

**Amirkabir University of Technology
(Tehran Polytechnic)**



**Department of
Computer Engineering**

Course : Statistical Pattern Recognition

Homework 3

Najmeh Mohammadbagheri

99131009

گزارش تمرین

سوال یک

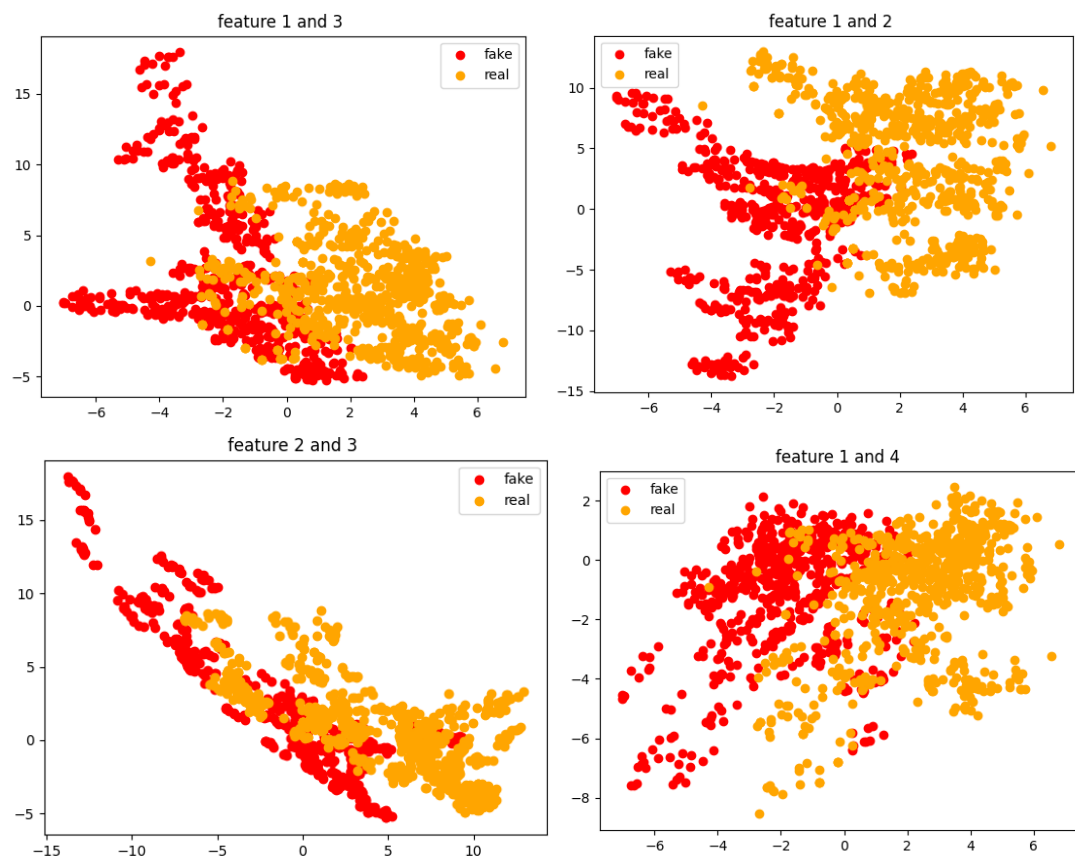
در فایل دستی نوشته شده است.

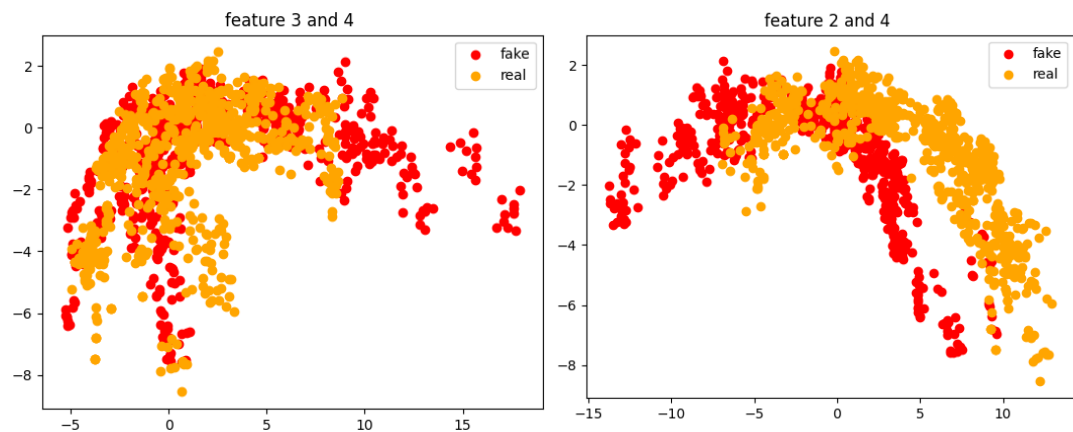
سوال دو

در فایل دستی نوشته شده است.

سوال سه

قسمت (a) ویژگی ها را دو به دو باهم رسم می کنیم و میزان درهم رفتگی دو کلاس واقعی و جعلی را بررسی می کنیم. هر دو ویژگی ای که میزان درهم رفتگی کمتری داشتند به عنوان بهترین ویژگی ها برای تصمیم گیری انتخاب می شوند. در این قسمت ویژگی یک و دو انتخاب می شود.



