به نام خدا

درس یادگیری ماشین

تمرین دوم

حدیثه مصباح

۸۱۰۱۰۲۲۵۳

پاسخ ۱.

۱-۱. مفهوم bias-variance trade off را با توجه به h_n در روش پارزن و k_n در روش KNN توضیح دهید.

۱-۲. نشان دهید که مدل KNN برای ($K \neq 1$) تابع توزیع نامناسبی را تعریف میکند که انتگرال آن در تمام فضا واگرا میباشد.

(۱.۱

بایاس و واریانس دو مفهوم مهم در یادگیری ماشین هستند که بر دقت پیش‌بینی‌های مدل تأثیر می‌گذارند. بایاس به خطای ناشی از ساده‌سازی بیش از حد مدل اشاره دارد که باعث می‌شود مدل نتواند روابط پیچیده میان داده‌ها را درک کند. واریانس به خطای ناشی از حساسیت بیش از حد مدل به داده‌های آموزشی اشاره دارد که موجب می‌شود مدل به نویز و نوسانات جزئی داده‌ها واکنش نشان دهد.

در مدل‌سازی، ما با دو مشکل عمده روبرو هستیم: underfitting و overfitting. Underfitting زمانی رخ می‌دهد که مدل نتواند داده‌های آموزشی را به خوبی بیاموزد و معمولاً با بایاس بالا و واریانس پایین همراه است. Overfitting زمانی اتفاق می‌افتد که مدل بیش از حد به داده‌های آموزشی وابسته شود و نتواند به خوبی به داده‌های جدید تعمیم داده شود، که این معمولاً با بایاس پایین و واریانس بالا همراه است.

در روش KNN، تعداد نزدیکترین همسایگان (k_n) تعیین می‌کند که چگونه مدل به داده‌های آموزشی واکنش نشان می‌دهد. هرچه k_n کوچک‌تر باشد، مدل حساس‌تر به داده‌های آموزشی است و ممکن است واریانس بیشتری داشته باشد. با افزایش k_n، مدل کمتر حساس می‌شود و بایاس بیشتری خواهد داشت.

در روش پارزن، پارامتر h_n که عرض کرنل را تعیین می‌کند، بر تعادل بین بایاس و واریانس تأثیر می‌گذارد. یک h_n بزرگ‌تر منجر به یک تخمین هموارتر و بایاس بیشتر می‌شود، در حالی که یک h_n کوچک‌تر باعث می‌شود مدل انعطاف‌پذیرتر باشد اما واریانس بیشتری داشته باشد.

بنابراین، در انتخاب مدل و تنظیم فراپارامترها، یافتن تعادل مناسب بین بایاس و واریانس برای ساخت مدلی که بتواند به خوبی به داده‌های دیده نشده تعمیم دهد، حیاتی است. این مصالحه یکی از ملاحظات کلیدی در یادگیری ماشین است.

(۱.۲

مدل KNN (k-nearest neighbors) یک روش غیرپارامتریک در یادگیری ماشین است که برای دسته‌بندی و رگرسیون استفاده می‌شود. اگر بخواهیم از KNN برای تخمین تابع چگالی احتمال استفاده کنیم، باید دقت داشته باشیم که تابع توزیعی که KNN تولید می‌کند (برای $K \neq 1$) ممکن است به‌طور کلی به عنوان یک تابع توزیع احتمال مناسب نباشد. این به دلیل آن است که تابع توزیع تخمین زده شده توسط KNN ممکن است انتگرالی واگرا داشته باشد وقتی $K \neq 1$ است.

$$\int p(x)dx \approx \sum_{i=1}^{N} p(x_i).V_i = \sum_{i=1}^{N} \frac{K}{NV_i}.V_i = k \neq 1|$$

این مشکل از این واقعیت نشأت می‌گیرد که KNN برای هر نقطه‌ای در فضای ویژگی یک کره می‌کشد و تعداد ثابتی از نزدیک‌ترین همسایه‌ها را در نظر می‌گیرد، نه یک فاصله ثابت. وقتی $K \neq 1$، حجم این کره‌ها به تعداد نقاط داده‌ای که در آن‌ها وجود دارد بستگی دارد، و این حجم‌ها می‌توانند بسیار نامتناسب باشند.

برای نمونه، در مناطقی که داده‌ها خیلی پراکنده هستند، KNN ممکن است کره‌های بزرگی را تولید کند که برای حفظ تعداد ثابت K از همسایه‌ها مورد نیاز است. در نتیجه، این مناطق دورافتاده وزن بیشتری در تخمین چگالی احتمال خواهند داشت، که می‌تواند منجر به تخمینی واگرا شود، زیرا وقتی انتگرال تابع توزیع چگالی احتمال را برای تمام فضا محاسبه کنیم، ممکن است مجموع وزن‌های مناطق پراکنده به بی‌نهایت میل کند.

در مقابل، در مناطق پرجمعیت داده‌ای، کره‌ها می‌توانند بسیار کوچک باشند، که این موضوع نیز می‌تواند باعث شود که تخمین چگالی احتمال در این مناطق بیش از حد بالا رود. این ناهماهنگی در اندازه کره‌ها می‌تواند به این معنی باشد که تابع چگالی احتمال تخمین زده شده توسط KNN برای $K \neq 1$ نمی‌تواند به درستی بر روی تمام فضای ویژگی انتگرال‌گیری شود و در نتیجه، یک تابع توزیع احتمال مناسب نخواهد بود.

پاسخ۲ .

توزیع یکنواخت $p(x)$ و پنجره پارزن $\varphi(x)$ به صورت زیر تعریف شده است.

$$p(x) \sim U(0,a)$$

$$\varphi(x) = \begin{cases} e^{-x} ; x > 0 \\ 0 ; x \leq 0 \end{cases}$$

۲-۱. نشان دهید که میانگین چنین تخمینی از پنجره پارزن به صورت زیر میشود.

$$\bar{p}_n(x) = \begin{cases} 0 ; x < 0 \\ \frac{1}{a}\left(1 - e^{-\frac{x}{h_n}}\right); 0 \leq x \leq a \\ \frac{1}{a}\left(e^{\frac{a}{h_n}} - 1\right) e^{-\frac{x}{h_n}} ; a \leq x \end{cases}$$

۲-۲. $\bar{p}_n(x)$ را بر حسب x برای $a = 1$ و $h_n = \{1 \ , \frac{1}{4} \ , \frac{1}{16}\}$ رسم کنید.

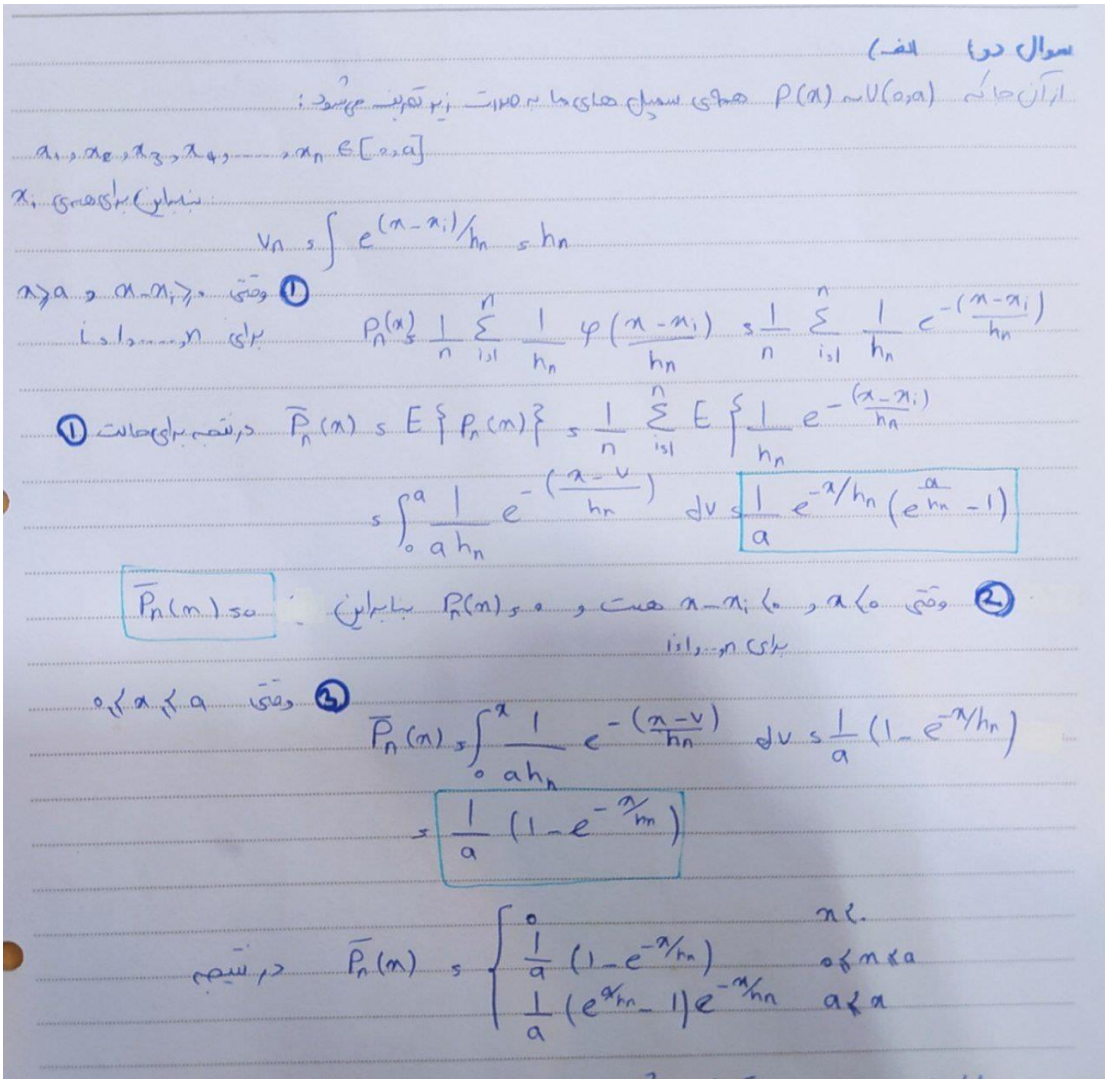
۲-۳. h_n چه قدر باشد تا در بازه $0 < x < a$ مقدار بایاس کمتر از ۱ درصد باشد.

۲-۴. در شرایط h_n بخش ۳-۵ و مقدار $a = 1$ ، $\bar{p}_n(x)$ را در بازه $0 < x < 0.05$ رسم کنید.

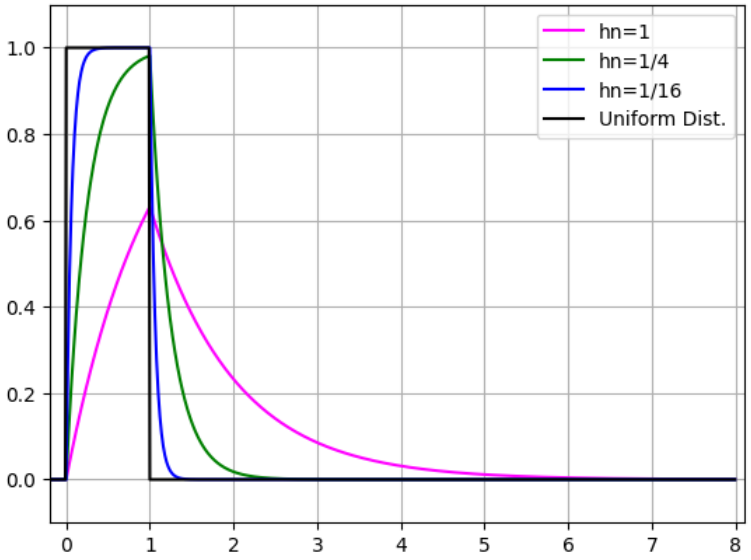
(۲.۱

سوال دو

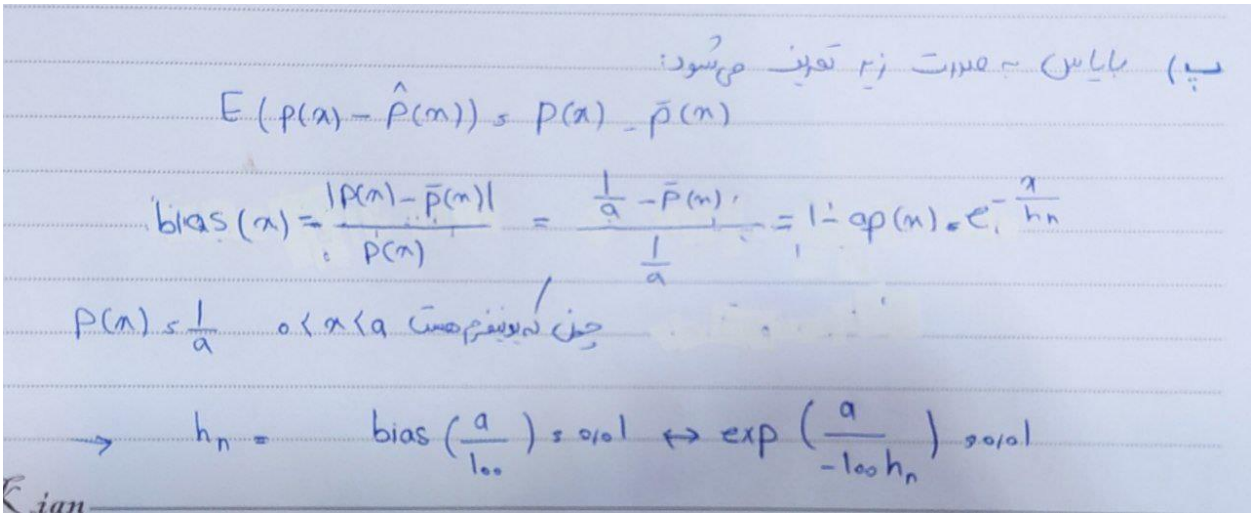
الف)

