دانشکده مهندسی برق

تکلیف کامپیوتری درس بازشناسی آماری الگو، سری اول

استاد

دکتر ابریشمی مقدم

نیمسال اول ۱۴۰۵-۱۴۰۴

مهلت تحویل: ۱۵ آبان ۱۴۰۴

با سلام و آرزوی شادی، موفقیت و سلامتی؛

لطفا در تحویل پاسخ‌های خود موارد زیر را مدنظر داشته باشید:

- فقط برنامه‌هایی که به زبان ترجیحا پایتون و یا متلب باشند قابل قبول خواهند بود.
- تحویل همزمان گزارش و کدها الزامی است.
- گزارش باید شامل خروجی‌های کدهای نوشته شده باشد، که موارد خواسته شده در سوالات هستند و سایر توضیحات خواسته شده دیگر در متن سوالات نیز پاسخ داده شود (از آوردن کد در گزارش خودداری کنید).
- لطفا کدهای برنامه به صورت ماجولار و همراه با توضیحات کافی باشند؛ طوری که بخش‌های مختلف برنامه کاملا قابل تفکیک بوده و اجرا و ارزیابی هر بخش توسط کاربر به آسانی و بدون نیاز به ورود به جزییات برنامه میسر باشد.
- فایل تحویلی پاسخ شما باید تنها یک فایل زیپ، تحت عنوان "SPR_CHW1_Student ID" محتوی دو پوشه باشد. گزارش خود را در پوشه اول با عنوان "Report" و کدهای خود را ترجیحا به فرمت Jupyter Notebook در پوشه "Codes" قرار دهید.
- با این که همکاری، مشورت، و استفاده از ابزارهای کمکی در حل سوالات پیشنهاد می‌شود، حتما به صورت مستقل به نوشتن کدها و گزارش بپردازید.
- ممکن است از دانشجویی خواسته شود در زمانی که تعیین خواهد شد جزئیات کدش را در جلسه‌ای مجازی توضیح دهد، نتایج را تحلیل کند و حتی تغییراتی در پارامترهای کد اعمال کند. در صورتی که دانشجویی تمارین را تحویل داده باشد اما نتواند کد خود را توضیح دهد و یا تغییراتی روی آن اعمال کند، و یا اینکه کد یا گزارش تحویلی به پاسخ دیگران شباهت غیرمنطقی داشته باشد، نمره تمرین صفر لحاظ شده و نمره‌ای منفی هم لحاظ خواهد شد.
- در صورت وجود هرگونه سوال یا ابهام، مشکل مربوطه را با آی دی تلگرام زیر در میان گذارید:

@omid_Emaa

در این تمرین از مجموعه داده "Seeds"[1] استفاده خواهیم کرد. این مجموعه شامل خواص هندسی دانه‌های متعلق به سه نوع مختلف گندم می‌باشد. پس از پیش‌پردازش داده‌ها می‌خواهیم با استفاده از تبدیل PCA راستاهایی را بیابیم که در آن داده‌ها بیشترین پراکندگی را داشته باشند. سپس فضای ویژگی را به ۲ بعد کاهش دهیم و در نهایت مقدمات را برای مشخص کردن مرزهای تصمیم‌گیری بین کلاس‌ها فراهم کنیم.

بخش اول (۱۰ امتیاز):

۱: ابتدا با استفاده از توابع و کتابخانه‌های مناسب (برای مثال کتابخانه pandas) داده‌ها را در محیط برنامه‌نویسی بازخوانی کنید، آن را ذخیره کرده و ابعاد آن را نمایش دهید.

۲: ستون class را در متغیر دیگری ذخیره کرده، این ستون را از ماتریس داده‌ها حذف کنید. سپس تعداد نمونه‌ها، تعداد ویژگی‌ها و نام ویژگی‌ها را گزارش دهید.

بخش دوم (۳۰ امتیاز):

۳: ماتریس کواریانس تمام نمونه‌ها را محاسبه کنید و گزارش دهید.

۴: مقادیر ویژه و بردارهای ویژه ماتریس کواریانس به‌دست‌آمده را با استفاده از توابع آماده به دست آورید و گزارش دهید.

۵: تبدیل PCA را بدون استفاده از توابع آماده و با استفاده از مقادیر و بردارهای ویژه اعمال کنید و فضای ویژگی را به دو بعد برسانید.

۶: حال در فضای تبدیل شده (دو بعدی) نمونه‌ها را به تفکیک کلاس‌ها نمایش دهید (هر کلاس یک رنگ داشته باشد و این رنگ در ادامه تمرین نیز ثابت بماند).

