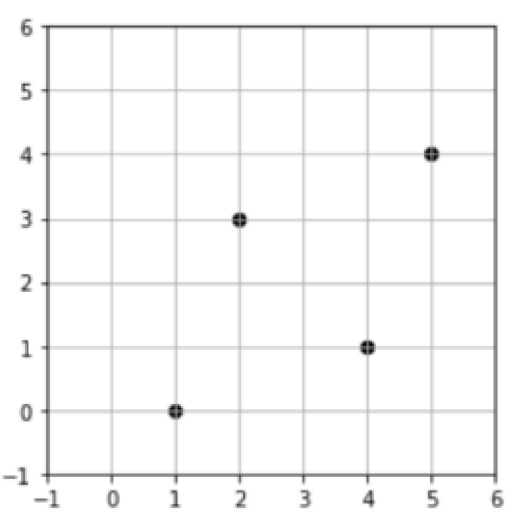
(الف) جهت‌های مؤلفهٔ اصلی با طول واحد (unit-length principal component directions) را برای X محاسبه کنید و مشخص نمایید که اگر فقط یک مؤلفهٔ اصلی درخواست شود، الگوریتم PCA کدام جهت را انتخاب می‌کند. پاسخ خود را به‌صورت دقیق و بدون تقریب بیان کنید (نیاز به نماد $\sqrt{\cdot}$ دارید). مراحل و محاسبات خود را نشان دهید.

(ب) نمودار زیر نقاط نمونه را از X نشان می‌دهد. می‌خواهیم نمایش یک‌بعدی از داده‌ها داشته باشیم. بنابراین:

۱. جهت مؤلفهٔ اصلی انتخاب‌شده را به‌صورت یک خط بر روی نمودار رسم کنید.

۲. تصویر (پروجکشن) هر یک از چهار نقطهٔ نمونه را روی این جهت محاسبه و رسم کنید.

۳. هر نقطهٔ پروجکت‌شده را با مقدار مختصات آن روی مؤلفهٔ اصلی برچسب‌گذاری کنید (مبدأ مختصات بر روی خط مؤلفهٔ اصلی برابر صفر در نظر گرفته شود). مقادیر مختصات مؤلفه‌ها را به‌طور دقیق و بدون گرد کردن بیان کنید.



پاسخ:

(الف) برای محاسبه جهت‌های مؤلفه اصلی، مراحل زیر طی می‌شود:

۱. مرکزی کردن داده‌ها

ماتریس داده اولیه:

$$X = \begin{pmatrix} 4 & 1 \\ 2 & 3 \\ 5 & 4 \\ 1 & 2 \end{pmatrix}$$

میانگین هر ستون:

$$\mu_1 = \frac{4+2+5+1}{4} = 3, \quad \mu_2 = \frac{1+3+4+2}{4} = 2.5$$

بنابراین، بردار میانگین:

$$\mu = \begin{pmatrix} 3 \\ 2.5 \end{pmatrix}$$

داده‌های مرکزی‌شده ($X_{\text{centered}} = X - \mu$):

$$X_{\text{centered}} = \begin{pmatrix} 1 & -1.5 \\ -1 & 0.5 \\ 2 & 1.5 \\ -2 & -0.5 \end{pmatrix}$$

۲. محاسبه ماتریس کوواریانس

$$X_{\text{centered}}^T X_{\text{centered}} = \begin{pmatrix} 10 & 2 \\ 2 & 5 \end{pmatrix}, \quad C = \frac{1}{3}\begin{pmatrix} 10 & 2 \\ 2 & 5 \end{pmatrix} = \begin{pmatrix} \frac{10}{3} & \frac{2}{3} \\ \frac{2}{3} & \frac{5}{3} \end{pmatrix}$$

۳. محاسبه مقادیر ویژه و بردارهای ویژه

حل معادله مشخصه:

$$\det(C - \lambda I) = \lambda^2 - 5\lambda + \frac{46}{9} = 0 \Rightarrow 9\lambda^2 - 45\lambda + 46 = 0$$

دیسکریمینانت:

$$\Delta = 45^2 - 4 \cdot 9 \cdot 46 = 369 \Rightarrow \sqrt{369} = 3\sqrt{41}$$

ریشه‌ها:

$$\lambda_{1,2} = \frac{15 \pm \sqrt{41}}{6}$$

بردار ویژه متناظر با λ_1:

$$\mathbf{v}_1 = \begin{pmatrix} 4 \\ \sqrt{41} - 5 \end{pmatrix}, \quad \|\mathbf{v}_1\| = \sqrt{82 - 10\sqrt{41}}, \quad \mathbf{u}_1 = \frac{1}{\sqrt{82 - 10\sqrt{41}}}\begin{pmatrix} 4 \\ \sqrt{41} - 5 \end{pmatrix}$$