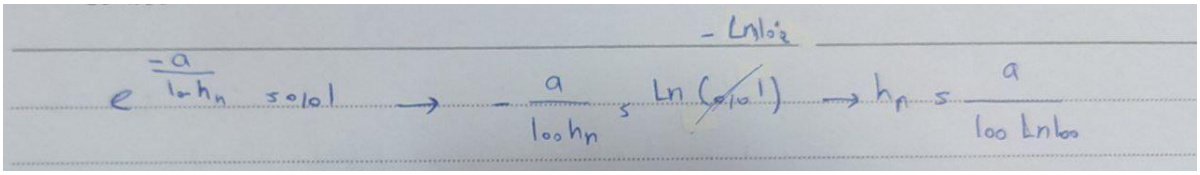


(۲.۲

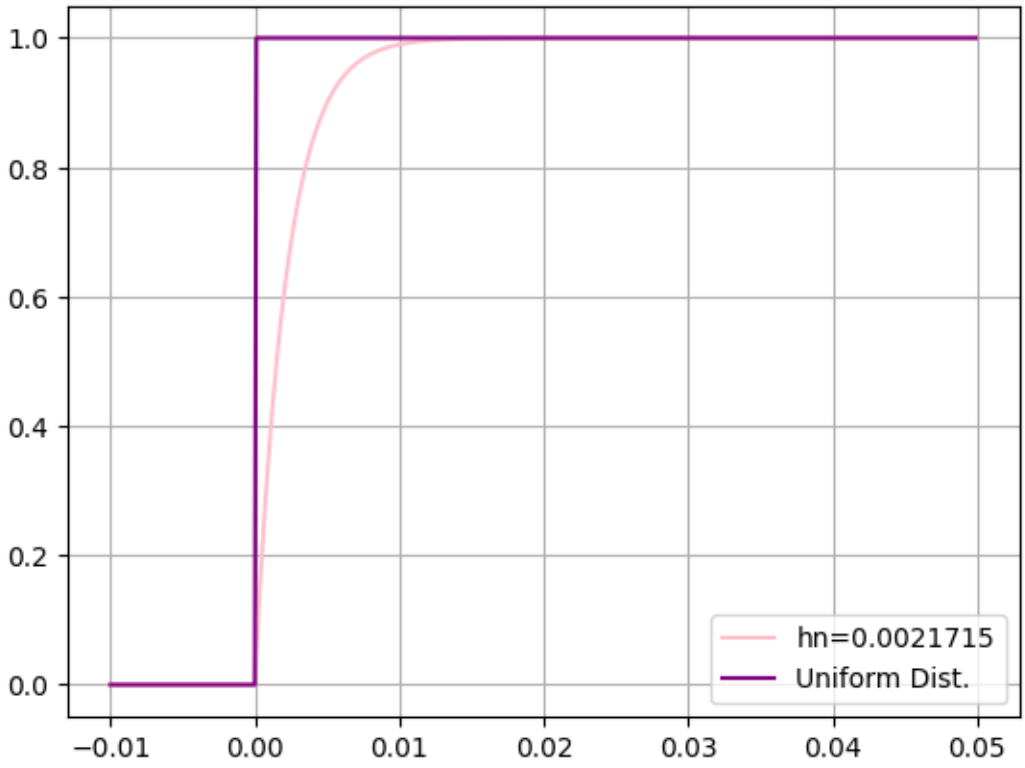


(۲.۳


پ) بایاس = عبارت زیر تعریف می‌شود:
$E(p(x)-\hat{p}(x)) = p(x)-\bar{p}(x)$
$bias(x) = \frac{|p(x)-\bar{p}(x)|}{p(x)} = \frac{\frac{1}{a}-\bar{p}(x)}{\frac{1}{a}} = 1-ap(x) = e^{-\frac{x}{h_n}}$
$p(x) = \frac{1}{a}$ $0<x<a$ چون توزیع یکنواخت است
$\rightarrow h_n = bias\left(\frac{a}{100}\right) = 0.01 \leftrightarrow \exp\left(\frac{a}{-100h_n}\right) = 0.01$


$-\ln 100$
$e^{\frac{-a}{100h_n}} = 0.01 \rightarrow -\frac{a}{100h_n} = \ln(0.01) \rightarrow h_n = \frac{a}{100\ln 100}$

(۲.٤



پاسخ 3.

توزیع نرمال $p(x) \sim N(\mu, \sigma^2)$ و پنجره پارزن $\varphi(x) \sim N(0,1)$ را در نظر بگیرید. نشان دهید که تخمین پنجره پارزن $p(x) = \frac{1}{nh_n}\sum_{i=1}^{n}\varphi(\frac{x-x_i}{h_n})$ برای h_n های کوچک دارای ویژگی های زیر است :

- $\tilde{p_n}(x) \sim N(\mu, h_n^2 + \sigma)$
- $p_n(x) - \tilde{p_n}(x) \cong \frac{1}{2}\left(\frac{h_n}{\sigma}\right)^2 \left[1 - \left(\frac{x-\mu}{\sigma}\right)^2\right] p(x)$
- $var[p_n(x)] \cong \frac{1}{2nh_n\sqrt{\pi}} p(x)$

سوال ۳ :

$$\tilde{P_n}(x) \sim N(\mu, h_n^2 + \sigma)$$

$$\bar{P}_n(x) = E[P_n(x)] = \frac{1}{n}\sum_{i=1}^{n} E\left[\frac{1}{h_n}\varphi\left(\frac{x-x_i}{h_n}\right)\right] \rightarrow$$

$$\bar{P}_n(x) = \int \frac{1}{h_n}\varphi\left(\frac{x-x'}{h_n}\right) p(x')\,dx'$$

$$\xrightarrow{iid} \varphi = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}(x^2)} \qquad P(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$

$$\rightarrow \bar{P}_n(x) = \frac{1}{h_n}\int_{-\infty}^{+\infty} \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}\left(\frac{x-x'}{h_n}\right)^2\right) \times \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{1}{2}\left(\frac{x'-\mu}{\sigma}\right)^2\right) dx'$$

e توان عبارت $\bar{P}_n(x)$ $\rightarrow$ $-\frac{1}{2}\left[\frac{x'^2(\sigma^2+h_n^2) - 2x'(x\sigma^2+\mu h_n^2) + x^2\sigma^2 + h_n^2\mu^2}{h_n^2\sigma^2}\right]$

برای راحتی حل بیشتر متغیر تغییر می‌دهیم:

$$\alpha = \frac{x\sigma^2+\mu h_n^2}{\sqrt{\sigma^2+h_n^2}} \qquad \beta = x^2\sigma^2 + h_n^2\mu^2 \qquad y = x'\sqrt{\sigma^2+h_n^2}$$

① $-\frac{1}{2}\left(\frac{y^2 - 2x'\left(\frac{a\sigma^2+\mu h_n^2}{\sqrt{\sigma^2+h_n^2}}\right)+\beta}{h_n^2\sigma^2}\right) = -\frac{1}{2}\left(\frac{(y-a)^2 - a^2 + b^2 + \beta}{h_n^2\sigma^2}\right)$ سپس $(y-a)^2 + c$ می‌خواهیم توان e شود

② $dx' = \frac{1}{\sqrt{\sigma^2+h_n^2}}\,dy$

①② $\frac{1}{\sqrt{2\pi}\sigma}\frac{1}{\sqrt{2\pi}h_n}\frac{1}{\sqrt{\sigma^2+h_n^2}} e^{-\frac{1}{2}\left(\frac{-a^2+\beta}{h_n^2\sigma^2}\right)^2} \underbrace{\int e^{-\frac{1}{2}\left(\frac{(y-a)^2}{h_n^2\sigma^2}\right)} dy}_{\sqrt{2\pi}h_n\sigma}$

$$* \text{ عبارت} = \frac{1}{\sigma^2+h_n^2}\left(x^2 + 2\mu x + \mu^2\right) = \frac{(x-\mu)^2}{\sigma^2+h_n^2}$$

بعد از ساده‌سازی و حذف کردن x از انتگرال

$$\tilde{P}_n(x) = \frac{1}{\sqrt{2\pi}}\frac{1}{\sqrt{\sigma^2+h_n^2}} e^{-\frac{1}{2}\left(\frac{(x-\mu)^2}{\sigma^2+h_n^2}\right)}$$

$\tilde{P}_n(x)$ دارای توزیع $N(\mu, \sigma^2+h_n^2)$ است

Kian

$$P_n(x) - \bar{P}_n(x) \le \frac{1}{2}\left(\frac{h_n}{\sigma}\right)^2\left[1-\left(\frac{x-\mu}{\sigma}\right)^2\right]p(x)$$

① براساس فرضیات سوال قبلی (بخش قبلی)

$$P(x) - \bar{P}_n(x) = \frac{1}{\sqrt{2\pi}\,\sigma} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2} - \frac{1}{\sqrt{h_n^2+\sigma^2}}\frac{1}{\sqrt{2\pi}}\exp\left(-\frac{1}{2}\left(\frac{(x-\mu)^2}{h_n^2+\sigma^2}\right)\right)$$

$$\frac{\bar{P}_n(x)}{P(x)} = \frac{\sigma}{\sqrt{\sigma^2+h_n^2}}\exp\left[-\frac{(x-\mu)^2}{2}\left(\frac{1}{h_n^2+\sigma^2}-\frac{1}{\sigma^2}\right)\right]$$

① حاشیه‌ای در راضی

$$P(x) - \bar{P}_n(x) = P(x)\left[1-\frac{\sigma}{\sqrt{\sigma^2+h_n^2}}\exp\left[\frac{(x-\mu)^2}{2}\left(\frac{1}{h_n^2+\sigma^2}-\frac{1}{\sigma^2}\right)\right]\right]$$

$$\frac{1}{\sqrt{1+\left(\frac{h_n}{\sigma}\right)^2}}$$

برای h_n های کوچک براساس راضی می‌توان نوشت:

$$\frac{1}{\sqrt{1+\left(\frac{h_n}{\sigma}\right)^2}} \approx 1-\frac{1}{2}\left(\frac{h_n}{\sigma}\right)^2$$

و با تقریب سری تیلور تابع نمایی داریم:

$$e^{\text{توان}} \simeq 1+\frac{h_n^2(x-\mu)^2}{\sigma^2(h_n^2+\sigma^2)}$$

با استفاده از جبر خطی داریم:

