درس شناسایی الگو

گزارش پروژه شماره 1

پردازش الگوی آماری

نام و نام خانوادگی دانشجو:

فاطمه پارسا

آذر 1399

Contents

مقدمه

یک مسئله دسته بندی با دو کلاس را در نظر بگیرید. به گونه ای که هر کلاس با یک توزیع گوسی دو بعدی مدل شده است. قصد داریم در این پروژه بر اساس ماتریس کواریانس و میانگین کلاس ها حالت های گوناگون دسته بند بیز را بررسی و مقایسه کنیم.

مسئله شماره 1

با استفاده از پارامتر های زیر 100000 نمونه از هر توزیع تولید کرده ایم. هر(x,y) یک بردار ویژگی است.

$$\mu_1 = \begin{bmatrix}1\\1\end{bmatrix} \quad \Sigma_1 = \begin{bmatrix}1 & 0\\0 & 1\end{bmatrix} \quad \mu_2 = \begin{bmatrix}4\\4\end{bmatrix} \quad \Sigma_2 = \begin{bmatrix}1 & 0\\0 & 1\end{bmatrix}$$

(تولید سمپل با استفاده از ماژول gaussian_generator_1A)

1_الف

فرض میکنیم $p(\omega_1)=p(\omega_2)$

1_الف_1

طراحی دسته بند بیز بر اساس کمترین خطا

احتمال های پیشین و ماتریس کواریانس هر دو کلاس با هم برابر هستند و ماتریس کواریانس هر دو برابر $\sigma 2I$ استبر این اساس می توان به منظور طراحی یک دسته بند برای هر کلاس تابع $g_i = \mathrm{Ln}\ p(x_i|\omega_i)$ را به دست آورد. توزیع کلاس ها بر حسب (x,y) یعنی همان احتمال شرطی کلاس ها گوسی است بنابراین $p(x_i|\omega_i)$ برابر است با :

$$\frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}}\exp\left[-\frac{1}{2}(\mathrm{x}-\boldsymbol{\mu})^t\Sigma^{-1}(\mathrm{x}-\boldsymbol{\mu})\right]$$

با محاسبه gi های هر دو کلاس و مساوی هم قرار دادن آنها به معادله خط دسته بند می رسیم. ماژول classifier_1A.py محاسبات مربوط به محاسبه معادله دسته بند را با داشتن ماتریس کواریانس و میانگین داده ها و فرمول توزیع گوسی انجام داده و تابع خطی جدا کننده این دو کلاس را به دست می آورد و در خروجی به صورت زیر اعلام می کند:

```
the classifier line is:  0 = -x - y + 5.0
```

1_الف_2

ترسیم دسته بند بیز به همراه نمونه های تولید شده ی هر دو کلاس یک و دو و تفسیر عملکرد دسته بند

ماژول classifier_1A با استفاده از معادله y=-x+5 به دست آمده خط دسته بند سبز رنگ را رسم میکند. نمونه های پایین خط به کلاس یک و نمونه های بالای خط به کلاس دو تعلق می گیرد. نمونه هایی که روی خط قرار دارند را به صورت تصادفی دسته بندی می کنیم. این خط عمود بر خط واصل دو کلاس بوده و دقیقا از وسط آن می گذرد.

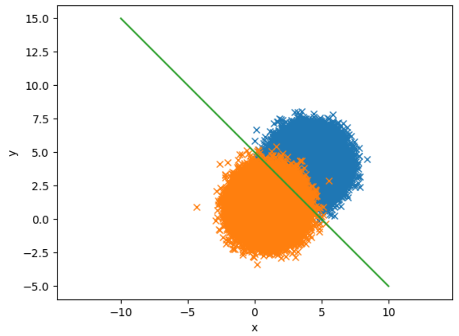

Figure 1 کلاسیفایر خطی حالت اول احتمالات برابر و واریانس یک

1_الف_3

گزارش تعداد داده هایی که اشتباه دسته بندی شده اند برای هر کلاس

تمام نمونه هایی که متعلق به کلاس یک هستند اما در بالا یا روی خط قرار گرفته اند و به صورت تصادفی دسته آنها اشتباها دو تعیین شده است در کلاس یک اشتباه دسته بندی شده اند.(کلاس دو به طور مشابه) ماژول misclassified_samples_1A.py مختصات تمام داده های تولید شده و دسته های آن ها را از ورودی گرفته و با استفاده از معادله دسته بند به دست آمده در قسمت قبل کلاس آنها را محاسبه وبا کلاس اصلی داده مقایسه می کند. اگر کلاس

محاسبه شده برابر با کلاس اصلی داده نباشد به عنوان یک داده اشتباه کلاسیفای شده شمارش می شود. خروجی مربوط به اجرای این برنامه برای داده های تولید شده در قسمت اول به صورت زیر است:

```
c1 misclassified number is:  1712
c2 misclassified number is:  1722
c2 misclassified number is:  3434
```

