رضا قربانی ۴۰۳۲۰۹۵۴۵ «تمرین سری دوم یادگیری ماشین - بخش تئوری»

همفکران: حسین شاه آبادی - حسین بغدادی اصل - سینا صیفوریه - فاطمه شاه آبادی

سوال ۱) الف

بردارهای میانگین:

$$\omega_1 = \left\{ \begin{pmatrix} 4 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \end{pmatrix} \right\} \longrightarrow \mu_{\omega_1} = \frac{1}{2} \begin{pmatrix} 4+1 \\ 1+0 \end{pmatrix} = \begin{pmatrix} 2.5 \\ 0.5 \end{pmatrix}$$

$$\omega_2 = \left\{ \begin{pmatrix} 2 \\ 5 \end{pmatrix}, \begin{pmatrix} 3 \\ 4 \end{pmatrix} \right\} \longrightarrow \mu_{\omega_2} = \frac{1}{2} \begin{pmatrix} 2+3 \\ 5+4 \end{pmatrix} = \begin{pmatrix} 2.5 \\ 4.5 \end{pmatrix}$$

ماتریس کوواریانس:

$$\Sigma = \frac{1}{N-1} \sum (x_n - \bar{x})(x_n - \bar{x})^T$$

$$\Sigma_{\omega_1} = \frac{1}{2-1} \left(\begin{pmatrix} 1.5 \\ 0.5 \end{pmatrix} (1.5 \;\; 0.5) + \begin{pmatrix} -1.5 \\ -0.5 \end{pmatrix} (-1.5 \;\; -0.5) \right) = \begin{bmatrix} 2.25 & 0.75 \\ 0.75 & 0.25 \end{bmatrix} + \begin{bmatrix} 2.25 & 0.75 \\ 0.75 & 0.25 \end{bmatrix}$$

$$\Sigma_{\omega_1} = \frac{1}{2} \begin{bmatrix} 4.5 & 1.5 \\ 1.5 & 0.5 \end{bmatrix}$$

$$\Rightarrow cov_T = \frac{2}{4} \Sigma_{\omega_1} + \frac{2}{4} \Sigma_{\omega_2} = \begin{bmatrix} 2.5 & 0.5 \\ 0.5 & 0.5 \end{bmatrix}$$

$$\Sigma_{\omega_2} = \begin{bmatrix} 0.5 & -0.5 \\ -0.5 & 0.5 \end{bmatrix}$$

eigen vector & eigen value

$$\det(cov_T - \lambda I) = \det \begin{bmatrix} 2.5-\lambda & 0.5 \\ 0.5 & 0.5-\lambda \end{bmatrix} = (2.5-\lambda)(0.5-\lambda) - 0.25 = 0$$

$$\longrightarrow \lambda = \frac{3 \pm \sqrt{5}}{2}$$

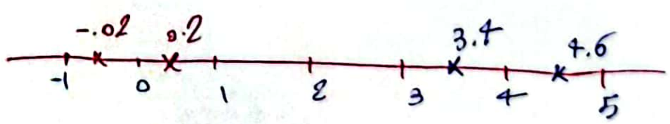
$$P_1 = V_1 = \begin{bmatrix} 2-\sqrt{5} \\ 1 \end{bmatrix} = -0.2$$

$$P_2 = V_2 = \begin{bmatrix} 2+\sqrt{5} \\ 1 \end{bmatrix} = 4.2$$

تصویر کردن دیتا

$$[4 \;\; 1] \begin{bmatrix} -0.2 \\ 1 \end{bmatrix} = -0.8 + 1 = 0.2 \qquad [2 \;\; 5] \begin{bmatrix} -0.2 \\ 1 \end{bmatrix} = 4.6$$

$$[1 \;\; 0] \begin{bmatrix} -0.2 \\ 1 \end{bmatrix} = -0.2 + 0 = -0.2 \qquad [3 \;\; 4] \begin{bmatrix} -0.2 \\ 1 \end{bmatrix} = 3.4$$



ب) اضافه کردن هر وکتور تابعی اطلاعات مربوط راستاهای v_1 و v_2 را نگه میدارد. تا حدی نیز پذیری و پارامترهای دیگر تاثیر نمیشه

سوال 2)

الف) چون مدل ensemble از فرمول $f_x(x)=\frac{1}{m}\sum_{i=1}^{m} f_i(x)$ بدست میاد اگر بایاس همه باهم برابر باشه و b باشه پس در نهایت ~~...~~ بایاس مدل نهایی نیز تغییر ~~...~~ نخواهد کرد

$$\boxed{Bias(f_e(x)) = b}$$

اما واریانس به شکل زیر تغییر می کند

$$Var(f_{en}(x)) = \frac{1}{m^2}\sum \underbrace{var(f_i(x))}_{\substack{\sigma^2 \\ \text{هر کدوم مشون}}} = \frac{1}{m^2}\times m\sigma^2 = \frac{\sigma^2}{m}$$

ب) این همبستگی تاثیری رو مقدار بایاس نخواهد داشت و در نهایت متوسط بایاس همان بایاس تک تک مدل های شود که ما فرض کردیم b است.