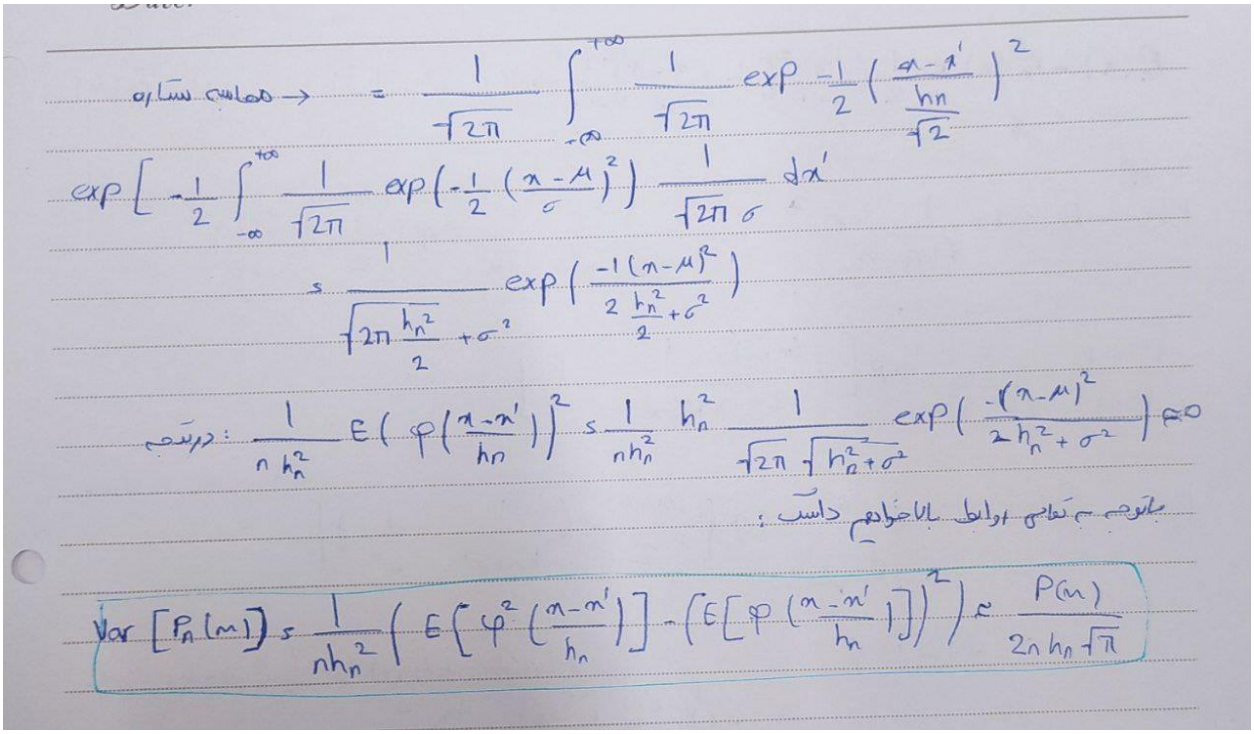
$$P(x) - \bar{P}_n(x) = \left(1-\left(1-\frac{1}{2}\left(\frac{h_n}{\sigma}\right)^2\right)\left(1+\frac{h_n^2(x-\mu)^2}{2\sigma^2(h_n^2+\sigma^2)}\right)\right) = \frac{1}{2}\left(\frac{h_n}{\sigma}\right)^2\left[1-\frac{(x-\mu)^2}{(h_n^2+\sigma^2)}\right]P(x)$$

$$P(x) - \bar{P}_n(x) \simeq \frac{1}{2}\left(\frac{h_n}{\sigma}\right)^2\left[1-\left(\frac{x-\mu}{\sigma}\right)^2\right]P(x)$$

$$\mathrm{Var}(P(x)) \le \frac{1}{2nh_n\sqrt{\pi}}P(x)$$

$$\mathrm{Var}[\bar{P}_n(x)] = \mathrm{Var}\left[\frac{1}{nh_n}\sum_{i=1}^{n}\varphi\left(\frac{x-x_i}{h_n}\right)\right] = \frac{1}{nh_n^2}\mathrm{Var}\left[\varphi\left(\frac{x-x'}{h_n}\right)\right]$$

$$= \frac{1}{nh_n^2}\left(\underbrace{E\left[\varphi^2\left(\frac{x-x'}{h_n}\right)\right]}_{*} - \underbrace{\left(E\left[\varphi\left(\frac{x-x'}{h_n}\right)\right]\right)^2}_{**}\right)$$



پاسخ ٤.

در هر ۲ سوال زیر متغیر های X و θ به ترتیب بیانگر نمونه های مشاهده شده و پارامتر های مسئله می باشند.

۴-۱. فرض کنید که توزیع پارامتر P(θ) می باشد. مراحل expectations و maximization را برای بیشینه کردن P(θ|X) بنویسید. (محاسبات را تنها به صورت پارامتری بنویسید.)

راهنمایی: از نامساوی جنسن می دانیم:

$$p(\theta \mid x) \propto p(x \mid \theta)p(\theta) \propto \left(\sum_z Q(z)\frac{p(x,z \mid \theta)}{Q(z)}\right)p(\theta)$$

$$ln(\sum_z Q(z)\frac{p(x,z \mid \theta)}{Q(z)}) \geq \sum_z Q(z)\{\ln(p(x,z \mid \theta)) - \ln(Q(z))\}$$

۴-۲. متغیرتصادفی X با ۴ حالت طبق جدول زیر مفروض است. فرض کنید θ یک عدد حقیقی در بازه [0,1] و احتمال هر حالت مطابق زیر میباشد.

State	Probability
A	$\frac{1}{3}$
B	$\frac{1}{3}(1-\theta)$
C	$\frac{2}{3}(\theta)$
D	$\frac{1}{3}(1-\theta)$

با فرض انجام n آزمایش روی X، حالت های A, B, C, D به تعداد n_a, n_b, n_c, n_d بار به دست آمده است. متاسفانه مقدار متغیر های n_a و n_c ناشناخته است. فرض کنید که تابع توزیع θ در ابتدا به صورت زیر نوشته شده است. مراحل E و M را نوشته و θ را بدست آورید.

$$p(\theta) = \frac{\Gamma_{(v_1+v_2)}}{\Gamma_{(v_1)}\,\Gamma_{(v_2)}}\theta^{v_1-1}(1-\theta)^{v_2-1}$$

راهنمایی: توجه کنید که داده های با لیبل های B و D پس از مشاهده مشخص است.

فرض کنید تابع درست‌نمایی به صورت زیر است:

$$L(\theta;\, x^1, \dots, x^m) = \prod_{i=1}^{m} p(x^i;\, \theta)$$

که در آن θ پارامترهای مدل و $x^1, \dots, x^m$ داده‌های مستقل در مجموعه آموزشی هستند. تابع $p(x^i;\, \theta)$ احتمال مشاهده داده x را با توجه به پارامترهای فعلی θ مشخص می‌کند.

اگر این تابع درست‌نمایی برای یک مدل مخلوط (مانند مدل با متغیرهای پنهان) باشد، می‌توانیم از الگوریتم EM (Expectation-Maximization) برای برازش پارامترها استفاده کنیم. در این روش، ما ابتدا انتظارات (یا تخمین‌هایی) برای داده‌های پنهان ایجاد می‌کنیم (مرحله E) و سپس با استفاده از این انتظارات، پارامترهای مدل را به‌روزرسانی می‌کنیم (مرحله M).

این فرایند به صورت تکراری ادامه می‌یابد تا زمانی که به یک تخمین ثابت برای پارامترها برسیم که دیگر تغییر قابل توجهی نکند. این تخمین‌های نهایی به ما اجازه می‌دهند تا مدل را بر اساس داده‌های موجود برازش دهیم.

تابع Loglikelihood به شرح زیر است:

$$l(\theta) = \sum_{i=1}^{m} logp(x;\theta) = \sum_{i=1}^{m} log \sum_{z} p(x,z;\theta)$$

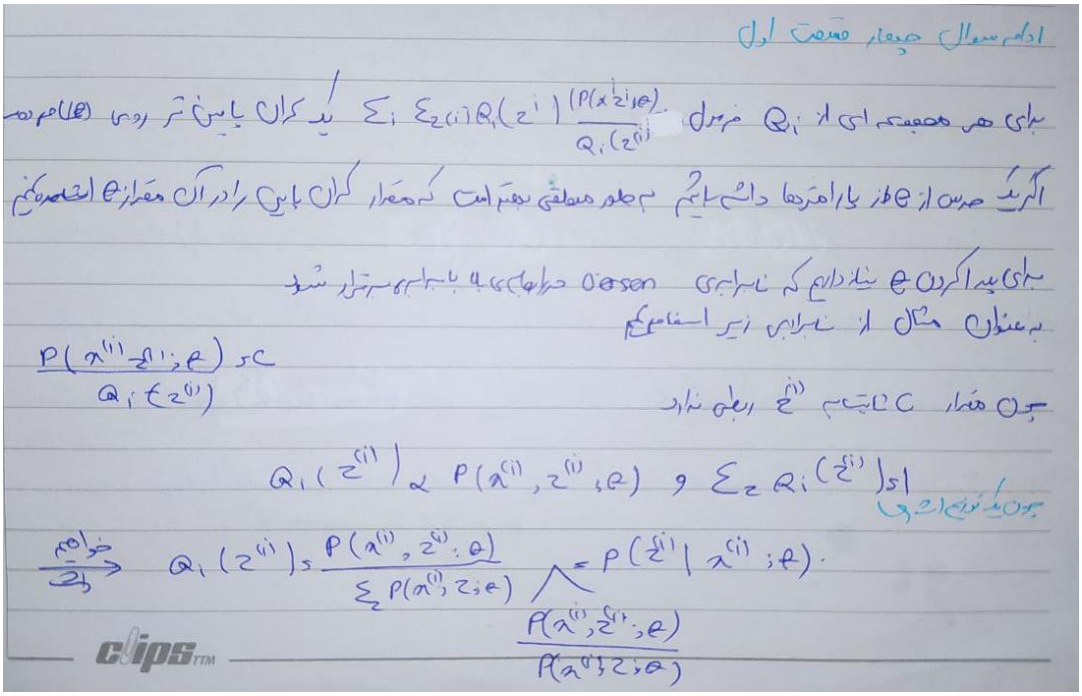
یافتن برآوردهای صریح درست‌نمایی حداکثری (ML) برای پارامترهای θممکن است دشوار باشد. در اینجا، $z^{(i)}$ متغیرهای تصادفی هستند که به آن‌ها دسترسی نداریم و اغلب این‌طور است که اگر $z^{(i)}$ مشاهده شود، تخمین ML آسان خواهد بود. در چنین مواقعی، الگوریتم EM یک روش کارآمد برای تخمین ML ارائه می‌دهد. برای هر i، Q_i مقداری توزیع بر روی z است $(\sum_z Q_i(z) = 1, Q_i(z) \geq 0)$حال روابط زیر را خواهیم داشت:

$$\sum_i \log\, p(x^{(i)};\theta) = \sum_i log \sum_{z^{(i)}} p\left(x^{(i)}, z^{(i)};\theta\right)$$
$$= \sum_i log \sum_{z^{(i)}} Q_i\left(z^{(i)}\right) \frac{\left(p\left(x^{(i)}, z^{(i)};\theta\right)\right)}{Q_i\left(z^{(i)}\right)}$$
$$\geq \sum_i \sum_{z^{(i)}} Q_i\left(z^{(i)}\right) \frac{\left(p\left(x^{(i)}, z^{(i)};\theta\right)\right)}{Q_i\left(z^{(i)}\right)}$$

آخرین مرحله این رویکرد این است که از نابرابری Jensen استفاده نماییم. به طور خاص، $f(x) = log\,(x)$ یک تابع مقعر است، زیرا $f''(x) = -\frac{1}{x^2} < 0$ در دامنه $x \in R^+$ منفی می‌باشد. همچنین، ترم $\sum_{z^{(i)}} Q_i\left(z^{(i)}\right) \frac{\left(p\left(x^{(i)}, z^{(i)};\theta\right)\right)}{Q_i\left(z^{(i)}\right)}$ در جمع سیگما فقط یک امید از کمیت $[\frac{p(x^{(i)}, z^{(i)};,\theta)}{Q_i(z^{(i)})}]$با توجه به $z^{(i)}$ بر اساس توزیع داده شده توسط Q_i ترسیم شده است. حابل با استفاده از نامساوی Jensen داده شده در صورت سوال خواهیم داشت:

$$f\left(E_{z^{(i)}\sim Q_i}\left[\frac{p\left(x^{(i)}, z^{(i)},;\theta\right)}{Q_i\left(z^{(i)}\right)}\right]\right) \geq E_{z^{(i)}\sim Q_i}\left[f\left(\frac{p\left(x^{(i)}, z^{(i)},;\theta\right)}{Q_i\left(z^{(i)}\right)}\right)\right]$$

بقیه ش توی عکسه وقت تایپ نبود 😊



Subject ..... Date .....

