شناسایی آماری الگو

تمرین های سری شش

فرهاد دلیرانی
۹۶۱۳۱۱۲۵

dalirani@aut.ac.ir
dalirani.1373@gmail.com

تمام‌کدها با پایتون 3.6 نوشته شده‌اند.

همچنین از پکیج‌های زیر استفاده کرده ام:

- numpy
- matplotlib

البته برای راحتی در نصب پایتون 3.6 و پکیج های مربوط به دیتاساینس که numpy و matplotlib هم جزیی از آن پکیج‌ها هستند از Anaconda 5.0.0 استفاده کرده‌ام که همه‌ی موارد گفته شده را بدون دردسر و سختی نصب می کند. تنها کافی است آن را از https://www.anaconda.com/download دانلود کنید و Installer باقی کار را انجام می‌دهد. البته به صورت مستقل هم، می‌توان آن‌ها را نصب کرد.

زبان برنامه نویسی: پایتون 3.6
پکیج‌ها: پکیج‌های گفته شده را برای راحتی در نصب با Anaconda نصب کردم.
ورژن Anaconda من: Anaconda 5.0.0 For Linux Installer که البته همین ورژن برای سایر سیستم عامل‌ها هم موجود است.
محیط برنامه نویسی: pyCharm Community Edition

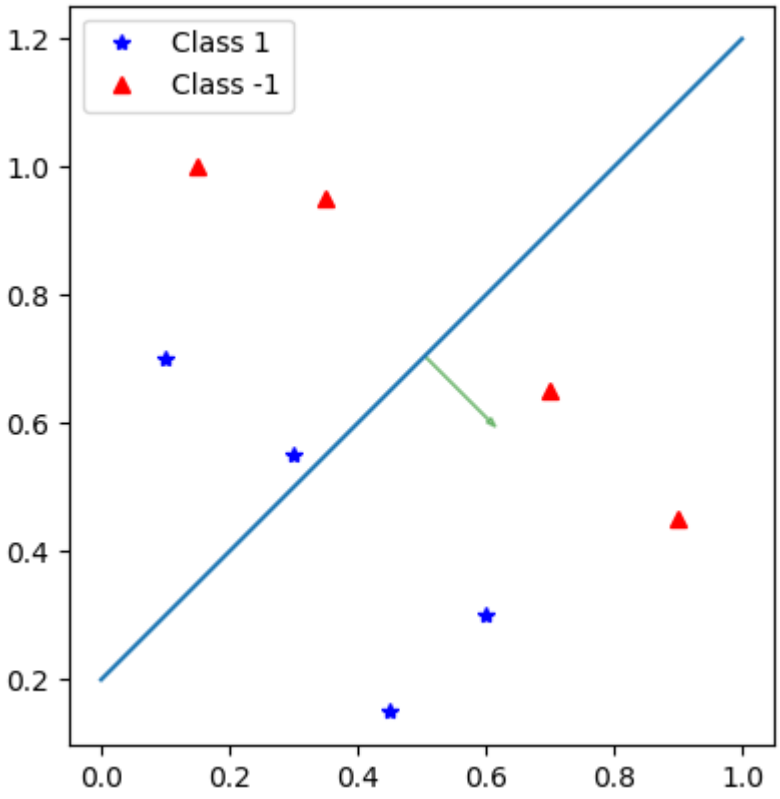
سوال ۱)

کدهای این بخش در فایل problem1.py موجود است.

بخش A) مرز تصمیم برابر است با `Transpose(w)*x + w0 = 0` که در ابتدا برابر است با

```
[1 -1][x0; x1] + 0.2 = 0
```

که در تصویر زیر داده‌ها و مرز تصمیم رسم شده را مشاهده می‌کنید:



همین طور که مشاهده می‌شود 4 نمونه اشتباه دسته‌بندی شده اند. دو نمونه‌ی قرمز سمت راست مرز تصمیم و دو نمونه‌ی آبی سمت چپ مرز تصمیم به اشتباه دسته بندی شده اند.