بردار متعامد (برای λ_2):

$$\mathbf{v}_2 = \begin{pmatrix} 5-\sqrt{41} \\ 4 \end{pmatrix}, \quad \|\mathbf{v}_2\| = \sqrt{82-10\sqrt{41}}, \quad \mathbf{u}_2 = \frac{1}{\sqrt{82-10\sqrt{41}}}\begin{pmatrix} 5-\sqrt{41} \\ 4 \end{pmatrix}$$

(ب) **۱. رسم جهت مؤلفه اصلی اول**

جهت مؤلفه اصلی اول به صورت خطی از نقطه میانگین عبور می‌کند:

$$y - 2.5 = \frac{\sqrt{41}-5}{4}(x-3)$$

۲. محاسبه تصاویر نقاط

نقاط مرکزی‌شده:

$$\mathbf{x}_1 - \mu = \begin{pmatrix} 1 \\ -1.5 \end{pmatrix}$$
$$\mathbf{x}_2 - \mu = \begin{pmatrix} -1 \\ 0.5 \end{pmatrix}$$
$$\mathbf{x}_3 - \mu = \begin{pmatrix} 2 \\ 1.5 \end{pmatrix}$$
$$\mathbf{x}_4 - \mu = \begin{pmatrix} -2 \\ -0.5 \end{pmatrix}$$

محاسبه ضرب داخلی با $\mathbf{u}_1$:

$$\mathbf{x}_1 : \frac{1.5(7-\sqrt{41})}{\sqrt{82-10\sqrt{41}}}$$
$$\mathbf{x}_2 : \frac{0.5(-13+\sqrt{41})}{\sqrt{82-10\sqrt{41}}}$$
$$\mathbf{x}_3 : \frac{0.5(1+3\sqrt{41})}{\sqrt{82-10\sqrt{41}}}$$
$$\mathbf{x}_4 : \frac{0.5(-11-\sqrt{41})}{\sqrt{82-10\sqrt{41}}}$$

۳. مختصات نقاط تصویرشده (بدون نرمال‌سازی)

نقطه	مختصات روی مؤلفه اصلی
$(4,1)$	$\frac{7-\sqrt{41}}{4}$
$(2,3)$	$\frac{-1+\sqrt{41}}{4}$
$(5,4)$	$\frac{13+\sqrt{41}}{4}$
$(1,2)$	$\frac{-11-\sqrt{41}}{4}$

۲. (۳۵ نمره) فرض کنید یک مسئلهٔ دسته‌بندی متون داریم، به‌طوری که یک سند X را می‌توان به‌صورت بردار باینری ویژگی‌های کلمات نمایش داد. به بیان دقیق‌تر:

$$X = [X_1, X_2, X_3, \ldots, X_m]$$

که

$$X_j = \begin{cases} 1 & \text{اگر کلمهٔ } j \text{ در سند } X \text{ حضور دارد} \\ 0 & \text{در غیر این صورت} \end{cases}$$

با استفاده از الگوریتم AdaBoost و یک یادگیرندهٔ ضعیف ساده تعریف‌شده به‌صورت زیر، می‌خواهیم سندها را به یکی از دو کلاس $\{-1, +1\}$ دسته‌بندی کنیم:

$$h(X; \theta) = yX_j,$$

که $\theta = (j, y)$ مشخصهٔ "کلمهٔ انتخاب‌شده" و "برچسب کلاس" برای آن است. دقت کنید که $y \in \{-1, +1\}$ است. به‌طور شهودی، برای هر کلمه دو یادگیرندهٔ ضعیف داریم:

- پیش‌بینی sports اگر سند شامل کلمهٔ football باشد.
- پیش‌بینی non-sports اگر سند شامل کلمهٔ football باشد.

(الف) تعداد کل یادگیرنده‌های ضعیف چند عدد است؟
پاسخ: دو یادگیرنده ضعیف برای هر کلمه، یعنی $2m$ یادگیرنده ضعیف.

(ب) این الگوریتم افزایشی را می‌توان برای انتخاب ویژگی‌ها نیز به‌کار برد. ویژگی‌ها را به ترتیبی که الگوریتم AdaBoost شناسایی می‌کند، انتخاب می‌کنیم.