Q_i را به عنوان توزیع Posterior $z^{(i)}$ در نظر گرفته و با فرض پارامترهای θ داریم:

① مرحله E-step ← انتخاب Q_i از معادله $Q_i(z^{(i)}) = \frac{P(x^{(i)}, z^{(i)}; \theta)}{\sum_z P(x^{(i)}, z^{(i)}; \theta)}$ یا کران پایین را ماکزیمم کردیم

② مرحله M-step ← حول ثابت نگه داشتن Q_i پارامترها را ماکزیمم کردیم تا تنظیم θ های جدید به دست آوریم

کل این دو مرحله EM می‌شود که شروع می‌کنیم:

① $Q_i(z^{(i)}) = P(z^{(i)} | x^{(i)}; \theta)$ برای هر i

② $\theta = \arg\max_\theta \sum_i \sum_{z^{(i)}} Q_i(z^{(i)}) \log \frac{P(x^{(i)}, z^{(i)}; \theta)}{Q_i(z^{(i)})}$

تا همگرایی ادامه می‌دهیم

ثابت می‌کنیم که $\theta^{(t)}$ و $\theta^{(t+1)}$ پارامترهای متوالی EM هست، داریم:

$$\ell(\theta^{(t)}) \le \ell(\theta^{(t+1)})$$

که همان log likelihood ما را بهبود می‌بخشد.

با توجه به نتایج بالا و نابرابری Jensen خواهیم داشت:

$$\ell(\theta^{(t)}) = \sum_i \sum_{z^{(i)}} Q_i^{(t)}(z^{(i)}) \frac{P(x^{(i)}, z^{(i)}; \theta^{(t)})}{Q_i^{(t)}(z^{(i)})}$$

پارامتر $\theta^{(t+1)}$ با ماکزیمم ساختن سمت راست به دست می‌آید

$$\ell(\theta^{(t+1)}) \ge \sum_i \sum_{z^{(i)}} Q_i^{(t)}(z^{(i)}) \frac{P(x^{(i)}, z^{(i)}; \theta^{(t+1)})}{Q_i^{(t)}(z^{(i)})}$$

$$\rightarrow \ge \sum_i \sum_{z^{(i)}} Q_i^{(t)}(z^{(i)}) \frac{P(x^{(i)}, z^{(i)}; \theta^{(t)})}{Q_i^{(t)}(z^{(i)})} = \ell(\theta^{(t)})$$

نابرابری اول از این جا می‌آید که $\ell(\theta) \ge \sum_i \sum_{z^{(i)}} \frac{Q_i(z^{(i)}) P(x^{(i)}, z^{(i)}; \theta)}{Q_i(z^{(i)})}$ برای هر مقدار Q_i و θ صادق است و برای معادله بهبود ما این $Q_i = Q_i^{(t)}$ و $\theta = \theta^{(t+1)}$ استفاده کردیم که $\theta^{(t+1)}$ از راه‌کار ② به دست آمده است. در نتیجه $\theta^{(t+1)}$ از $\theta^{(t)}$ بزرگتر است.

و در نهایت مرحله‌ی استفاده شده برای به دست آوردن $\ell(\theta^{(t)})$ حالا انجام شد و از $Q_i^{(t)}$ برای برابر شدن نابرابری جنسن با برابر $\theta^{(t)}$ استفاده شد تا بهبود کند.

در نهایت این باعث می‌شود که احتمال به صورت یکنواخت همگرا شود و EM هم تا حدی همگرا که انجام می‌شود. و یک آزمون همگرایی متعارف این هست که بررسی کنیم آیا افزایش $L(\theta)$ بین تکرارهای متوالی کمتر از برخی پارامترهای محمول هست یا خیر و در نهایت اعلام همگرایی اگر EM باعث بهبود آرام $\ell(\theta)$ شود

$$J(Q,\theta) = \sum_i \sum_{z^{(i)}} Q_i(z^{(i)}) \left[\log(P(x^{(i)}, z^{(i)}; \theta)) - \log(Q_i(z^{(i)}))\right]$$

این رابطه را تخریب می‌کنیم

ما از روی رابطه‌های قبلی خود می‌دانیم که $\ell(\theta) \geq J(Q,\theta)$ و همچنین می‌دانیم EM را یک صعود مختصاتی در آن مشاهده کرد که در آن مرحله E آن را نسبت به Q به حداکثر می‌رساند و مرحله M آن را با توجه به θ به حداکثر می‌رساند

(۴.۲

سوال چهارم: قسمت ۲)

$L(\theta) = P(x, z \mid \theta) P(\theta)$

$\ln(L(\theta)) = \ln P(\theta) + \ln P(x, z \mid \theta)$

$P(x, z \mid \theta) = P(x \mid z, \theta) P(z \mid \theta)$

$\ln P(x, z, \theta) = \ln P(x \mid z, \theta) + \ln P(z \mid \theta)$

① $x = b$ $\quad \ln P(x = b \mid z, \theta) + \ln P(z \mid \theta) \rightarrow n_b \ln\left(\frac{1}{3}(1-\theta)\right)$

② $x = d$ $\quad \ln P(x = d \mid z, \theta) + \ln P(z \mid \theta) \rightarrow n_d \ln\left(\frac{1}{3}(1-\theta)\right)$

③ $z = a$, a or c $\quad \ln P(x = a \text{ or } c \mid z = a, \theta) + \ln P(z = a \mid \theta)$

حالت اول $\rightarrow (n_a + n_c)\left(1 + \ln \frac{1}{3}\right)$

$P(z = a \mid \theta) P(x = a \text{ or } c \mid z, \theta) = \frac{1}{1 + 2\hat{\theta}}$

نتیجه $\rightarrow \frac{1}{1 + 2\theta}(n_a + n_c)\left(1 + \ln\left(\frac{1}{3}\right)\right)$

④ $x = a$ or c, $z = c$ $\quad \ln(z = c \mid x = a)\left(\ln P(x = a \text{ or } c \mid z = c, \theta)\right) \; \ln P(z = c \mid \theta)$

$\frac{2\hat{\theta}}{1 + 2\hat{\theta}}(n_a + n_c) \ln\left(\frac{2}{3} \theta\right)$

Kian

Subject Date

$$\ln(L(\theta)) = \ln(P(\theta)) + (n_b + n_d)\ln\left(\frac{1}{3}(1-\theta)\right) + \frac{1}{1+2\hat{\theta}}(n_a + n_c) + \ln\left(\frac{1}{3}\right) + \frac{2\hat{\theta}}{1+2\hat{\theta}}(n-(n_b+n_d))\ln\left(\frac{2}{3}\theta\right) \quad (1)$$

$$\ln(P(\theta)) = \ln\left(\frac{\Gamma(v_1+v_2)}{\Gamma(v_1+v_2)}\theta^{v_1-1}(1-\theta)^{v_2-1}\right)$$

$$\ln(c) + (v_1-1)\ln\theta + (v_2-1)\ln(1-\theta) \quad (2)$$

حال با مشتق گیری $\frac{\partial(\ln(L(\theta)))}{\partial(\theta)} = \frac{2(v_1-1)}{2\theta} + \frac{(1-v_2)}{1-\theta} + \frac{2\hat{\theta}}{1+2\hat{\theta}}(n-(n_b+n_d))\frac{2}{2\theta}$

$$\frac{v_2-1}{1-\theta} = \frac{2v_1-2}{2\theta} + \frac{(n-(n_b+n_d))}{\theta}\left(\frac{\hat{\theta}\times 2}{1+2\hat{\theta}}\right)$$

$$\frac{v_2-1}{1-\theta} = \frac{\alpha}{2\theta} \quad \Rightarrow \quad 2\theta v_2 - 2\theta = \alpha - \alpha\theta$$

$$\theta(2v_2 - 2 + \alpha) = \alpha$$

$$\theta = \frac{\alpha(2v_1-2)(n-(n_b+n_d))\left(\frac{\hat{\theta}\times 2}{1+2\hat{\theta}}\right)}{2v_2 + 2b(2v_1+2)(n-(n_b+n_d))\frac{2\hat{\theta}}{1+2\hat{\theta}}}$$

پاسخ ۵.

فرض کنید K بازیکن در روز t وجود دارد. یکی از آنها به تعداد m_t بار بازی می کند و تعداد w_t بازی را می برد. شما تنها تعداد کل این افراد، تعداد کل راند های بازی شده و تعداد بازی های برده شده توسط بازیکن را می دانید اما نمیدانید کدام یک از K بازیکن در کدام روز بازی کرده است. شما می خواهید از یادگیری ماشین برای حل این مسئله استفاده کنید. برای هر یک از K بازیکن شما یک مدل احتمالی می سازید که در آن فرد با احتمال p_k بازی را می برد. بنابرین در روز t اگر بازیکن i ام به تعداد m_t بار بازی کرده باشد ، احتمال آن که w_t بازی را ببرد توسط یک توزیع دوجمله ای بیان می شود. (I)

در این مسئله شما باید از یک مدل ترکیب شده با K متغیر تصادفی دوجمله ای با پارامترهای $p_1, p_2, \dots, p_k$ استفاده کنید و برای N روز داده شده به صورت $(m_1, w_1), \dots (m_n, w_n)$ بکار ببرید. به این صورت که در ابتدا در روز t ما ابتدا یک بازیکن از کل آنها با احتمال π انتخاب می کنیم (c_t). در مرحله بعد در روز t این بازیکن m_t بار بازی می کند و با دانستن اینکه بازیکن c_t است، تعداد برد w_t با یک متغیر دوجمله ای توصیف می شود. (II)

۵-۱. روابط توصیف شده (I) و (II) را بنویسید.

۵-۲. مرحله E را برای بروزرسانی Q با توجه به پارامتر های مرحله قبل بدست آورید.(نشان دهید در دور i ام $Q_t^{(i)}[k]$ چیست)

۵-۳. برای هر مدل ترکیبی مرحله M برای π در دور i ام از رابطه زیر محاسبه می شود.

$$\pi^i[k] = \frac{\sum_{t=1}^{n} Q_t^i[k]}{n}$$

مرحله M را برای بروزرسانی پارامتر های مدل $p^i_1, p^i_2, \dots, p^i_k$ در دور i ام برحسب داده و مقادیر $Q_t^{(i)}$ بدست آورید. در ابتدا مرحله maximization را برای پارامتر ها نشان داده و مسئله بهینه سازی را حل کنید.

(۵.۱

برای مدل کردن بخش I ، می‌توانیم از یک رویکرد احتمالاتی با استفاده از توزیع‌های دوجمله‌ای بهره ببریم. در این مسئله، ما K بازیکن داریم. باید تعداد بازی‌های روزانه و بردها را به گونه‌ای نمایش دهیم که با فرضیات مسئله سازگار باشد. فرض کنید در روز t یک بازیکن m_t بار بازی کرده و w_t بار برنده شده است. احتمال برنده شدن هر بازیکن با شرط اینکه هر بازیکن k که(k=1,2,3,……..,K) با احتمال p_t پیروز می‌شود، مشخص می گردد. با توجه به این مولفه‌ها، می‌توانیم احتمال مشاهده w_t برد از m_t بازی برای بازیکن iام در روز t را با استفاده از توزیع

دوجمله‌ای تعریف کنیم. توزیع دوجمله‌ای، تعداد موفقیت‌ها (در این مسئله، بردها) را در تعداد مشخصی از آزمایش‌ها (بازی‌های انجام شده) بر اساس احتمال موفقیت در هر آزمایش توصیف می‌کند.رابطه ی ان به شکل زیر است.

$$P(w_t|m_t, p_i) = \binom{m_t}{w_t} p_i^{w_t}(1-p_i)^{m_t-w_t}$$

برای تخمین احتمالات k_p هر بازیکن، بر اساس داده‌های مشاهده‌شده از بردها و باخت‌ها، می‌توانیم از یک فرآیند تکراری مانند روش EM (Expectation-Maximization) استفاده کنیم. در این رویکرد، ابتدا تخمین می‌زنیم که کدام بازیکن به احتمال زیاد در هر روز بازی کرده است. این کار با استفاده از داده‌های موجود و یک مدل اولیه انجام می‌شود (مرحله Expectation).

سپس، بر اساس این تخصیص اولیه، تخمین‌های احتمالات k_p را به‌روزرسانی می‌کنیم (مرحله Maximization). این فرآیند به صورت تکراری انجام می‌شود تا زمانی که به یک معیار خاص برای همگرایی برسیم یا تغییرات در تخمین‌ها کمتر از یک آستانه مشخص شوند. به این ترتیب، می‌توان احتمالات برنده شدن هر بازیکن را به طور دقیق‌تری بر اساس داده‌های موجود تخمین زد.