بخش B)

یکی از نقطه‌هایی که اشتباه دسته‌بندی شده است، نقطه‌ی (0.1,0.7) است که وزن‌ها را طبق `perceptron single sample rule` آپدیت می‌کنیم:

$$\mathbf{a}^{k+1} = \mathbf{a}^{k} + \eta * \mathbf{y}$$

که قبل از آپیدت میزان a برابر است با

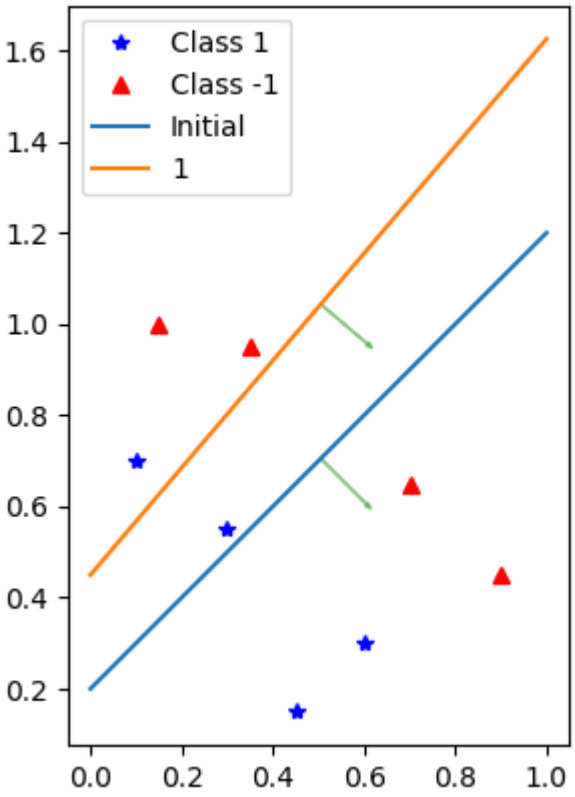
```
[ 0.2  1.  -1. ]
```

بعد از آپدیت مقدار a برابر است با

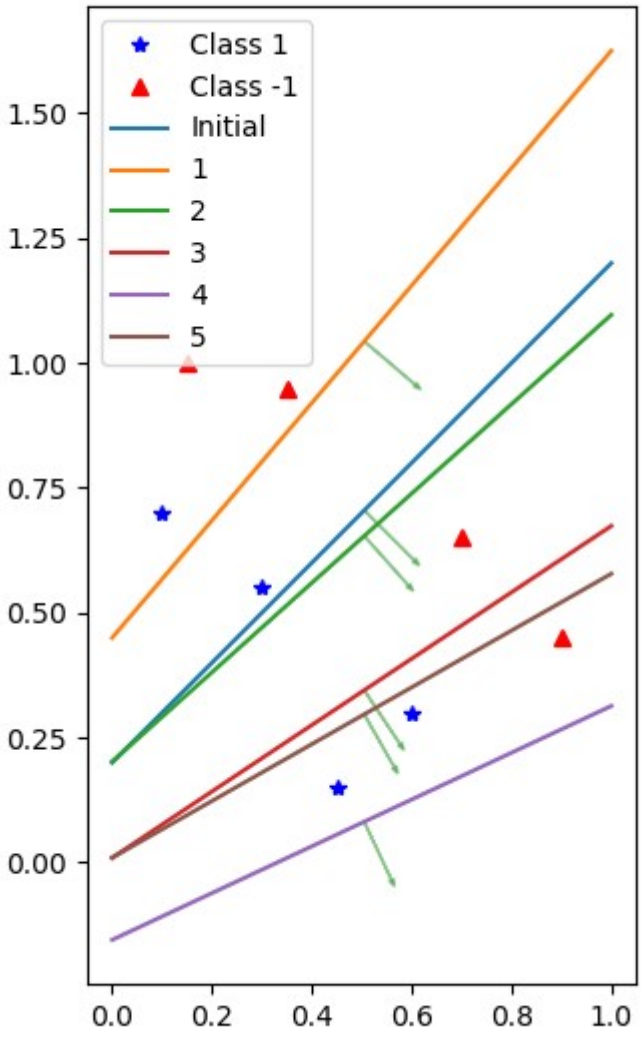
```
[ 0.5   1.03 -0.79]
```

ضریب یادگیری برابر 0.19 است.

در تصویر زیر مرز جدید را مشاهده می‌کنید:



بخش C) همان کار بخش B را چهار بار دیگر تکرار می‌کنیم:



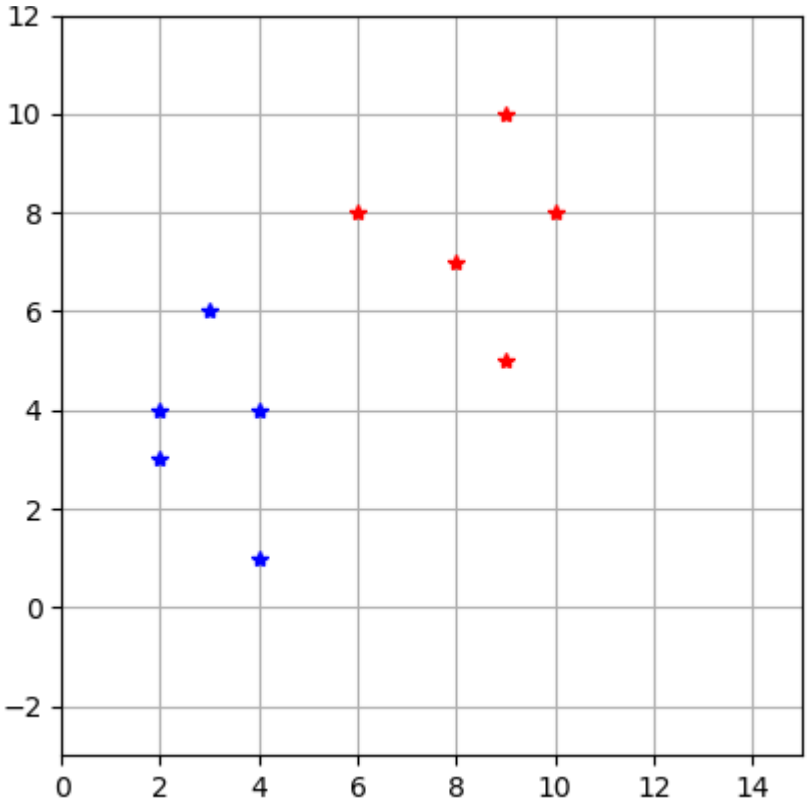
که در هر بخش نقطه‌های زیر انتخاب شده‌اند و وزن‌ها را قبل و بعد از آپدیت مشاهده می‌کنید:

```
Misclassified Point: [ 1.   0.1  0.7]
Weight before update: [ 0.2  1.  -1. ]
Weight after update: [ 0.39   1.019 -0.867]

Misclassified Point: [-1.   -0.7  -0.65]
Weight before update: [ 0.39   1.019 -0.867]
Weight after update: [ 0.2    0.886  -0.9905]

Misclassified Point: [-1.   -0.9  -0.45]
Weight before update: [ 0.2    0.886  -0.9905]
Weight after update: [ 0.01   0.715 -1.076]

Misclassified Point: [-1.   -0.9  -0.45]
Weight before update: [ 0.01   0.715 -1.076]
Weight after update: [-0.18    0.544  -1.1615]

Misclassified Point: [ 1.    0.3   0.55]
Weight before update: [-0.18    0.544  -1.1615]
Weight after update: [ 0.01   0.601 -1.057]
```

سوال ۲)

کدهای این بخش از سوال در فایل `problem2.py` موجود است.