اگر x و y دو متغیر تصادفی باشند که همبستگی داشته باشند

$$var(x+y) = var(x) + var(y) + 2cov(x,y)$$

با استفاده از این variance، $f_{en}(x)$ را محاسبه می کنیم

$$var(f_{en}(x)) = \frac{1}{m^2}\sum_{i=1}^{m} var(f_i(x)) + \frac{2}{m^2}\underbrace{\sum_{i\neq j} cov(f_i(x), f_j(x))}_{m(m-1)\cdot\rho\sigma^2}$$

$$\boxed{var(f_{en}(x)) = \frac{2\rho\sigma^2(m-1)}{m} + \frac{\sigma^2}{m}}$$

پ) • در حوزه یادگیری ماشین، وقتی الگوریتم ها و مدل ها نیاز مند بهینه سازی مشتق پذیر (مثل روش های مبتنی بر گرادیان) اما AdaBoost از خطا ها و وزن دهی های مستقیم دادی به روزرسانی یادگیرنده ها استفاده می کند و نیازی به مشتق گیری ندارد. در این روش، هر یادگیرنده ضعیف می تواند هر مدلی باشد که توانایی عملکرد بهتر از تصادف (به طور دقیق تر، بهتر از 50 درصد) را داشته باشد. معمولاً از درخت های تصمیم به عنوان یادگیرنده استفاده می شود، اما نوع یادگیرنده محدودیتی ندارد

• در مقابل، Bagging به شکلی متفاوت عمل می‌کند؛ مدل‌ها به صورت مستقل و موازی آموزش داده می‌شوند که این کار به محاسبات کمتری نیاز دارد. از سوی دیگر، Bagging با تمرکز روی خطاهای یادگیرنده قبلی، مدل‌های جدیدی تولید می‌کند و این روند به طور تکراری ادامه می‌یابد. به همین دلیل، Boosting معمولاً به تعداد مدل‌های بیشتری نیاز دارد و از لحاظ محاسباتی سنگین‌تر و پیچیده‌تر از Bagging است رفرنس (Zhou, Z.-H (2012). Ensemble methods: Foundations & Algorithms)

سوال 3) الف)

-اگر بدون پیاده‌سازی بخوایم جواب بدیم میشه گفت برای k=1 هر نقطه به نزدیک‌ترین همسایه خود نگاه می‌کنه و بر اون اساس دسته بندی می‌شود پس خطا توی حالت صفره. از اونجایی که در تصویر فاصله زیادی بین نقاط کلاس‌های مختلف وجود داره میشه فهمید که k=1 احتمالاً کمترین مقدار را دارد.

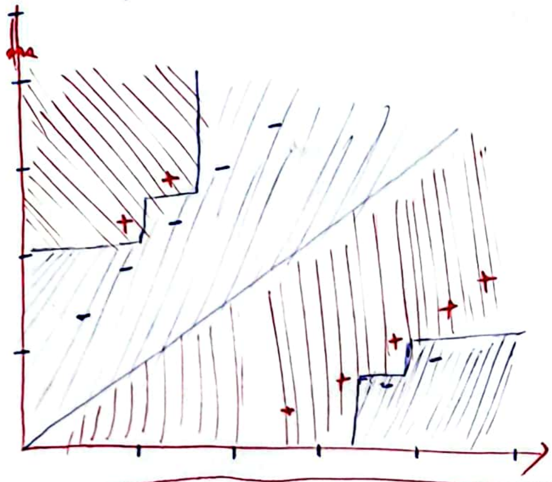
ب) مقدار k در عملکرد مدل knn تاثیر بسیار زیادی اگر k را خیلی کم بگیریم مدل به داده‌های نویزی یا outlier توجه زیادی می‌کند که در نهایت باعث overfit شدن مدل می‌شود و اگر مقدار k را خیلی زیاد بگیریم ممکن است داده‌های 2 کلاس مختلف به اشتباه در 1 دسته قرار بگیرند و مدل underfit بشود

پ) برای kهای مختلف خطای دسته بندی را با روش LOOCV حساب می‌کنیم. با این روش میاییم هر بار یه نمونه از داده‌ها را به عنوان داده تست در نظر می‌گیریم و با بقیه داده‌ها مدل رو آموزش می‌دیم. بعد خطای کلی میشه میانگین خطاهای پیش‌بینی‌ها. برای انجام این کار از LeaveOneOut از کتابخانه sklearn استفاده کردیم.