برای مدلسازی بخش II ، ما یک فرآیند دو مرحله‌ای را برای هر روز در نظر می‌گیریم. این فرآیند شامل دو مرحله است:

انتخاب بازیکن: در روز t یک بازیکن c_t از بین K بازیکن با احتمال $\pi(c_t)$ انتخاب می‌شود. این احتمال نشان می‌دهد که هر بازیکن چه شانسی برای بازی کردن در روز t دارد. به عبارت دیگر، ما برای هر بازیکن یک احتمال اختصاص می‌دهیم که نشان‌دهنده میزان احتمال بازی کردن آن‌ها در روز مشخص است. این احتمالات می‌توانند بر اساس عوامل مختلفی مانند عملکرد قبلی، سطح مهارت، و دیگر معیارهای مرتبط تعیین شوند.

نتایج بازی: وقتی یک بازیکن c_t انتخاب می‌شود، آنها m_t بازی را در روز t انجام می‌دهند. تعداد بازی‌های برده شده، w_t بسته به احتمال برنده شدن بازیکن انتخابی، به عنوان یک متغیر تصادفی دوجمله‌ای مدل می‌شود.

با این توضیحات، می‌توانیم روابط مربوطه را به این صورت توصیف کنیم:

- $c_t \in \{1, 2, ..., K\}$ نشان‌دهنده بازیکنی است که در روز t بازی می‌کند.

- احتمال انتخاب بازیکن$_t$ در روز t با $\pi(c_t)$ نشان داده می‌شود.

- بر اساس اینکه بازیکن c_t در روز t بازی می‌کند، تعداد بردهای w_t از توزیع دوجمله‌ای با پارامترهای m_t (تعداد بازی‌های انجام شده) و $\pi(c_t)$ (احتمال برنده شدن بازیکن(c_t)) پیروی می‌کند.

- احتمال مشاهده w_t پیروزی در روز t از رابطه‌ای خاص به دست می‌آید که بر اساس توزیع دوجمله‌ای و پارامترهای مذکور تعیین می‌شود.

$$P(w_t|m_t, c_t) = \binom{m_t}{w_t} p_{c_t}^{w_t}(1 - p_{c_t})^{m_t - w_t}$$

احتمال کلی داده‌های مشاهده شده برای تمام N روز به شرح زیر است

$$L = \prod_{t=1}^{N} \pi(c_t) \binom{m_t}{w_t} p_{c_t}^{w_t}(1 - p_{c_t})^{m_t - w_t}$$

برای مدل کلی که برای N روز در نظر گرفته شده و شامل داده‌های مشاهده $(m_1, w_1), \dots, (m_N, w_N)$ شده است، احتمال داده‌ها با توجه به پارامترهای $p_1, p_2, \dots, p_K$ مدل و π می‌تواند به صورت زیر بیان شود:

$$P\big((m_1, w_1), \dots, (m_N, w_N)\big|p_1, \dots, p_k, \pi\big) = \prod_{t=1}^{N} \sum_{c_t=1}^{K} \pi(c_t) \binom{m_t}{w_t} p_{c_t}^{w_t}(1 - p_{c_t})^{m_t - w_t}$$

در این مدل، ما احتمالات مربوط به همه بازی‌ها را در تمام روزها محاسبه می‌کنیم. این کار شامل مجموعه‌ای از احتمالات برای هر بازیکن است که می‌توانستند در هر روز بازی کنند. در اینجا، دو هدف اصلی مدل وجود دارد:

برآورد احتمال برنده شدن P_t برای هر بازیکن و احتمال انتخاب $(c_t\pi)$ برای هر بازیکن: این به معنی تخمین احتمال این است که یک بازیکن خاص در یک روز مشخص انتخاب شده باشد.دو بخش بعدی در عکس هست وقت نشد تایپ شه

5.2

سوال دهم:

وقتی ما الگوریتم EM را برای حل احتمال می‌زنیم در مرحله (E) ما باید به سراغ داده‌های مشاهده نشده برویم و آن‌ها را حساب می‌کنیم که شامل $\pi(C_k)$ و P_k برای هر بایاس k هستند و log likelihood

و در نهایت ما می‌خواهیم که به داده‌های مشاهده نشده، مجموع بازی‌ها و احتمال این که هر بایاس C_k در هر روز t بازی می‌کند را حساب کنیم.

$Q_t^{(i)}[k]$ را به عنوان احتمال این که بایاس k ما در تکرار i ام الگوریتم در روز t بازی کرده، نشان می‌دهیم.

مرحله M ← مرحله E

$$Q_t^{(i)}[k] = \text{Posterior} = \frac{\pi^{(i)}k \cdot \binom{m_t}{w_t}(P_k^{(i)})^{w_t}(1-P_k^{(i)})^{m_t-w_t}}{\sum_{j=1}^{k}\pi^{(i)}[j]\binom{m_t}{w_t}(P_j^{(i)})^{w_t}(1-P_j^{(i)})^{m_t-w_t}}$$

$\pi^{(i)}k$ تخمین فعلی احتمال انتخاب بایاس k هست.

$P_k^{(i)}$ برآورد فعلی احتمال سر زدن بایاس k در یک بازی هست

مخرج هم یک عامل نرمال‌ساز هست که همه‌ی بایاس‌ها جمع می‌شوند تا مطمئن باشیم این یک توزیع احتمال روی همه‌ی بایاس‌ها هست.

5.3

هدف ما در این بخش حداکثر کردن احتمال log likihood داده‌های مشاهده شده با توجه به پارامترهای فعلی هست.

$$\text{log likihood} = \ln\left(\prod_{i=1}^{N}\prod_{k=1}^{K} P_i^{w_t}\binom{m_t}{w_t}(1-P_t)^{m_t-w_t}_{C_i}\right)$$

$$\sum_{i=1}^{N}\sum_{k=1}^{K}\left(\ln\binom{m_t}{w_t} + w_t\ln(P_t) + (m_t-w_t)\ln(1-P_t) + \ln(C_i)\right)$$

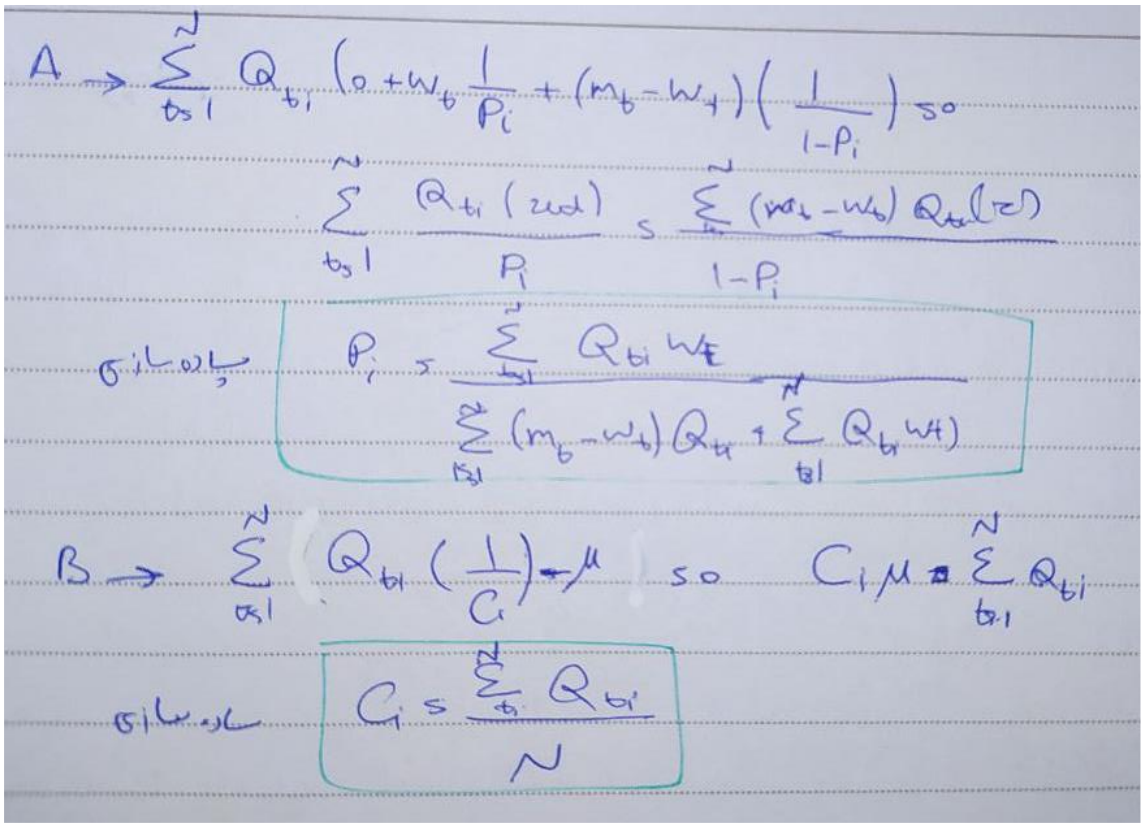
با توجه به راه‌حل سوال

$$\frac{\binom{m_t}{w_t}P_i^{w_t}(1-P_t)^{m_t-w_t}_{C_i}}{\sum_{k=1}^{K}\binom{m_t}{w_t}P_k^{w_t}(1-P_i)^{m_t-w_t}_{C_i}} \times \left\{\sum_{t=1}^{N}\sum_{i=1}^{K}\left(\ln\binom{m_t}{w_t} + w_t\ln(P_i) + (m_t-w_t)\ln(1-P_t) + \ln(C_i)\right) + \lambda(C_1+\dots+C_n)\right\}$$

برای λ

M-step

$$\underbrace{\frac{\partial L}{\partial R_i} = 0}_{A} \qquad \underbrace{\frac{\partial L}{\partial P_i} = 0}_{B}$$



پاسخ ٦.

در این سوال هدف پیاده سازی الگوریتم KNN و استفاده از آن به عنوان طبقه بند میباشد.

۶-۱. همانطور که میدانید الگوریتم KNN ساده و شهودی است، هنگام پیش‌بینی، فاصله بین هر یک از نقاط داده موجود را محاسبه می‌کند و آن را همانند نزدیک‌ترین کلاس به آن طبقه‌بندی می‌کند. یک کلاس KNN ساخته و با استفاده از کتابخانه numpy این الگوریتم را پیاده سازی کنید.

۶-۲. مجموعه داده[1] iris را لود کرده و اطلاعات کلی دیتاست شامل تعداد کلاس و تعداد سمپل ها و فرمت داده ها و ... بیان کنید.

۶-۳. Scatter plot مجموعه داده iris را رسم کنید.

۶-۴. مجموعه داده iris را به دو دسته آموزش و ارزیابی تقسیم کنید.

۶-۵. به کمک کلاس KNN که در بخش ۱ پیاده سازی کردید ، مدلی بر روی داده های آموزش به ازای k برابر با ۵ آموزش داده سپس دقت مدل را بر روی داده های آموزش و ارزیابی گزارش کنید.

۶-۶. بخش ۵ را به ازای k های متفاوت (۱ تا ۱۰) تکرار کنید و نمودار دقت بر روی داده های ارزیابی به ازای k های متفاوت را رسم کرده و بهترین k را بر اساس آن گزارش کنید.