۷: حال تبدیل PCA را با استفاده از تابع آماده موجود در کتابخانه sklearn به دست آورده و در فضای تبدیل‌شده نمونه‌ها را به تفکیک کلاس‌ها نمایش دهید و با قسمت ۵ مقایسه کنید.

---

[1] https://archive.ics.uci.edu/dataset/236/seeds

۸: مجموع مقادیر ویژه‌هایی که در تبدیل PCA مشارکت داشته‌اند را محاسبه کنید. مجموع کل مقادیر ویژه‌ها را نیز محاسبه کنید. امید ریاضی نسبی مربعات خطای ایجاد شده در اثر کاهش ابعاد را محاسبه کنید (این خطا برابر یک منهای نسبت مجموع مقادیر ویژه حاضر در تبدیل، به مجموع کل مقادیر ویژه است).

۹: منحنی توزیع‌های حاشیه‌ای هر کلاس را با رنگ ویژه آن کلاس رسم کنید (دو منحنی به ازای هر کلاس مدنظر است).

بخش سوم (۳۰ امتیاز):

۱۰: در فضای دو بعدی بدست آمده از PCA ، پارامترهای توزیع کلاسها و کل نمونه‌ها را جداگانه تخمین بزنید (بردار میانگین و ماتریس کوواریانس). روابط مورد نیاز تخمین میانگین و ماتریس کواریانس نمونه‌ها به‌صورت زیر هستند:

$$\hat{\mu} = \frac{1}{N}\sum_{i=1}^{N} x_i$$

$$\hat{\Sigma} = \frac{1}{N-1}\sum_{i=1}^{N}(x_i - \hat{\mu})(x_i - \hat{\mu})^t$$

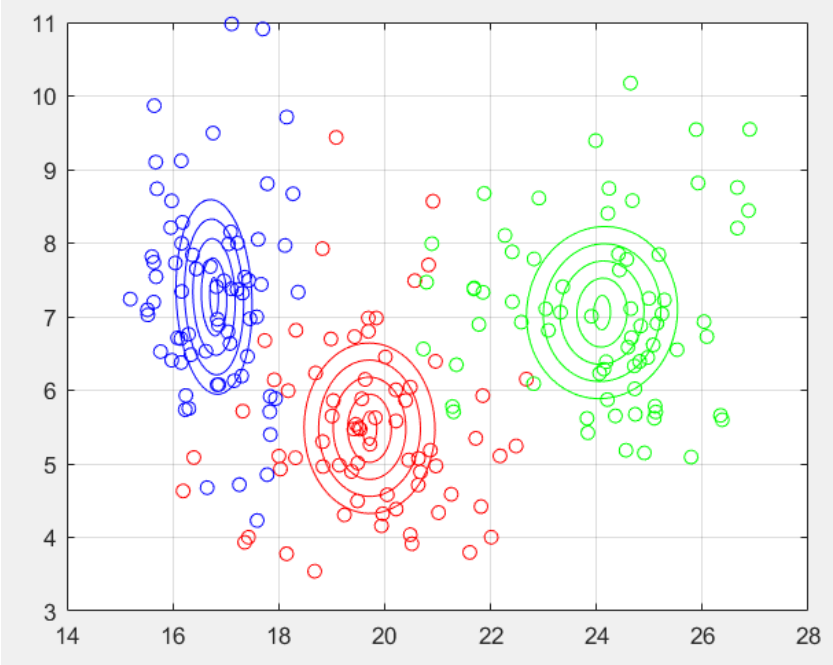
در روابط فوق $\hat{\mu}$ و $\hat{\Sigma}$ به ترتیب معرف تخمین بردار میانگین و تخمین ماتریس کواریانس هستند، و N معرف تعداد نمونه‌ها به ازای هر کلاس (چنانچه تخمین میانگین یا کواریانس هر کلاس مدنظر باشد) یا تعداد کل نمونه‌ها (چنانچه تخمین میانگین و کواریانس کل نمونه‌ها مدنظر باشد) است.

۱۱: معکوس ماتریس‌های کواریانس را محاسبه کنید.

۱۲: با فرض گوسی بودن توزیع کلاس‌ها و با توجه به پارامترهای به‌دست آمده، کانتور توزیع هر یک از انواع دانه‌ها را در کنار هم رسم کنید.