بخش a) در شکل زیر داده‌های رسم شده را مشاهده می‌کنید:



بخش B) در این بخش LDA را محاسبه می‌کنیم:

میانگین دو کلاس را اینگونه به دست می‌آوریم:

$$\mathbf{m}_i = \frac{1}{n_i} \sum_{\mathbf{x} \in \mathcal{D}_i} \mathbf{x}$$

که برابر می‌شود با:

```
Mean Class 1:
 [[ 3. ]
 [ 3.6]]

Mean Class 2:
 [[ 8.4]
 [ 7.6]]
```

برای محاسبه‌ی Scatter تو کلاس اینگونه عمل می‌کنیم:

$$\mathbf{S}_i = \sum_{\mathbf{x} \in \mathcal{D}_i} (\mathbf{x} - \mathbf{m}_i)(\mathbf{x} - \mathbf{m}_i)^t$$

که برای کلاس یک و دو برابر می‌شود با

```
Scatter Class 1:
 [[  4.   -2. ]
 [ -2.   13.2]]

Scatter Class 2:
 [[  9.2  -0.2]
 [ -0.2  13.2]]
```

و within class scatter را اینگونه محاسبه می‌کنیم:

$$\mathbf{S}_W = \mathbf{S}_1 + \mathbf{S}_2$$

که برابر می‌شود با

```
Scatter within(Sw):
 [[ 13.2  -2.2]
 [ -2.2  26.4]]
```

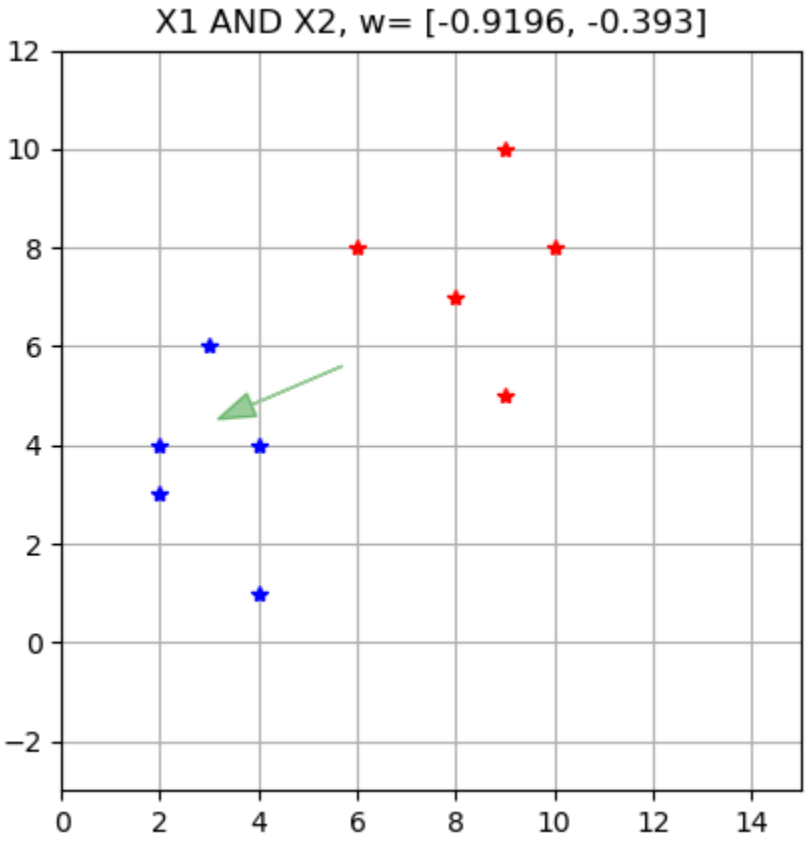
برای محاسبه‌ی بردار w به صورت زیر عمل می‌کنیم:

$$\boxed{\mathbf{w} = \mathbf{S}_W^{-1}(\mathbf{m}_1 - \mathbf{m}_2).}$$

که برابر می‌شود با

```
W:
 [[-0.44046095]
 [-0.18822023]]

Normalized W:
 [[-0.91955932]
 [-0.39295122]]
```