با استفاده از کد زیر ما میتوانیم الگوریتم KNN را پیاده سازی کنیم

```python
import numpy as np

class KNN:
    def __init__(self, k=3):
        self.k = k

    def fit(self, X, y):
        self.X_train = X
        self.y_train = y

    def predict(self, X):
        y_pred = [self._predict(x) for x in X]
        return np.array(y_pred)

    def _predict(self, x):
        distances = [np.linalg.norm(x - x_train) for x_train in self.X_train]
        k_indices = np.argsort(distances)[:self.k]
        k_nearest_labels = [self.y_train[i] for i in k_indices]
        most_common = np.bincount(k_nearest_labels).argmax()
        return most_common
```

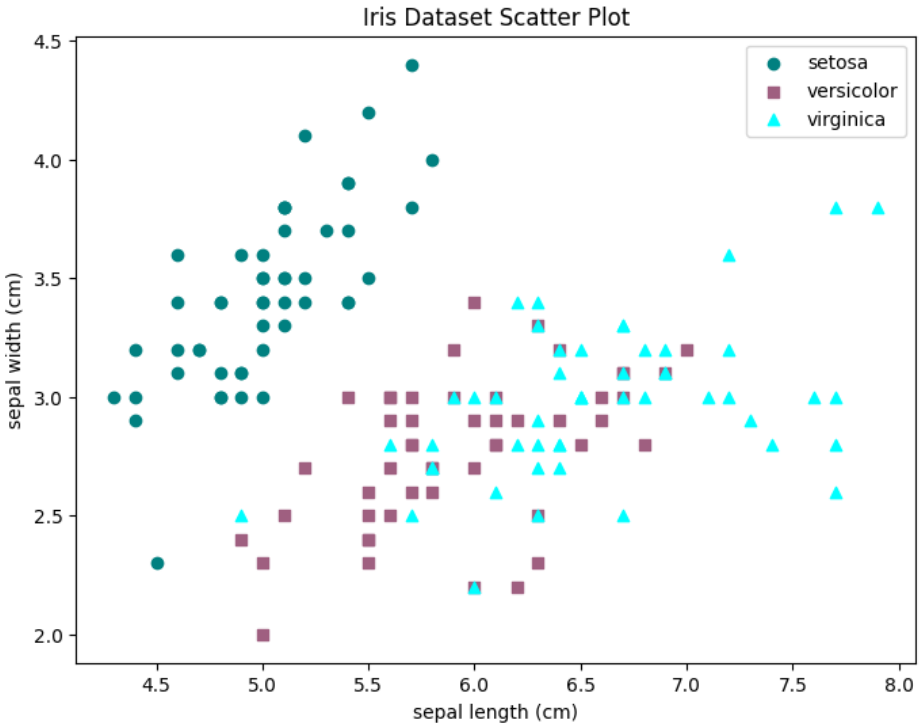
در قسمت بعدی کد ما خوصوصیات دیتا ست را بررسی میکنیم که به شرح زیر میباشد

```
Number of classes: 3
Class names: ['setosa' 'versicolor' 'virginica']
Number of samples: 150
Number of features: 4
Feature names: ['sepal length (cm)', 'sepal width (cm)', 'petal length (cm)', 'petal width (cm)']
```

من یک قسمت نرمالایزشن هم به کدم اضافه کردم

```python
from sklearn.preprocessing import MinMaxScaler
# Initialize the MinMaxScaler
scaler = MinMaxScaler()

# Fit and transform the data
X = scaler.fit_transform(data)
```

در مرحله بعدی ما scatter plot دیتا ست Iris را میکشیم



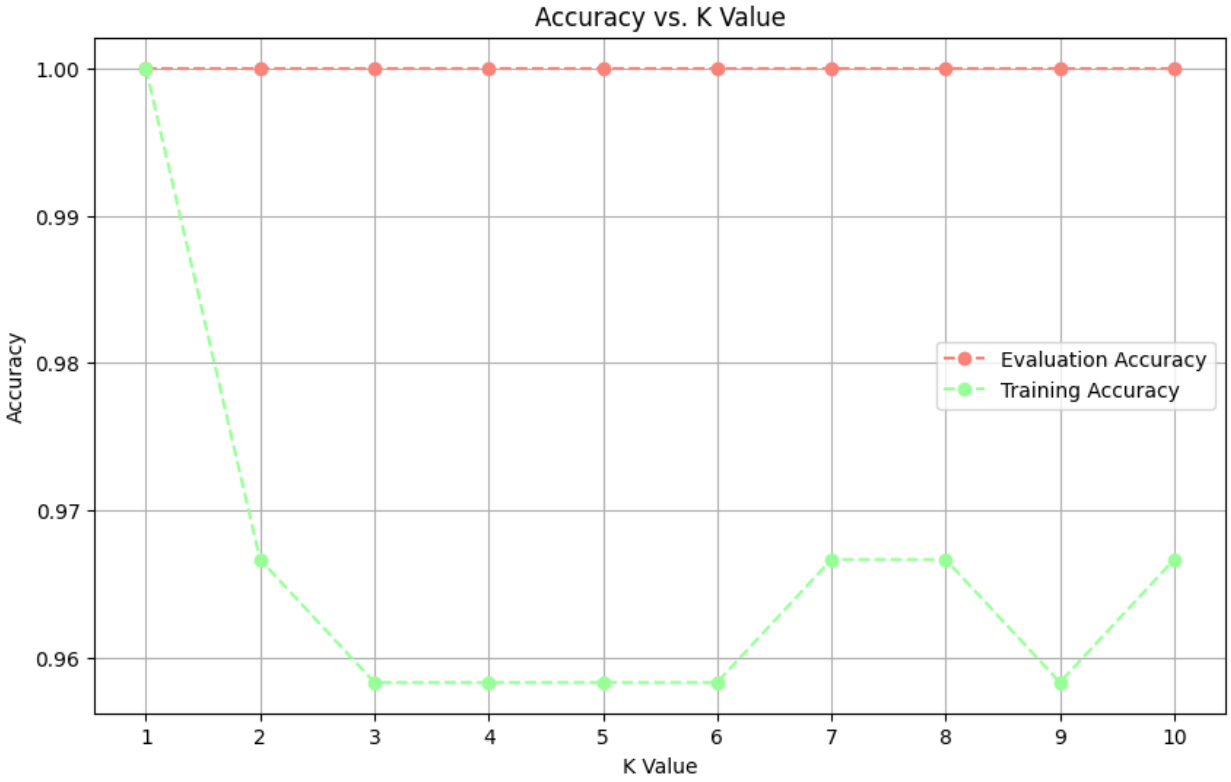
بعد از تقسیم داده به دو بخش اموزش و تست ما با استفاده از الگورتیم KNN که ٤قبلا خودمان تعریف کرده ایم مدل خود را به ازای $k = 5$ آموزش میدهی مکه دقت ان به شرح زیر است

```
Training accuracy: 95.83%
Evaluation accuracy: 100.00%
```

در مرحله بعدی به ازای $K = 1$ تا $k=10$ این کارا تکرار میکنیم که دقیت ان به شرح زیر است

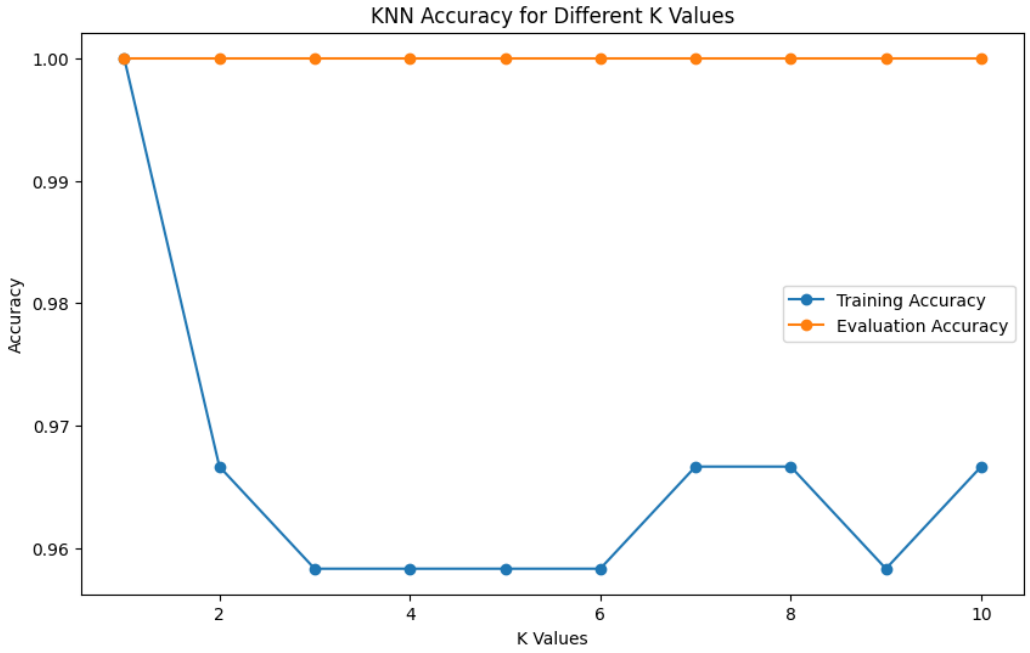
```
K = 1: Training Accuracy = 1.0000, Evaluation Accuracy = 1.0000
K = 2: Training Accuracy = 0.9667, Evaluation Accuracy = 1.0000
K = 3: Training Accuracy = 0.9583, Evaluation Accuracy = 1.0000
K = 4: Training Accuracy = 0.9583, Evaluation Accuracy = 1.0000
K = 5: Training Accuracy = 0.9583, Evaluation Accuracy = 1.0000
K = 6: Training Accuracy = 0.9583, Evaluation Accuracy = 1.0000
K = 7: Training Accuracy = 0.9667, Evaluation Accuracy = 1.0000
K = 8: Training Accuracy = 0.9667, Evaluation Accuracy = 1.0000
K = 9: Training Accuracy = 0.9583, Evaluation Accuracy = 1.0000
K = 10: Training Accuracy = 0.9667, Evaluation Accuracy = 1.0000
```

و در نهایت نمودار ان را میکشیم که به نظر میاید به ازای $K = 1$ بهترین نتیجه نشان داده میشود




```
The best k based on evaluation accuracy is: 1 with an accuracy of 1.0000
```

و در نهایت من برای این که متوجه درست کارکردن الگوریتم خود بشوم با استفاده از کتابخانه نیز این کد را پیداه کردم که نمودار مشابه ای به دست آمد که کد ان در فایل پایتون هست.



پاسخ ۷.

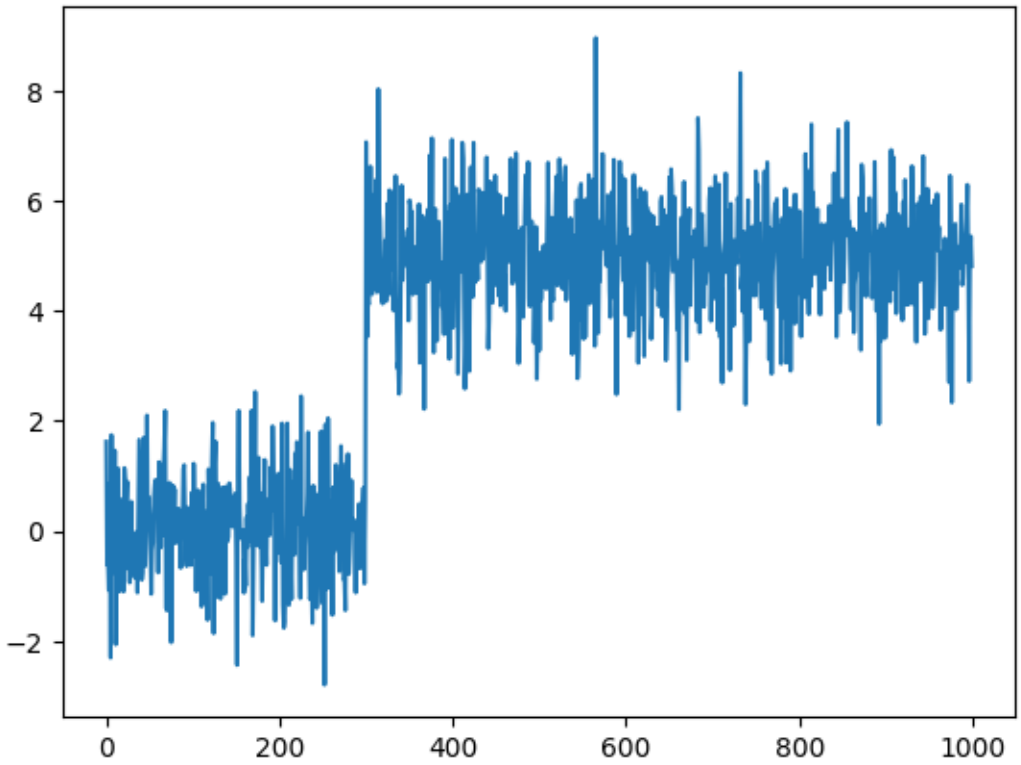
۷-۱. به کمک دستور زیر مجموعه داده X را تولید کنید.

```
import numpy as np
N = 1000
np.random.seed(1)
X = np.concatenate((np.random.normal(0, 1, int(0.3 * N)),
np.random.normal(5, 1, int(0.7 * N))))[:, np.newaxis]
```