توجه کنید باید یک مجموعه نقطه با مکان هندسی $(x - \hat{\mu})^t\hat{\Sigma}^{-1}(x - \hat{\mu}) = k$ به ازای:

$k = \{0.1, 0.3, 0.5, 0.7, 0.9\}$ را برای هر کلاس رسم کنید. خروجی شکلی مشابه پلات زیر خواهد بود:



بخش چهارم (۶۰ امتیاز):

Self-Organizing-Map که به اختصار SOM هم نامیده می‌شود، یک تکنیک شبکه عصبی unsupervised است که هدفش نگاشت یک فضای n بعدی به یک فضایی با بعد پایین‌تر (معمولاً گرید دوبعدی) است، به‌طوری‌که ساختار هندسی و روابط توپولوژیک داده تا حد ممکن حفظ شوند. خروجی معمولاً شبکه‌ای از نورون‌هاست که هر نورون دارای یک بردار وزن است.

۱۳: در مورد این روش مطالعه کرده و به طور خلاصه نحوه کارکرد آن برای رسیدن به مهم کاهش بعد را توضیح دهید.

۱۴: یک SOM دوبعدی با شبکه مستطیلی (اندازه ۱۰×۱۰) با استفاده از کتابخانه minisom بسازید. وزن‌های اولیه را به‌صورت تصادفی مقداردهی کنید. آموزش را با نمونه‌های داده‌ها و بدون استفاده از برچسب class انجام دهید. الگوریتم باید شامل: انتخاب تصادفی یک ورودی نمونه x، یافتن BMU (Best Matching Unit) با معیار فاصله اقلیدسی بین x و وزن‌های نورون‌ها، بروزرسانی وزن BMU و نورون‌های همسایه بر اساس یک تابع همسایگی و نرخ یادگیری که با زمان کاهش می‌یابد، باشد. می‌توانید برای این کار از تابع Minisom استفاده کنید.

```
# --- Create SOM (10x10 grid) with random initialization ---
som = MiniSom(x=10, y=10,.....)
```

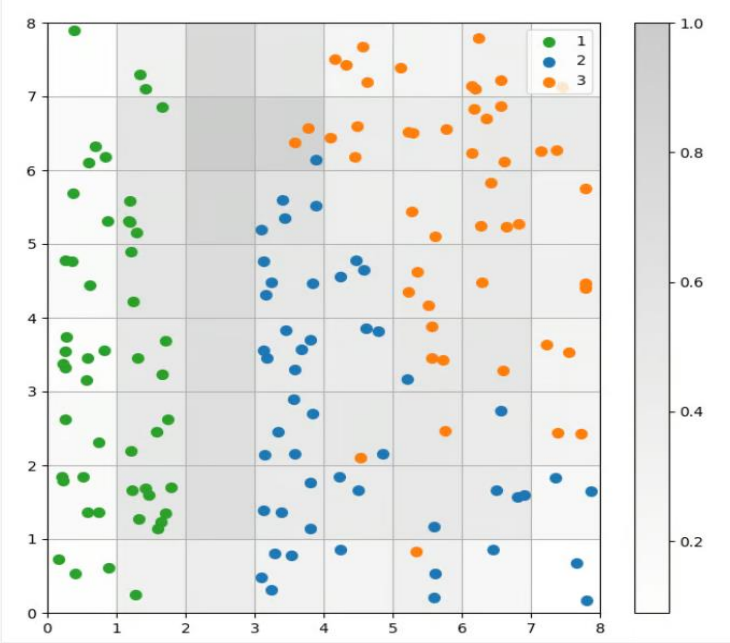
۱۵: ماتریس U یا همان U-matrix را ترسیم کرده و نتایج را تفسیر کنید.

۱۶: همان ماتریس U را بار دیگر همراه با برچسب کلاس ترسیم کرده و نتایج را تفسیر کنید.

۱۷: scatter plot داده ها بر روی شبکه SOM را ترسیم کرده و نتایج را تفسیر کنید.

راهنمایی : باید مختصات نورون برنده برای هر نمونه داده یافته شده، ماتریس U ترسیم شده و یک scatter plot از تمامی نورون‌های برنده برای هر نمونه داده کشیده شود. برای مثال، خروجی برای دیتاستی دیگر(دیتاست Iris) و با شبکه مستطیلی ۸ * ۸ به شکل زیر در آمده است :

```
# code guide for this part ////// get the X and Y coordinates of the winning neuron for each data -
#point w_x, w_y = zip(*[som.winner(d) for d in data])
w_x = np.array(w_x)
w_y = np.array(w_y)
# plot the distance map(U-matrix)
plt.figure(...)
plt.pcolor(som.distance_map().T, .....)
plt.colorbar()
# make a scatter plot of all the winning neurons for each data point
for c in np.unique(target):
    .....
plt.legend(loc='upper right')
plt.grid()
plt.show()
```



"موفق باشید"