1_الف_4

ترسیم کران چرنوف به عنوان تابعی از β و پیدا کردن β بهینه برای خطای کمینه

در ماژول p(error)_chernoff_bhattacharyya_1A.py تابع p(error) را با نام perror_chernoff_bound(B) بر حسب β نوشتیم که مقدار p(error) را بر می گرداند . با محاسبه 10000 p(error) به ازای 10000 β متفاوت بین صفر و یک، نمودار p(error) بر حسب β را ترسیم میکنیم. با پیدا کردن مقدار کمینه p(error) مقدار β بهینه را به دست می آوریم. خروجی برای این دسته از داده ها به این صورت بود: β بهینه = 0.49 و p(error)= 0.07

```
min B in is: 0.4999499949995
so the p(error) for this B by chernoff bound is : 0.07452850809672486
```

نمودار ترسیم شده p(error) بر حسب β به شکل زیر است:

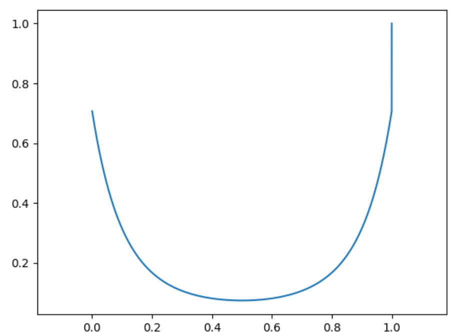

Figure2 نمودار خطا بر حسب بتا کران چرنوف

برای محاسبه کران باتاچاریا کافیست β را 0.5 قرار دهیم. که تقریب بسیار خوبی برای β بهینه است. خروجی اجرای تابع perror_chernoff_bound(B) به ازای β=0.5

```
p(error) when B=0.5 by bhattacharyya bound is: 0.07452850
```

1_ب

فرض می کنیم p(w$_1$)=0.2 و p(w$_2$)=0.8

1_ب_1

طراحی دسته بند بیز بر اساس کمترین خطا

توضیحات و فرمول های قسمت 1_الف_1 برای این قسمت نیز صادق است. با این تفاوت که این بار قسمت دوم فرمول x_0 صفر نمیشود:

$$-\frac{\sigma^2}{\|\boldsymbol{\mu}_i-\boldsymbol{\mu}_j\|^2}\ln\frac{P(\omega_i)}{P(\omega_j)}(\boldsymbol{\mu}_i-\boldsymbol{\mu}_j)$$

ماژول classifier_1B.py محاسبات را انجام می دهد و در نهایت خط دسته بند به صورت زیردر می آید:

```
the classifier line is:  0 = -x - y + 4.5379018
```

1_ب_2

ترسیم دسته بند بیز به همراه نمونه های تولید شده ی هر دو کلاس یک و دو و تفسیر عملکرد دسته بند

با استفاده از معادله خط به دست آمده در مرحله قبل خط قرمزرا رسم می کنیم. این خط دسته بند جدید است و خط سبز دسته بند قسمت الف است. همانطور که مشخص است به دلیل اینکه **p (w$_1$) < p (w$_2$)** خط دسته بند به سمت دسته یک نزدیکتر شده است. و محدوده داده هایی که در در دسته دو قرار می گیرند بیشتر شده است.

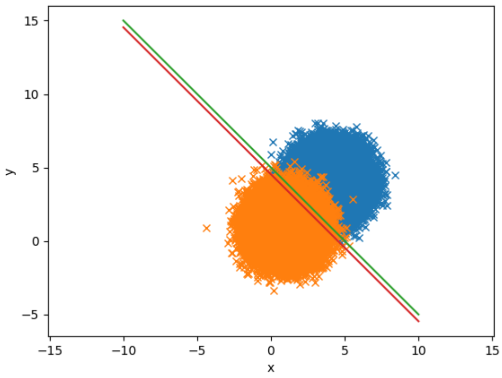

Figure 3کلاسیفایر خطی حالت دوم وقتی احتمال ها با هم برابر نیست.

1_ب_3

گزارش تعداد داده هایی که اشتباه دسته بندی شده اند برای هر کلاس

با ماژول misclassified_samples_1B تعداد داده هایی که اشتباه دسته بندی شده اند را برای تمام سمپل های تولید شده حساب میکنیم و میبینیم که تعداد غلط ها زیادتر از قسمت 1_الف_3 شده و تعادل بین دو کلاس به هم ریخته. دلیل این اتفاق این است که ما کلاسیفایر را برای داده ها با احتمال های متفاوت ساختیم اما در این تست تعداد داده های هر دوکلاس برابر بود. اگر همین عمل را با تعداد داده های متعادل بر اساس احتمال رخداد آنها انجام دهیم نتیجه شبیه قسمت 1_الف_3 می شود.

```
c1 misclassified number is:  3688
c2 misclassified number is:  710
c2 misclassified number is:  4398
```

1_ب_4

ترسیم کران چرنوف به عنوان تابعی از β و پیدا کردن β بهینه برای خطای کمینه

```
min B in is: 0.577057705770577
so the p(error) for this B by chernoff bound is : 0.0565233
```

از ماژول p(error)_chernoff_bhattacharyya_1B.py استفاده می کنیم.