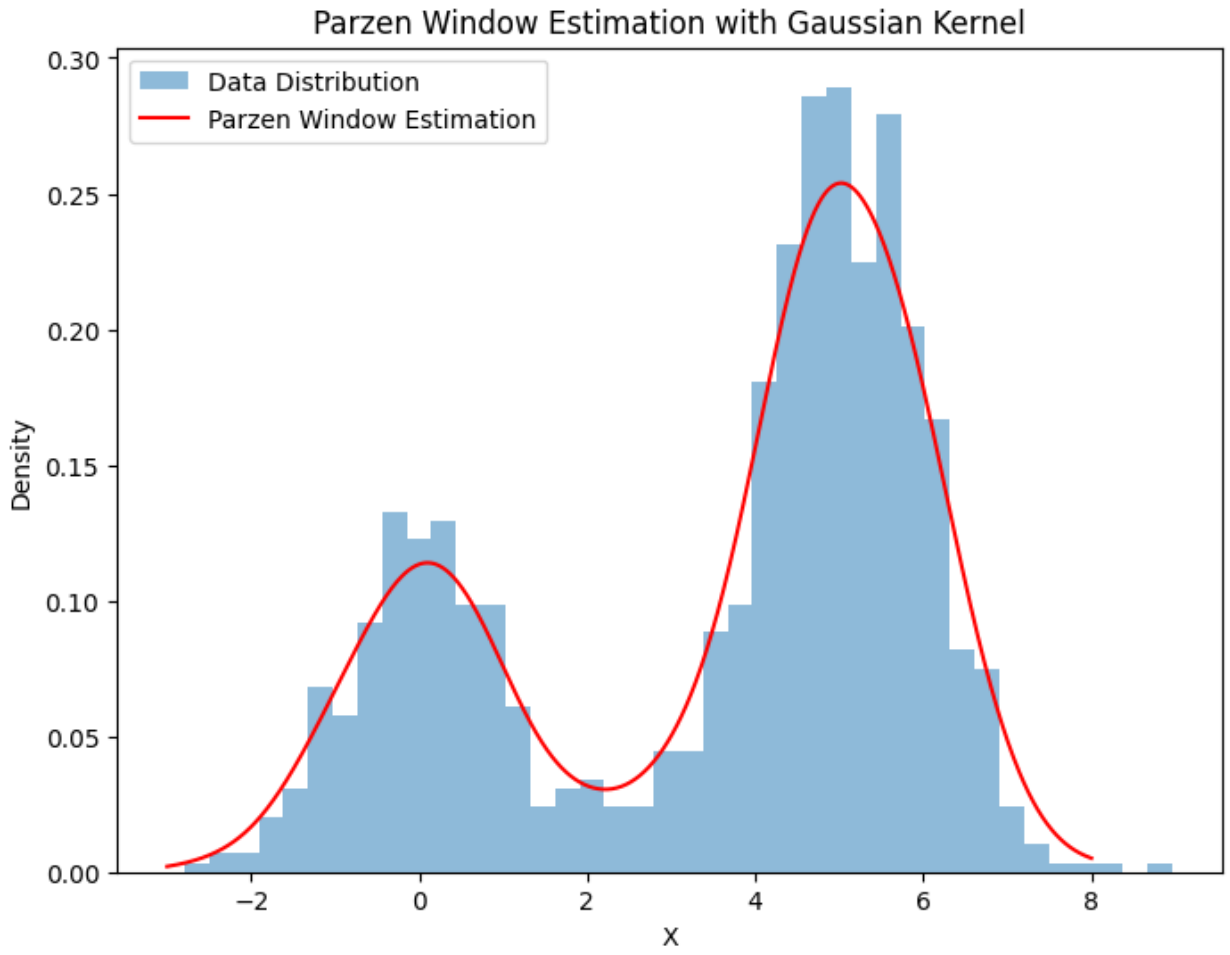
۷-۲. توزیع دیتا X را با استفاده از روش پنجره پارزن با کرنل گوسی بدست آورید.

۷-۳. تاثیر اندازه پنجره پارزن را روی توزیع تخمین زده شده بررسی کنید (حداقل ۳ اندازه مختلف مثلا : ۱۰ و ۱ و ۰.۱)

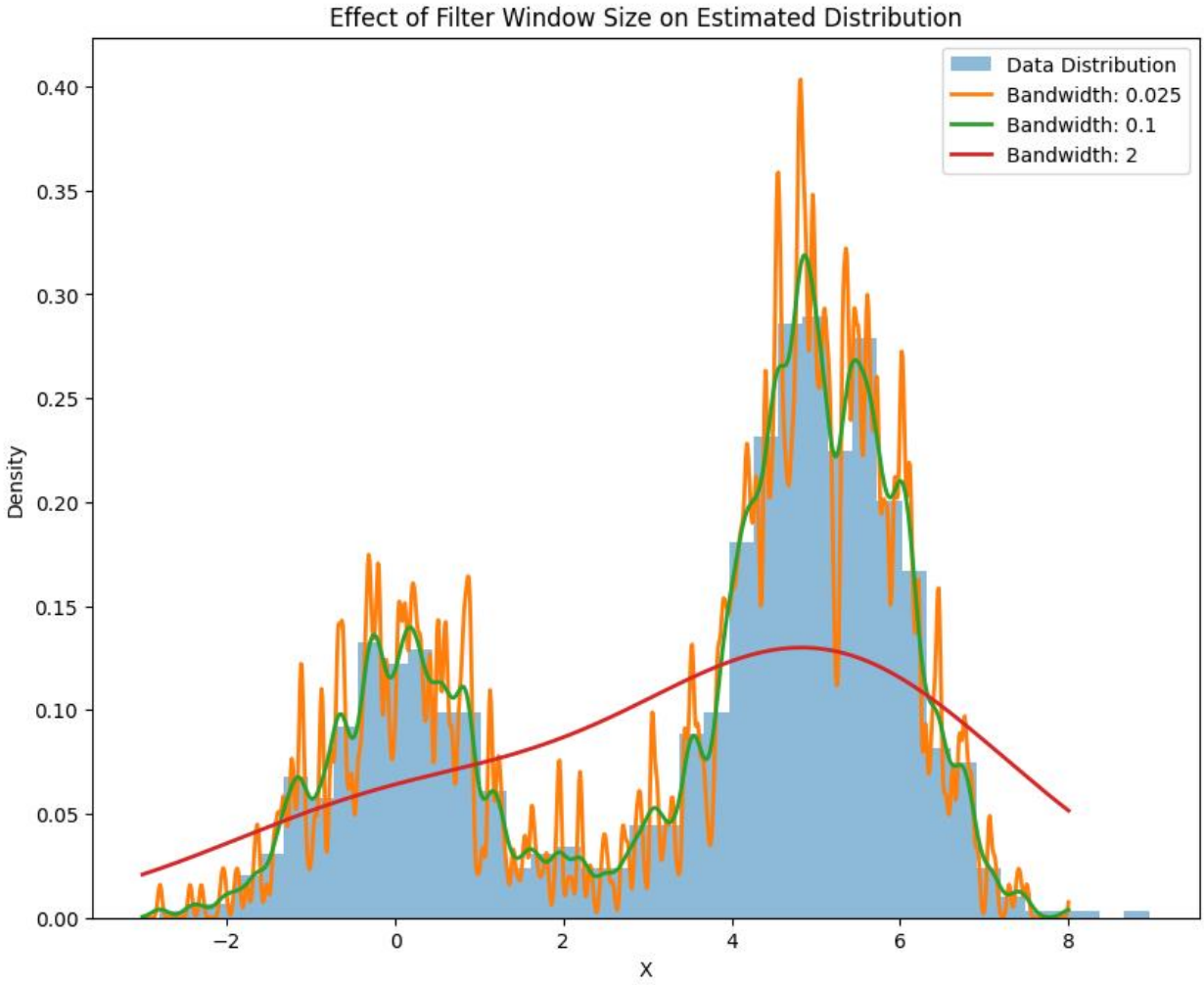
بعد از ساخت مجموعه داده اول ان را نمایش میدهیم



بعد از نمایش ان ما با استفاده از پنجره پارزن توضیع گوسی ان را نمایش میدهیم



و در نهایت با h های 2 , 0.1 , 0.025 ان را رسم میکنیم



در زمینه تجزیه و تحلیل داده‌ها و آمار، اندازه پنجره پارزن به پارامتر هموارسازی مربوط می‌شود، که به عنوان پهنای باند در تخمین چگالی هسته‌ای (KDE) شناخته می‌شود. KDE یک روش غیر پارامتریک برای تخمین تابع چگالی احتمال یک متغیر تصادفی است.

نمودار شامل یک توزیع داده و سه توزیع تخمینی با استفاده از پهنای باندهای مختلف است. پهنای باند کوچکتر منجر به یک خط حساس‌تر می‌شود که داده‌ها را به دقت دنبال می‌کند، که می‌تواند منجر به بیش برازش و واریانس بالا شود. در مقابل، پهنای باند بزرگتر تخمین را صاف می‌کند، که ممکن است منجر به کم برازش داده‌ها و نادیده گرفتن ساختار ظریف‌تر آن‌ها شود.

هدف از تنظیم اندازه پنجره فیلتر یافتن تعادلی است که ویژگی‌های مهم توزیع داده‌ها را بدون بیش برازش یا کم برازش بگیرد. از نمودار، به نظر می‌رسد که پهنای باند ۰.۱ شکل توزیع داده‌ها را نزدیک‌تر از بقیه دنبال می‌کند، در حالی که پهنای باند ۲ بیش از حد بزرگ است و باعث هموار شدن زیادی می‌شود و پهنای باند ۰.۰۲۵ بیش از حد کوچک است و نویز زیادی را ثبت می‌کند.

این جنبه‌ای حیاتی از تجزیه و تحلیل داده‌ها است، زیرا انتخاب پهنای باند بر نتایجی که می‌توان از داده‌ها استخراج کرد تأثیر می‌گذارد.

پاسخ ۸.

۸-۱ ابتدا دیتاست زیر را با استفاده از قطعه کد زیر ایجاد کنید.

```
from sklearn import cluster, datasets, mixture
noisy_moons=datasets.make_moons(n_samples=500, noise=0.11)
```

۸-۲. یک بار هر کلاس را با توزیع نرمال تقریب بزنید و پارامترهای آن را به دست آورده و کانتورهای مربوطه را رسم نمایید.

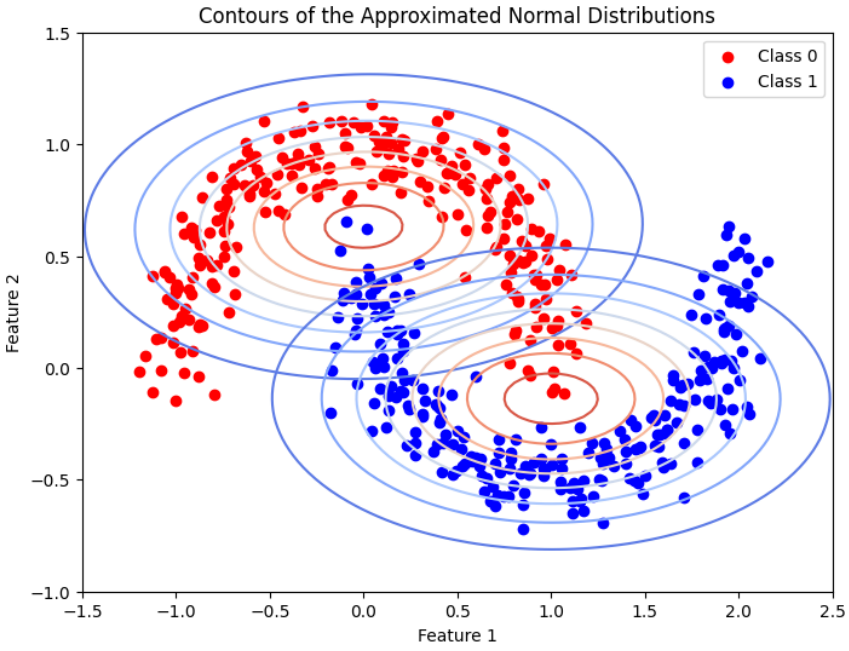
۸-۳. این بار از روش GMM استفاده کنید. روش GMM را با تعداد مولفه های ۱ تا ۱۶ تست کنید و شکل داده ها و کانتورها را برای تعداد مولفه برابر با ۳ و ۸ و ۱۶ بدست بیاورید.

۸-۴. تعداد مولفه های بهینه را با توجه به متریک های AIC و BIC به دست بیاورید.

