بنام خدا

علی قاسم زاده ۴۰۱۱۰۶۲۳۹

تمرین دوم ml

۱- الف)

$$\mu = \frac{1}{4}\left(\begin{bmatrix}4\\1\end{bmatrix}+\begin{bmatrix}1\\0\end{bmatrix}+\begin{bmatrix}2\\5\end{bmatrix}+\begin{bmatrix}3\\4\end{bmatrix}\right) = \begin{bmatrix}2.5\\2.5\end{bmatrix}$$

$$\Sigma = \frac{1}{4}\left(\left(\begin{bmatrix}4\\1\end{bmatrix}-\begin{bmatrix}2.5\\2.5\end{bmatrix}\right)\left(\begin{bmatrix}4\\1\end{bmatrix}-\begin{bmatrix}2.5\\2.5\end{bmatrix}\right)^T + \cdots + \left(\left(\begin{bmatrix}3\\4\end{bmatrix}-\begin{bmatrix}2.5\\2.5\end{bmatrix}\right)\left(\begin{bmatrix}3\\4\end{bmatrix}-\begin{bmatrix}2.5\\2.5\end{bmatrix}\right)^T\right)$$

$$= \frac{1}{4}\begin{bmatrix}5 & 1\\1 & 17\end{bmatrix} \longrightarrow \text{مقادیر ویژه}: \left(\frac{5}{4}-\lambda\right)\left(\frac{17}{4}-\lambda\right)-1=0 \Rightarrow \lambda^2-\frac{11}{2}\lambda+\frac{21}{4}=0$$

$$\longrightarrow \lambda_1 = \frac{11+\sqrt{37}}{4},\ \lambda_2 = \frac{11-\sqrt{37}}{4}$$

$$v_1 = \begin{bmatrix}-6+\sqrt{37}\\1\end{bmatrix},\ v_2 = \begin{bmatrix}-6-\sqrt{37}\\1\end{bmatrix}$$

حالا دیتا ها را در سطرهای یک ماتریس می چینیم به اسم X، خواهیم داشت:

$$X = \begin{bmatrix}4 & 1\\1 & 0\\2 & 5\\3 & 4\end{bmatrix} \longrightarrow Xv_1 \approx \begin{bmatrix}1.33\\0.08\\5.17\\4.25\end{bmatrix},\ Xv_2 = \begin{bmatrix}-47.33\\-12.08\\-19.17\\-32.25\end{bmatrix}$$

در راستای v_1، داده ها کلاس فاصله زیادی دارند یعنی واریانس بالا ولی در راستای v_2 فاصله کمتر دارند ← در راستای v_2 بهتر داده ها قابل جداسازی اند. همچین در v_2 نمی توانیم در مقدار یک بعدی تفکیک کنیم ولی در v_1 می توانیم یک threshold بزاریم بین 4.25 , 1.33 که دو تا دسته جداسازی شوند

ب)

بردار هر بردار v که به تمامی داده ها اضافه شود، داریم که:

$$\mu' = \sum(x_i + v) = \mu + v$$

$$\rightsquigarrow \Sigma' = \frac{1}{n}\sum_i (x_i' - \mu')(x_i' - \mu')^T = \frac{1}{n}\sum_i (x_i + v - \mu - v)(x_i + v - \mu - v)^T = \frac{1}{n}\sum_i (x_i-\mu)(x_i-\mu)^T$$

$$\longrightarrow \Sigma' = \Sigma \longrightarrow \text{PC dont change}$$

Scanned with CamScanner

۲-

$$f_{ensemble}(x) = \frac{1}{M}\sum_{i=1}^{M} f_i(x)$$

برای سادگی با f_e نشان می‌دهیم

الف)

$$Var(f_e) = Var\left(\frac{1}{M}\sum_{i=1}^{M} f_i(x)\right) = \frac{1}{M^2} Var\left(\sum_{i=1}^{M} f_i(x)\right)$$

سوال
طبق فرض f_iها مستقل بودند، پس داریم که،

$$\frac{1}{M^2}\sum_{i=1}^{M} Var(f_i(x))$$

طبق فرض سوال $Var(f_i)$ها برابرند

پس داریم که،

$$\frac{1}{M^2} \times M \; Var(f_1(x)) = \frac{Var(f_1(x))}{M}$$

$$\Rightarrow \boxed{Var(f_e) = \frac{Var(f_1(x))}{M}}$$

حالا اگر M را زیاد کنیم، واریانس f_e کم خواهد شد.