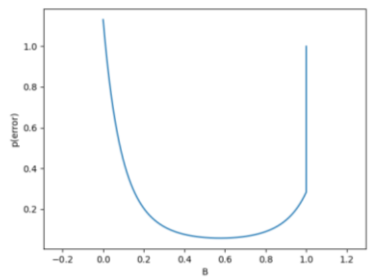

Figure 4مقدار p(error) بر حسب بتا از طریق کران چرنوف

1_ب_5

محاسبه کران باتاچاریا

```
bhattacharyya bound is: 0.059622
```

مسئله شماره 2

مانند مسئله 1 و این بار با استفاده از پارامتر های جدید زیر 100000 نمونه از هر توزیع تولید کرده ایم. قسمت الف و ب سوال یک را دوباره و این بار با پارامتر های جدید انجام می دهیم. (تولید داده ها توسط ماژول gaussian_generator_2A.py)

$$\mu_1 = \begin{bmatrix}1\\1\end{bmatrix} \quad \Sigma_1 = \begin{bmatrix}1 & 0\\0 & 1\end{bmatrix} \quad \mu_2 = \begin{bmatrix}4\\4\end{bmatrix} \quad \Sigma_2 = \begin{bmatrix}4 & 0\\0 & 8\end{bmatrix}$$

2_الف

فرض میکنیم $p(w_1)=p(w_2)$

2_الف_1

طراحی دسته بند بیز بر اساس کمترین خطا

ماژول classifier_2A.py با استفاده از فرمول های مربوط به منحنی گوسی و پارامتر های کلاس ها، معادله منحنی کلاسیفایر را به صورت زیر در خروجی اعلام می کند:

$$-0.125x^2 - 0.145833333333333y^2 + 0.166666666666667y + 1.82191196759991=0$$

2_الف_2

ترسیم دسته بند بیز به همراه نمونه های تولید شده ی هر دو کلاس

با ترسیم معادله به دست آمده در مرحله قبل شکل زیر به دست می آید. داخل بیضی کلاس یک و خارج آن کلاس دو است.

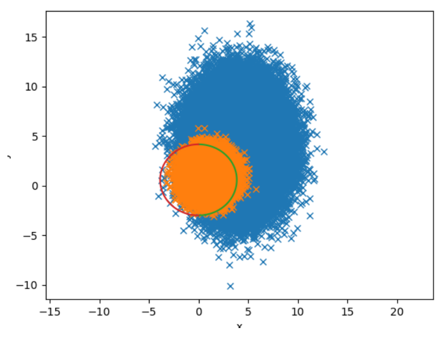

Figure 5کلاسیفایر بیز در حالتی که واریانس ها با هم متفاوتند

2_الف_3

گزارش تعداد داده هایی که اشتباه دسته بندی شده اند برای هر کلاس

تعداد داده هایی که اشتباه دسته بندی شده است را مانند سوال یک و این بار بر اساس تابع کلاسیفایر جدید به دست می آوریم. از حاصل اجرای ماژول misclassified_samples_2A.py خروجی به شکل زیر است:

```
c1 misclassified number is:  16691
c2 misclassified number is:  41950
c2 misclassified number is:  58641
```

2_الف_4

ترسیم کران چرنوف به عنوان تابعی از β و پیدا کردن β بهینه برای خطای کمینه

مینیموم خطا را بر اساس کران چرنوف را با ماژول perror_chernoff_bound_2A.py مشابه سوال یک اما برای داده های جدید به دست می آوریم :

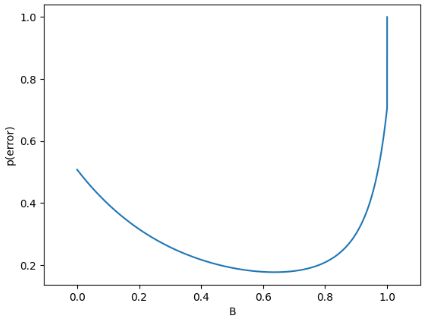

Figure6 p(error) بر حسب بتا از طریق کران چرنوف

```
min B in is: 0.6370637063706371
so the p(error) for this B by chernoff bound is : 0.177009362
```