با نگاه کردن به خروجی کد متوجه شدیم که خطای حدود 0.7 برای kهای کوچک به دست اومده که به دلیل حساسیت به noise و overfitting است که باعث خطای بالا تو LOOCV می‌شود. در k=5 تا k=7 خطا 0.285 است به حداقل مقدار خطا رسیده است که این مقدار k، optimal است

برای kهای بزرگتر حداقل و حتا به بالا) خطا به 1 میرسه که به این دلیل underfit شدن اتفاق میوفته

ت) برای بدست آوردن مرز تصمیم در 1NN از sklearn استفاده کردم و به شکل زیر رسیدیم



سوال 4) الف

مرکز cluster

در واقع label های داده‌ها و μ های خوایم بچینیم μ که باشه رو پیدا تا تابع هزینه حداقل بشه

$$J_i = \sum_{x_j \in S_i} \|x_j - \mu_i\|^2 \longrightarrow$$

برای پیدا کردن مقدار بهینه μ باید از تابع نسبت به μ مشتق بگیریم

$$\frac{\partial J_i}{\partial \mu_i} = \frac{\partial}{\partial \mu_i}\left(\sum_{x_j \in S_j} \|x_j - \mu_i\|^2\right) = 0 \qquad (x_j - \mu_i)^T (x_j - \mu_i)$$

$$\frac{\partial}{\partial \mu_i}\left((x_j - \mu_i)^T (x_j - \mu_i)\right) = -2(x_j - \mu_i)$$

$$\sum -2(x_j - \mu_i) = 0 \longrightarrow -2\left(\sum_{x_j \in S_i} x_j - |S_i| \mu_i\right) = 0$$

$$\longrightarrow \mu_i = \frac{1}{|S_i|} \sum_{x_j \in S_i} x_j$$

بنابراین μ برابر میشود میانگین نقاط داخل cluster

ب) الگوریتم k-means به ~~مقدار~~ مقداردهی اولیه مراکز خوشه‌ها حساس است و اگر این مقداردهی به درستی انجام نشود ممکن است به یک راه حل suboptimal converge کنیم

از آنجایی که شرط همگرایی مرکز cluster اینه که: (فرض می‌کنیم μ مرکز قدیمی cluster j است)

$$\sum_{x \in C_j} \|x^{(i)} - \mu_j\|^2 \geq \sum_{x \in C_j} \|x^{(i)} - \mu_j\| \longrightarrow J_{new} \leq J_{old}$$

چون تابع هزینه J غیرمنفی است و همواره مقدار مثبت دارد از آنجایی که تعداد پارتیشن ها محدود است حداقلی برای J وجود دارد و این تابع نمی تواند به طور بی نهایت کاهش یا بوسیله این فرایند کاهش در یک نقطه متوقف می شود و الگوریتم به وضعیت پایدار می رسد.

به طور کلی الگوریتم K-means به دلیل کاهش مداوم تابع هزینه و وجود حداقل مقدار برای این تابع حتما در یک نقطه به همگرایی می رسد

ب)

دلایل حفظ اختصاص خوشه قبلی:

1. ثبات در خوشه بندی: باعث کاهش تغییرات غیرضروری و ثبات در فرایند خوشه بندی میشه این ثبات خصوصا هنگامی الگوریتم به همگرایی نزدیک می شود مفید است

2. جلوگیری از نوسانات: در صورت برابر بودن فاصله، تخصیص تصادفی باعث رفتار نوسانی می شود. حتی نوسانات ممکن است مانع از همگرایی موثر شود

- مشکلات احتمالی در صورت عدم رعایت این اصل: تاخیر در همگرایی - افزایش هزینه محاسباتی - تابع خوشه بندی sub optimal

رفرنس: Orlowsk et. al 2004

سوال 5)

$f_u(x) = (x^T u)u \longrightarrow$ y تصویر هر نقطه روی u

- خطای MSE:

$$|x(i) - f_u(x(i))|^2 = |x(i) - (x(i)^T u)u|^2 = |x(i)|^2 - 2(x(i)^T u)(u^T x(i)) + (x(i)^T u)^2 |u|^2$$

$$\xrightarrow{|u|^2=1} = |x(i)|^2 - (x(i)^T u)^2$$

- عبارت هدف برای کمینه کردن

$$\frac{1}{m} \sum_{i=1}^{m} \left(|x(i)|^2 - (x(i)^T u)^2\right)$$

- حذف ثابت

$$\xrightarrow[\text{از } u]{|x(i)|^2 \text{ مستقل}} \sum_{i=1}^{m} (x(i)^T u)^2$$

بیشینه کردن از طریق ماتریس کوواریانس

$$C = \frac{1}{m} X X^T$$

اگر X یه ماتریسی باشه که m تا ستون داره ماتریس کوواریانس میشه
هدف اینه که $u^T C u$ رو تحت $u^T u = 1$ است max کنیم که مسئله خارج قسمت رایلی برC است
وقتی این اتفاق میوفته که u بردار ویژه متناظر با بزرگترین مقدار ویژه ی C باشه این نشون میده که u باید مولفه اول اصلی PC1 باشه تا MSE کمینه شه
پس u که MSE رو min می کنه همون مولفه اول اصلی PC1 است که حکم سوال ثابت میشه