حالا bias را بررسی می‌کنیم،

تابع تخمینی — تابع واقعی

$$Bias = E[\hat{f}(x)] - f(x)$$

$f(x)$ که ثابت است پس $E[f_e(x)]$ را حساب می‌کنیم،

چون f_iها با بایاس یکسانی دادند طبق فرض سوال:

$$E[f_e(x)] = E\left[\frac{1}{M}\sum_{i=1}^{M} f_i(x)\right] = \frac{1}{M}\sum_{i=1}^{M} E(f_i(x)) = E(f_1(x))$$

$$\longrightarrow Bias = E(f_1(x)) - f(x) \longrightarrow$$

بایاس با تخمین M تخمینی نمی‌کند و ثابت می‌ماند.

ب) ابتدا واریانس را در این حالت حساب می‌کنیم، داریم که،

$$Var\left(\frac{1}{M}\sum_{i=1}^{M} f_i(x)\right) = \frac{1}{M^2} Var\left(\sum_{i=1}^{M} f_i(x)\right) = \frac{1}{M^2}\left(\sum_{i=1}^{M} Var(f_i(x)) + 2\sum_{1\le i<j\le M} Cov(f_i(x), f_j(x))\right.$$

Scanned with CamScanner

طبق فرض سوال داریم که $cov(f_i(x), f_j(x)) = \rho$

$$\Rightarrow Var(f_e) = \frac{1}{M^2}\left(M\, Var(f_i(x)) + 2\left(\frac{M^2-M}{2}\right)\rho\right) = \frac{Var(f_i(x))}{M} + \frac{(M^2-M)}{M^2}\rho$$

$$\Rightarrow \boxed{Var(f_e) = \frac{Var(f_i(x))}{M} + \frac{M-1}{M}\rho} = \rho + \frac{1}{M}\left(Var(f_i(x)) - \rho\right)$$

$$\frac{M-1}{M}\rho = \left(1-\frac{1}{M}\right)\rho$$

← با افزایش M به سمت ∞ ، مقدار واریانس برابر با ρ می‌شود

حالا اگر در صورت ثابت بودن M ، با افزایش ρ ، واریانس زیاد می‌شود

اگر $Var(f_i(x)) - \rho > 0$ باشد، با افزایش M واریانس کم می‌شود.

اگر $Var(f_i(x)) - \rho = 0$ باشد، با افزایش M واریانس تغییر نمی‌کند.

اگر $Var(f_i(x)) - \rho < 0$ باشد، با افزایش M واریانس ۰ زیاد می‌شود

در مورد بایاس، چون E خطی است و داخلش جمع مرتبط ادل داریم، تغییر ایجاد نمی‌شود نسبت به حالت مستقل ← بایاس باز هم تغییر نمی‌کند.

ب) یادگیرنده های ضعیف ada boost نیاز به مشتق پذیر بودن ندارند همانطور که در الگوریتم موجود در اسلایدها داریم، آپدیت وزن های یادگیرنده ضعیف بدون مشتق گیری انجام می‌شود

boosting هزینه بیشتری دارد، زیرا boosting به صورت سری کار می‌کند و موازی انجام نمی‌شوند.

همچنین مدل های پیچیده تر نیز دارند چون از تمام دیتا برای یادگیری استفاده می‌کند برخلاف bagging که روی بخشی از دیتا کار می‌کند.

Scanned with CamScanner

۳-

الف) اگر k=1 باشد، هر نقطه هر نزدیک‌ترین به خودش است، پس خطای training صفر می‌شود.

ب) اگر k زیاد باشد، عملاً داریم می‌بینیم عموم برچسب‌ها چی هستند براساس این لیبل می‌زنیم، به‌عبارتی bias خیلی زیادی داریم. اگر هر k خیلی کم باشد، ممکن است دینار کمی با استثناء برچسب داشته باشند نزدیک آن نقطه باشد به آن نقطه هر استثناء برچسب زده می‌شود، در صورتی که اگر همسایه‌ها بیشتر می‌دیدیم ممکن بود درست برچسب بزنیم.

پ) با استفاده از کد پایتون برای ۱۲ — k=1 ، بهترین k، ۵ و ۷ بودند با خطای ۰/۲۸۵۷

ت) در شکل صفحه‌ی بعد آورده شده است.