۱. آیا ممکن است الگوریتم، یک یادگیرندهٔ ضعیف را بیش از یک‌بار انتخاب کند؟ توضیح دهید.
پاسخ: الگوریتم بوستینگ هر ضریب جدید α را با فرض اینکه تمام ضرایب قبلی ثابت باقی می‌مانند بهینه می‌کند. بنابراین، این الگوریتم این ضرایب را به‌صورت مشترک بهینه نمی‌کند. تنها راه برای اصلاح آرا اختصاص‌یافته به یک یادگیرنده ضعیف در مراحل بعدی، افزودن دوباره همان یادگیرنده ضعیف است. از آنجا که در اینجا فقط مجموعه‌ای گسسته از یادگیرنده‌های ضعیف داریم، صحبت از انتخاب دوباره دقیقاً همان یادگیرنده ضعیف نیز منطقی به نظر می‌رسد.

۲. اگر ویژگی‌ها را بر اساس اطلاعات متقابل $I(i; Y)$ مرتب کنیم، آیا این ترتیب می‌تواند بهتر و آگاهانه‌تر از ترتیب به‌دست‌آمده توسط AdaBoost باشد؟ توضیح دهید.
پاسخ: الگوریتم بوستینگ ترکیبی خطی از طبقه‌بندهای ضعیف (که در اینجا ویژگی‌ها هستند) ایجاد می‌کند. بنابراین، الگوریتم هر طبقه‌بند ضعیف جدید (ویژگی) را نسبت به پیش‌بینی خطی که بر اساس ویژگی‌های از پیش گنجانده‌شده انجام شده، ارزیابی می‌کند. معیار اطلاعات متقابل هر ویژگی را به‌صورت جداگانه بررسی می‌کند و به همین دلیل نمی‌تواند تشخیص دهد که چگونه تعامل چندین ویژگی ممکن است به بهبود پیش‌بینی خطی کمک کند.

۳. (۴۰ نمره) درستی یا نادرستی گزاره‌های زیر را با ذکر دلیل مشخص کنید:

(الف) الگوریتم k-NN یک روش غیرپارامتریک است که در مرحلهٔ پیش‌بینی از تمام داده‌های آموزشی به‌طور مستقیم استفاده می‌کند.
پاسخ: درست. الگوریتم kNN- یک روش غیرپارامتریک است، یعنی نیازی به فرض توزیع خاصی برای داده‌ها ندارد. این الگوریتم در زمان پیش‌بینی، مستقیماً از داده‌های آموزشی استفاده می‌کند و k همسایه نزدیک را پیدا می‌کند. به جای ساختن یک مدل کلی، تصمیم‌گیری بر اساس نزدیکی داده‌ها انجام می‌شود.

(ب) در روش «رگرسیون خطی وزن‌دار محلی» (Locally Weighted Linear Regression)، به دلیل وزن‌دهی محلی، مدل قادر است خارج از ناحیهٔ نمونه‌های آموزشی به‌درستی تعمیم یابد.

پاسخ: نادرست. در رگرسیون خطی وزن‌دار محلی، برای هر پیش‌بینی یک مدل خطی با وزن‌دهی به داده‌های نزدیک ساخته می‌شود. این روش در محدوده داده‌های آموزشی دقیق است، اما خارج از آن تعمیم ضعیفی دارد. دلیلش این است که وزن‌ها با دور شدن از داده‌ها کاهش می‌یابد و پیش‌بینی بی‌معنی می‌شود.

(ج) در الگوریتم Value Iteration، در صورتی که سیاست به‌دست‌آمده برای یک استیت در T مرحله تغییر نکند، سیاست به‌دست‌آمده در این T مرحله همان سیاست بهینه خواهد بود. (به‌طور دقیق‌تر اگر داشته باشیم $\pi_k = \pi_{k+1} = \cdots = \pi_{k+T}$, آنگاه برای هر $i = 0, 1, \ldots, T$، π_{k+i} همان سیاست بهینه است.)