در تصویر زیر بردار w را مشاهده می‌کنید:



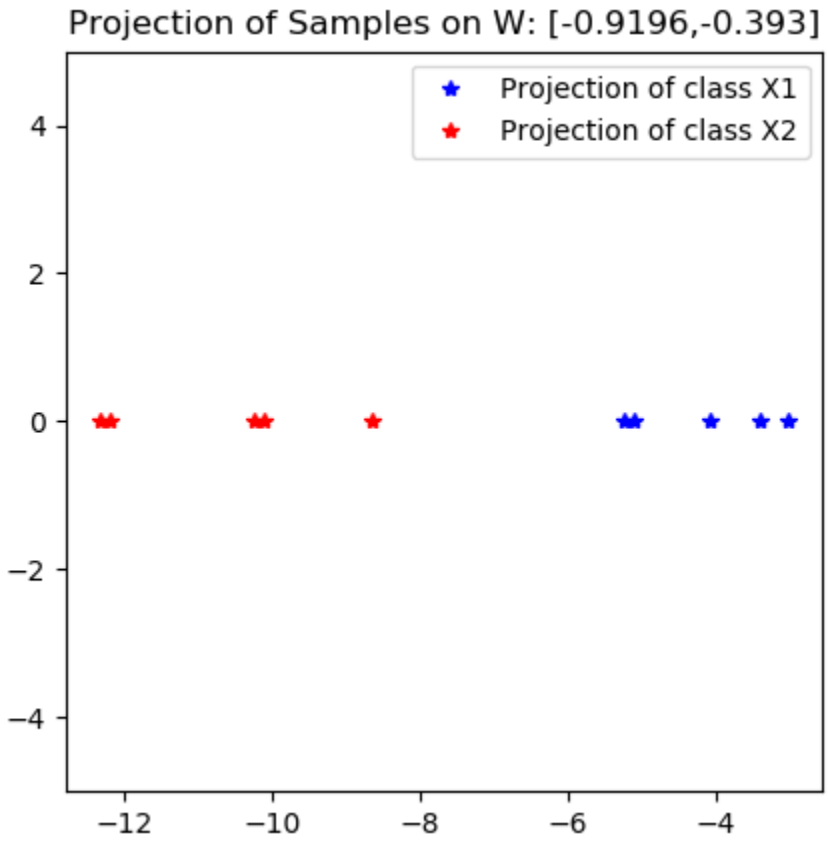
بخش c)برای نگاشت داده‌ها به W از رابطه‌ی زیر استفاده می‌کنیم:

$$y = \mathbf{w}^t \mathbf{x}$$

که در شکل زیر مقادیر عددی نقطه‌های پروجکت شده را می‌بینید:

```
Projection of Class1:
 [[ 4.07118849  3.41092352  3.0179723   5.11638527  5.25004215]]

Projection of Class2:
 [[ 12.20554606   8.66096567  10.24078996  10.10713308  12.33920294]]
```

در تصویر زیر نقاط پروجک شده را مشاهده می‌کنید:



بخش D) همان‌طور که در شکل بالا مشاهده می‌کنید در فضای جدید یک بعدی که داده‌ها به آن نگاشت شده‌اند، داده‌های هر دو کلاس کاملا از هم جدا پذیر هستند و جدا پذیری داده‌ها بر روی این خط ماکسیموم است زیر LDA جواب بهینه را برای جدا پذیری می‌دهد زیرا فاصله‌ی میانگین دو کلاس نسبت به مجموع scatter ها راحداکثر می‌کند.