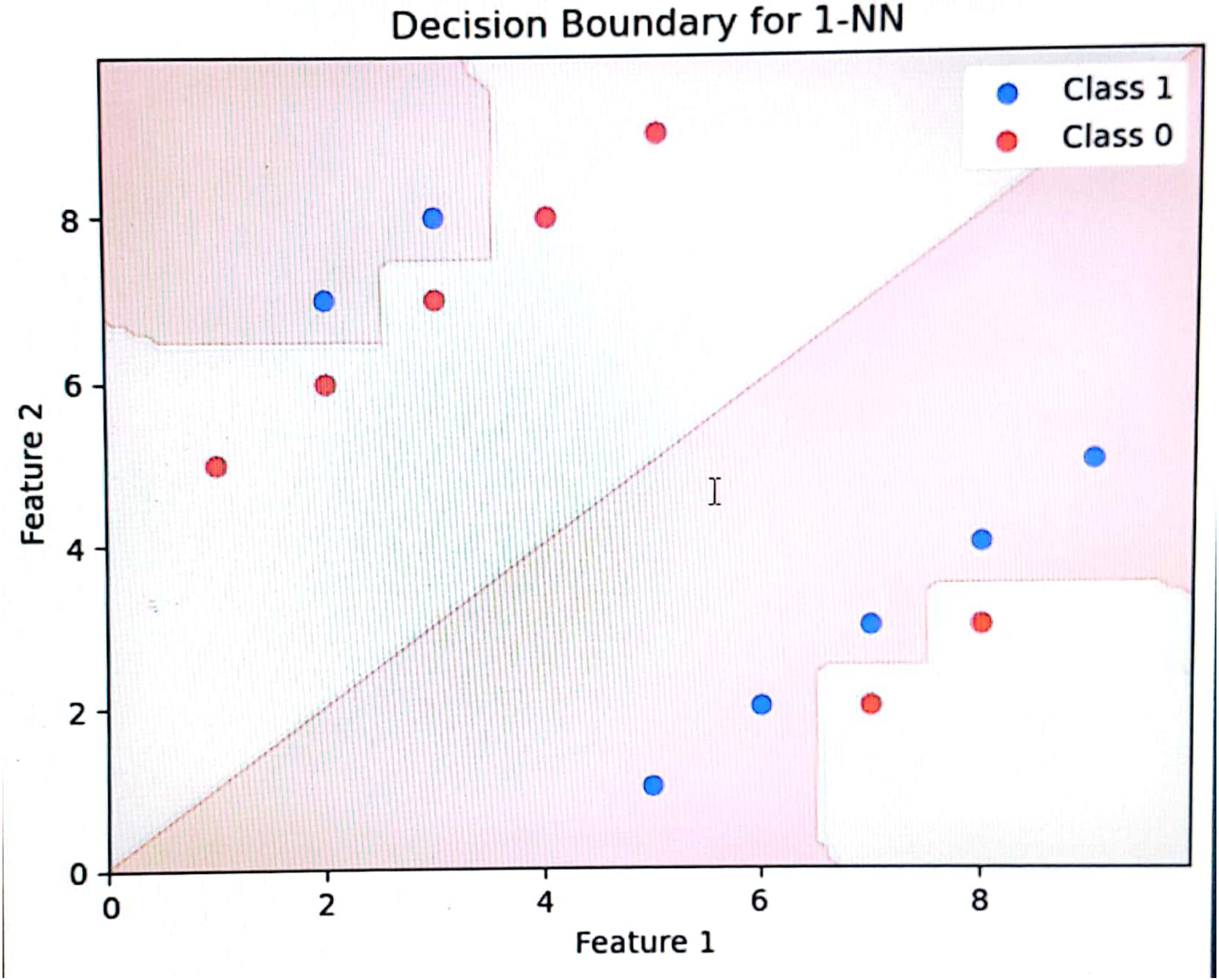
Scanned with CamScanner

```python
import numpy as np
from sklearn.model_selection import LeaveOneOut, cross_val_score
from sklearn.neighbors import KNeighborsClassifier

pos = np.array([[2, 7], [3, 8], [5, 1], [6, 2], [7, 3], [8, 4], [9, 5]])
neg = np.array([[1, 5], [2, 6], [3, 7], [4, 8], [5, 9], [7, 2], [8, 3]])
X = np.vstack((pos, neg))
y = np.array([1] * 7 + [0] * 7)
LeaveOne = LeaveOneOut()
errors = {}

for k in range(1, 14):
    knn = KNeighborsClassifier(n_neighbors=k)
    scores = cross_val_score(knn, X, y, cv=LeaveOne, scoring="accuracy")
    errors[k] = 1 - np.mean(scores)

errors
```

✓ 0.2s

```
{1: 0.7142857142857143,
 2: 0.7142857142857143,
 3: 0.42857142857142856,
 4: 0.6428571428571428,
 5: 0.2857142857142857,
 6: 0.3571428571428571,
 7: 0.2857142857142857,
 8: 0.6428571428571428,
 9: 1.0,
 10: 0.7142857142857143,
 11: 1.0,
 12: 0.7142857142857143,
 13: 1.0}
```