پاسخ: نادرست. MDP ای را در نظر بگیرید که در آن state ها، خانه‌های یک نوار با $T+3$ خانه هستند. همچنین برای این MDP داریم (دقت کنید که سمت راست s_1، استیت s_2 قرار گرفته است):

$$S = \{s_1, s_2, \ldots, s_{T+1}, s_{T+2}, s_{T+3}\}, \quad A = \{r(\text{move to right})\,, l(\text{move to left})\,\}$$

تابع انتقال:

$$T(s_i, D, s_j) = \begin{cases} 1 & D = r \wedge 1 \leq i \leq T+1 \wedge j = i+1 \\ 1 & D = l \wedge i = 1 \wedge j = T+3 \\ 0 & \text{در غیر این صورت} \end{cases}$$

تابع پاداش:

$$R(s_i, D, s_j) = \begin{cases} 10 & D = r \wedge j = i+1 = T+2 \\ 1 & D = l \wedge i = 1 \wedge j = T+3 \\ 0 & \text{در غیر این صورت} \end{cases}$$

$$(\text{start state, final states}) = (s_1, \{s_{T+2}, s_{T+3}\})$$

به وضوح در این MDP داریم: (برای همهٔ استیت‌های غیرترمینال، سیاست بهینه حرکت به سمت راست است)

$$\forall 1 \leq i \leq T+1; \quad V^*(s_i) = 10 \Rightarrow \pi^*(s_i) = r$$

حال دقت کنید که در حین اجرای value iteration برای این MDP خواهیم داشت (با $\gamma = 1$):

$$\forall 1 \leq i \leq T+1; \quad V_0(s_i) = 0$$

$$V_{k+1}(s_i) = \max_a Q(s_i, a) = Q(s_i, r) = \sum_j T(s_i, r, s_j)\left(R(s_i, r, s_j) + \gamma V_k(s_j)\right)$$

بنابراین می‌توان نوشت:

$$\forall 1 \leq i \leq T+1; \quad V_k(s_i) = \begin{cases} 10 & k \geq T+2-i \\ 1 & i = 1 \wedge 1 \leq k \leq T \\ 0 & \text{در غیر این صورت} \end{cases}$$

بنابراین در T مرحلهٔ ابتدایی اجرای value iteration بر روی این MDP، سیاست به‌دست‌آمده برای استیت s_1، حرکت به سمت چپ می‌باشد (با مقدار value برابر ۱) که با سیاست بهینه برای این استیت (حرکت به سمت راست) یکسان نمی‌باشد. پس این عبارت نادرست است.

(د) الگوریتم Q-learning تنها زمانی مقادیر بهینه (Q-values) را یاد می‌گیرد که کنش‌هایی (actions) که در نهایت انتخاب می‌شوند براساس سیاست (policy) بهینه باشند.

پاسخ: نادرست. تا زمانی که سیاست مورد استفاده به گونه‌ای باشد که همه‌ی حالات را پیدا کند، Q-learning حتی با یک سیاست رندوم نیز کار می‌کند. Q-values در نهایت به مقادیر بهینه همگرا خواهند شد.

(ه) در یک MDP قطعی (یعنی به ازای هر state و action یک state مشخص به‌صورت قطعی رخ می‌دهد) الگوریتم Q-learning با استفاده از نرخ یادگیری $\alpha = 1$ در هر بروزرسانی، به‌طور صحیح مقادیر Q-values را یاد می‌گیرد.

پاسخ: درست. می‌دانیم نرخ یادگیری در این الگوریتم تنها به این علت وجود دارد که، برای به‌روزرسانی Q-values از یک نمونه (sample) استفاده می‌شود که به‌صورت قطعی نمی‌دانیم MDP به کدام s' انتقال خواهد یافت، بنابراین برای تخمین آن نیاز به وزن‌دهی داریم. در به‌روزرسانی Q، برای یک حالت s و اکشن a خواهیم داشت:

$$Q(s,a) = R(s,a,s') + \gamma \max_{a'} Q(s',a')$$

که دقیقاً همان رابطه به‌روزرسانی است که در Q-learning انجام می‌دهیم.

(ی) در یک MDP با حالات محدود و پاداش با باند مشخص و $\gamma < 1$، اگر همهٔ پاداش‌ها با عدد ثابت c جمع شوند، سیاست بهینه تغییر نمی‌کند.

پاسخ: درست. افزودن یک مقدار ثابت به تمام پاداش‌ها تأثیری بر روی ترتیب نسبی ارزش‌های حالت‌ها ندارد. (البته اگر فرض می‌کردیم که MDP می‌تواند یک حالت terminal داشته باشد، گزاره ممکن بود نادرست شود. پس به پاسخ‌هایی که برای این حالت مثال نقض آورده‌اند نیز نمره داده شود.)