بعد از ساختن دیتای مورد نظر با استفاده از کد زیر توزیع نرمال دیتاست زیر را به دست می آوریم

```
# Calculate mean and covariance for each class
mean_0 = np.mean(class_0, axis=0)
cov_0 = np.cov(class_0, rowvar=False)

mean_1 = np.mean(class_1, axis=0)
cov_1 = np.cov(class_1, rowvar=False)

# Function to calculate the Gaussian density
def gaussian_density(x, mean, cov):
    n = mean.shape[0]
    diff = (x - mean).reshape(-1, 1)
    return np.exp(-0.5 * np.dot(np.dot(diff.T, np.linalg.inv(cov)), diff)) / \
           ((2 * np.pi)**(n/2) * np.sqrt(np.linalg.det(cov)))

# Creating a grid of points to plot
x, y = np.linspace(-1.5, 2.5, 100), np.linspace(-1, 1.5, 100)
X, Y = np.meshgrid(x, y)
Z0 = np.zeros(X.shape)
Z1 = np.zeros(X.shape)

# Calculating the density for each point in the grid
for i in range(X.shape[0]):
    for j in range(X.shape[1]):
        Z0[i, j] = gaussian_density(np.array([X[i, j], Y[i, j]]), mean_0, cov_0)
        Z1[i, j] = gaussian_density(np.array([X[i, j], Y[i, j]]), mean_1, cov_1)
```

Contour plot هر کلاس که به شکل جدا رسم شده است به صورت زیر میباشد:



در محله ی بعدی از GMM استفاده میشود

مدل مخلوط گاوسی (GMM) یک روش آماری برای مدل‌سازی توزیع‌های احتمالاتی پیچیده است که می‌توانند به عنوان ترکیبی از چندین توزیع نرمال ساده‌تر در نظر گرفته شوند. در اینجا کلیدی‌ترین مفاهیم و کاربردهای GMM را توضیح می‌دهیم:

۱ . ترکیب چندگانه: GMM فرض می‌کند که داده‌ها از ترکیبی از چندین توزیع نرمال (گاوسی) تولید شده‌اند. هر یک از این توزیع‌های نرمال را می‌توان به عنوان یک "جزء" یا "گروه" در نظر گرفت.

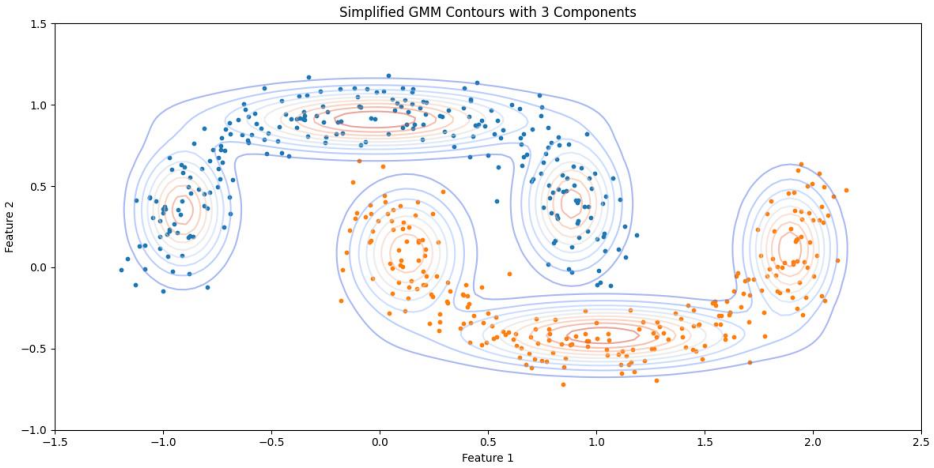
۲ . پارامترها: هر جزء دارای سه پارامتر اصلی است: میانگین (که مرکز توزیع را مشخص می‌کند)، کوواریانس (که شکل و گستردگی توزیع را تعیین می‌کند)، و وزن (که نشان‌دهنده احتمال وجود داده‌ها در آن جزء است).

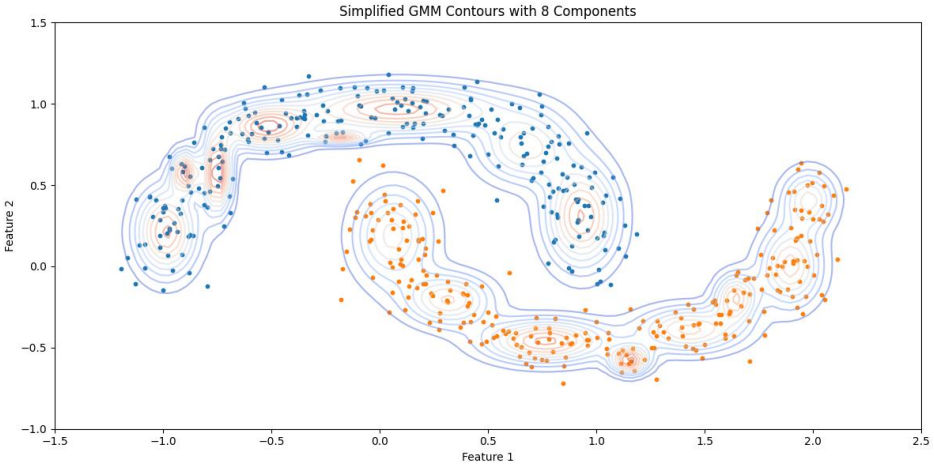
۳ . یادگیری پارامترها: برای یادگیری پارامترهای جزءها، معمولاً از الگوریتم EM (Expectation-Maximization) استفاده می‌شود. این الگوریتم به صورت تکراری احتمال تعلق هر نقطه داده به هر جزء را برآورد می‌کند و سپس پارامترهای جزءها را براساس این احتمالات به‌روزرسانی می‌کند .

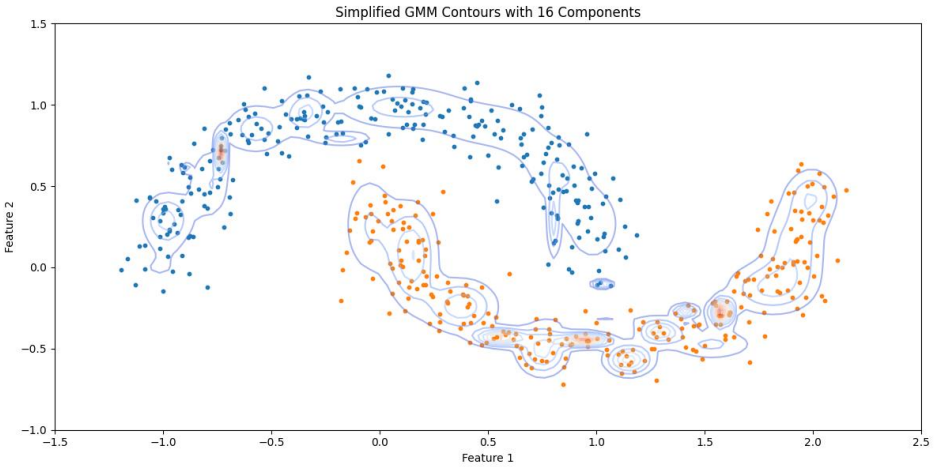
٤ . کاربردها: GMM در بسیاری از زمینه‌هایی که نیاز به مدل‌سازی توزیع‌های پیچیده داده‌ها وجود دارد، کاربرد دارد. این شامل شناسایی الگو، تجزیه و تحلیل خوشه‌ای، کاهش بُعد و سایر کاربردهای مرتبط با یادگیری ماشین است.

۵. مزایا و معایب: مزیت اصلی GMM در انعطاف‌پذیری آن در مدل‌سازی توزیع‌های پیچیده است. با این حال، انتخاب تعداد صحیح جزءها و تعیین پارامترهای اولیه می‌تواند چالش‌برانگیز باشد. همچنین، GMM ممکن است در مواجهه با داده‌های دارای ابعاد بالا یا داده‌هایی که به خوبی از هم جدا نشده‌اند، با مشکلاتی روبرو شود.

توجه:باتوجه به این که در قسمت قبل برای هر کلاس باید بکشیم و این قسمت هم گفته شده که حال این کار را GMM انجام بدید پس ما برای هر کلاس انجام میدیم GMM را.







و در نهایت با استفاده از متود های BIC و AIC تعداد بهینه را مشخص میکنیم

AIC (Akaike Information Criterion)

AIC یک معیار برای ارزیابی کیفیت مدل‌های آماری است که هم کیفیت برازش داده‌ها و هم تعداد پارامترهای مدل را در نظر می‌گیرد.AIC بر اساس فرمول زیر محاسبه می‌شود:

$$\mathrm{AIC} = 2k - 2\ln(L)$$

که در آن k تعداد پارامترهای مدل است و L احتمال بیشینه (maximum likelihood) است.AIC سعی دارد تعادلی بین پیچیدگی مدل (تعداد پارامترها) و کیفیت برازش مدل به داده‌ها ایجاد کند.

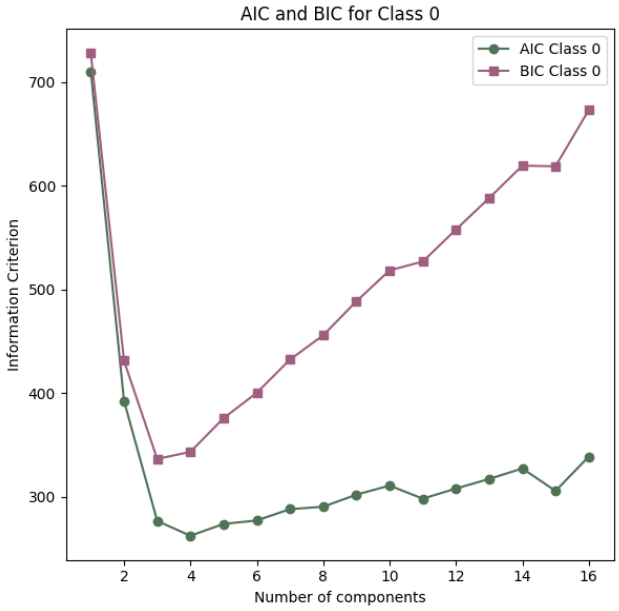
BIC (Bayesian Information Criterion)

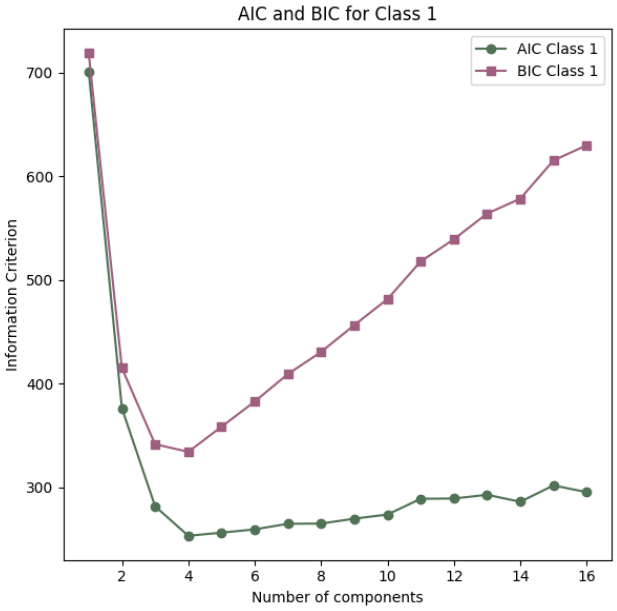
BIC، که گاهی به عنوان معیار اطلاعات شوارتز (Schwarz Criterion) نیز شناخته می‌شود، یک شاخص برای انتخاب مدل است که هم برازش داده‌ها و هم اندازه نمونه و تعداد پارامترهای مدل را در نظر می‌گیرد.BIC بر اساس فرمول زیر محاسبه می‌شود:

$$\mathrm{BIC} = \ln(n)k - 2\ln(L)$$

که در آن n اندازه نمونه، k تعداد پارامترهای مدل و L احتمال بیشینه است.

مانند AIC، BIC نیز به دنبال یافتن تعادل بین پیچیدگی مدل و برازش داده‌ها است. با این حال، BIC به اندازه نمونه نیز توجه دارد و بنابراین در موقعیت‌هایی با نمونه‌های بزرگ، به شدت به پیچیدگی مدل حساس‌تر است.





تعداد بهینه کلاس ها هم به شرح زیر میباشد

```
Optimal number of components based on AIC for Class zero: 4
Optimal number of components based on BIC for Class zero: 3
Optimal number of components based on AIC for Class one: 4
Optimal number of components based on BIC for Class one: 4
```

پی نوشت : کد هر سه سوال اخر ضمیمه شده است.