Scanned with CamScanner



Scanned with CamScanner

الف) برای پیدا کردن مینیمم، مشتق می‌گیریم و برابر صفر می‌ذاریم:

$$\frac{\partial L}{\partial \mu_t} = \frac{\partial}{\partial \mu_t} \sum_{j=1}^{k} \sum_{x_i \in S_i} \|x_i - \mu_j\|^2$$

وقتی که $j = t$ است مشتق ناصفر است ←

$$\frac{\partial L}{\partial \mu_t} = \frac{\partial}{\partial \mu_t} \sum_{x_i \in S_i} \|x_i - \mu_t\|^2 \Longrightarrow = -2 \sum_{x_i \in S_i} (x_i - \mu_t) = 0$$

$$\rightsquigarrow M\mu_t - \sum_{x_i \in S_i} x_i = 0 \longrightarrow \boxed{\mu_t^* = \frac{\sum_{x_i \in S_i} x_i}{M}}$$

همان مقدار میانگین نقاط هر خوشه می‌شود μ_t بهینه برای آن خوشه.

ب) • الگوریتم همواره همگرا می‌شود زیرا در هر مرحله داریم خطا را کمتر می‌کنیم (طبق اسلایدها)
و تعداد حالات انتخاب مرکز ثقل هر خوشه محدود است ← الگوریتم همگرا می‌شود

علت کاهش خطا هر این است که هر مرحله داریم در جهت عکس گرادیان μ حرکت می‌کنیم.

الگوریتم به مقدار دهی اولیه خوشه‌ها حساس است و برای بهبود انتخاب اولیه مرکز ثقل، الگوریتم‌هایی ارائه شده است. (ممکن است چند تا مرکز ثقل داشته باشیم، GD در مینیمم محلی بیفتد.)

پ) اگر x_j داخل خوشه‌ی قبلی نماند، ممکن است هی بین خوشه‌ها مختلف جابه‌جا شود و همگرایی الگوریتم را دچار اختلال کند در این صورت این روند که هر سری خطا کم شود، برقرار نمی‌شود.

Scanned with CamScanner

برای پیدا کردن argmin عبارت $\|x - \vec{v}^{\alpha u}\|^2$ از این عبارت نسبت به α مشتق می‌گیریم و برابر صفر قرار می‌دهیم،

$$\frac{\partial}{\partial \alpha}(x-\alpha u)^T(x-\alpha u) = -u^T(x-\alpha u) = 0 \rightarrow u^Tx - \alpha u^Tu = 0 \rightarrow \boxed{\alpha = u^Tx}$$

$$\Longrightarrow \boxed{f_u(x) = \overbrace{(u^Tx)}^{x^Tu} u}$$ بود. $\alpha = u^Tx$ ، $v = \alpha u$ برداری بود که $f_u(x)$

$$\rightsquigarrow \underset{\substack{u \\ u^Tu=1}}{\text{argmin}} \frac{1}{m}\sum \|x^{(i)} - f_u(x^{(i)})\|^2 = \frac{1}{m}\sum \|x^{(i)} - u^Tx^{(i)}u\|^2$$

$$= \frac{1}{m}\sum \left(x^{(i)T}x^{(i)} + (u^Tx^{(i)})u^Tu(u^Tx^{(i)}) - 2(u^Tx^{(i)})^2\right)$$

$$= \frac{1}{m}\sum \left(x^{(i)T}x^{(i)} - (u^Tx^{(i)})^2\right) \rightsquigarrow \underset{\substack{u \\ u^Tu=1}}{\text{argmin}} \quad \equiv \text{عبارت}$$

$$* \underset{\substack{u \\ u^Tu=1}}{\text{argmax}} \frac{1}{m}\sum (u^Tx^i)^2 \rightarrow \underset{u}{\text{argmax}} \|Xu\|^2 \rightarrow$$

uای که یک شده/ بردار دیتا/ متناظر با بزرگترین مقدار ویژه است، در این صدق می‌کند.

طبق اسلایدها داریم که،

$$Var(Xv) = \frac{1}{n}\sum_{i=1}^{n}(x_i^Tv)^2$$

$$\rightarrow Var(Xv) = \frac{1}{n}\|Xv\|^2 = \frac{1}{n}v^TX^TXv = \frac{1}{n}v^T\Sigma v$$

عبارت * هم معادل همین رابطه است، در PCA هر v بردار دیتا متناظر با بزرگترین مقدار ویژه است.

uای که MSE بین نقاط و تصویر آنها را کمینه می‌کند، همان مولفه اول PCA است.

Scanned with CamScanner