2_الف_5

محاسبه کران باتاچاریا

برای محاسبه کران باتاچاریا کافیست β را 0.5 قرار دهیم. که تقریب بسیار خوبی برای β بهینه است. خروجی اجرای تابع perror_chernoff_bound(B) به ازای β=0.5

```
p(error) when B=0.5 by bhattacharyya bound is: 0.19099
```

2_ب

فرض می کنیم $p(w_1)=0.2$ و $p(w_2)=0.2$

2_ب_1

طراحی دسته بند بیز بر اساس کمترین خطا

مشابه قسمت های قبل حاصل اجرای ماژول classifier_2B.py یک منحنی کلاسیفایر میدهد.

```
the classifier line is:  0 = -0.125*x**2 - 0.145833333333333*y**2 +
0.166666666666667*y + 1.35981384722661
```

2_ب_2

ترسیم دسته بند بیز به همراه نمونه های تولید شده ی هر دو کلاس یک و دو و تفسیر عملکرد دسته بند

بیضی حاصل از معادله قبل را رسم کرده ایم. (بیضی کوچکتر). تفاوت این بیضی با بیضی قسمت 2_الف_2 در این است که بیضی جدید کوچک تر شده. به دلیل اینکه احتمال کلاس دو بیشتر شده و محدوده برای کلاس دو بیشتر میشود. در تصویر زیر برای مقایسه بهتر هر دو بیضی آورده شده است.

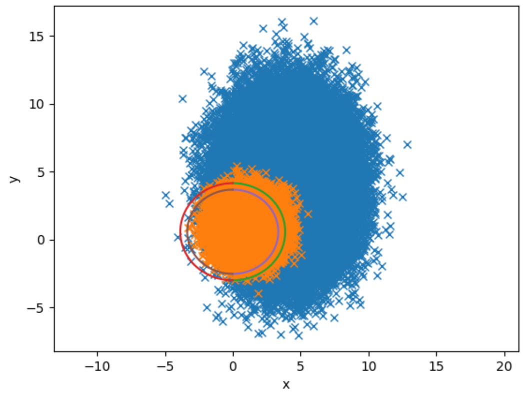

Figure7 کلاسیفایر بیز در حالتی که احتمال ها و واریانس ها متفاوتند

2_ب_3

گزارش تعداد داده هایی که اشتباه دسته بندی شده اند برای هر کلاس

ماژول misclassified_samples_2B.py

```
c1 misclassified number is:  18192
c2 misclassified number is:  40258
c2 misclassified number is:  58450
```

2_ب_4

ترسیم کران چرنوف به عنوان تابعی از β و پیدا کردن β بهینه برای خطای کمینه

نمودار مربوط به p(error) بر حسب کران چرنوف و بتا:(ماژول perror_chernoff_2B.py)

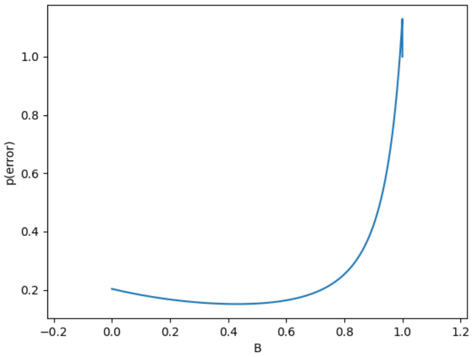

Figure 8 کران چرنوف

```
min B in is: 0.42994299429942995
so the p(error) for this B by chernoff bound is : 0.150940191
```

2_ب_5

محاسبه کران باتاچاریا

برای محاسبه کران باتاچاریا کافیست β را 0.5 قرار دهیم. که تقریب بسیار خوبی برای β بهینه است.

خروجی اجرای تابع perror_chernoff_bound(B) به ازای β=0.5

```
p(error) when B=0.5 by bhattacharyya bound is: 0.15279604
```

مسئله شماره 3

کلاسیفای کردن قسمت 2_ب این بار با تابع کمترین فاصله:

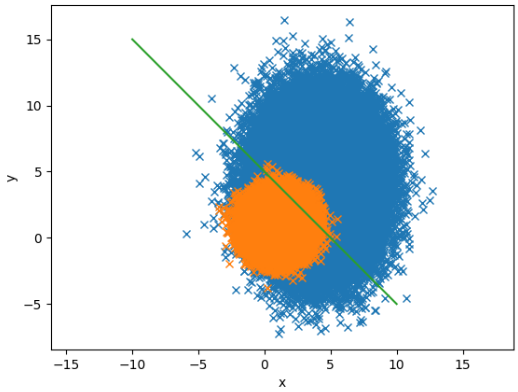

Figure 9 کلاسیفایر مینیموم فاصله در حالتی که احتمال ها برابر اند

```
c1 misclassified number is:  1704
c2 misclassified number is:  19501
c2 misclassified number is:  21205
```