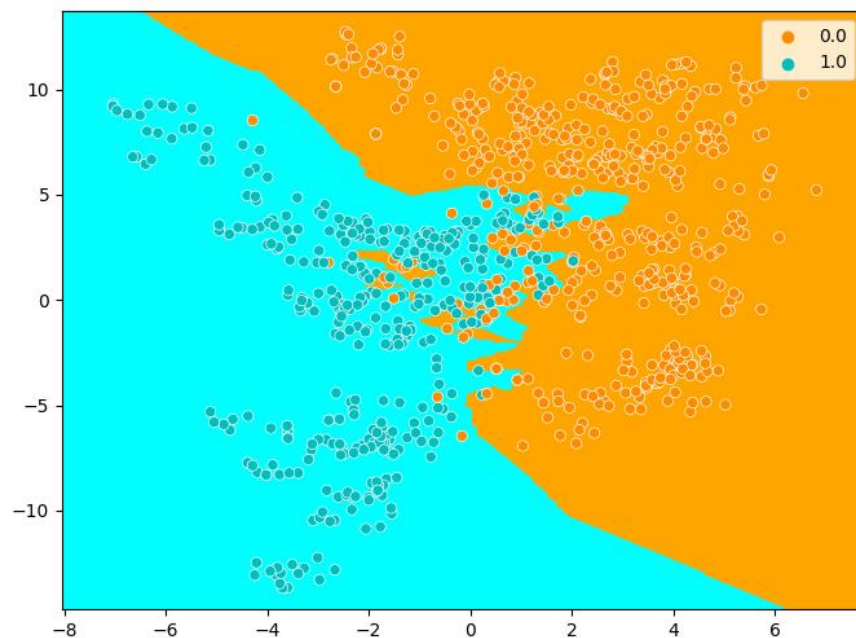


=> دو ویژگی یک و دو انتخاب می‌شود. چون میزان درهم رفتگی‌شان کمتر از بقیه حالات است.

قسمت (b)

در این قسمت برای رنگ کردن نواحی مختلف، تعداد نقاط زیادی از صفحه تولید شد و به عنوان داده‌ی تست، پیش‌بینی شد. از تابع `meshgrid` در این تمرین استفاده شد.

در این سوال پیاده‌سازی `KNN` انجام شده و برای بررسی درستی عملکرد از کتابخانه آماده نیز استفاده شده است. در شکل ۱ مرز تصمیم برای $k=1$ رسم شده است.

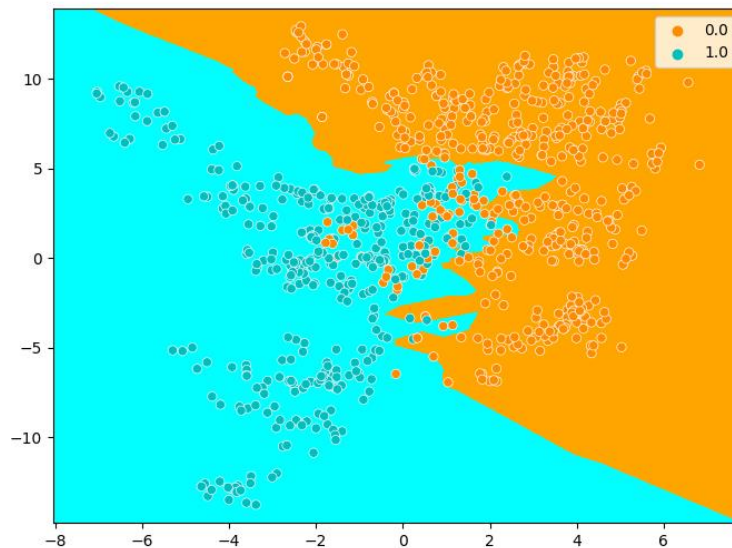


شکل ۱ مرز تصمیم برای $k=1$

قسمت C) در این حالت نیز همان دو ویژگی اول و دوم در نظر گرفته می‌شوند، چون داده‌ها در این حالت میزان درهم رفتگی و همپوشانی کمتری داشتند.

قسمت d)

شکل ۲ مرز تصمیم را برای حالت $k=3$ نشان می‌دهد.



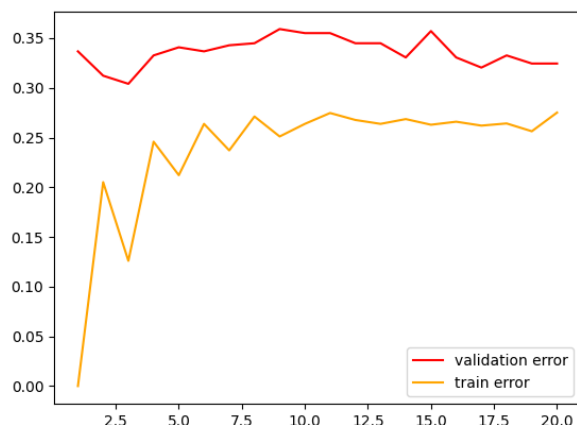
شکل ۲ مرز تصمیم برای $k=3$

سوال چهار

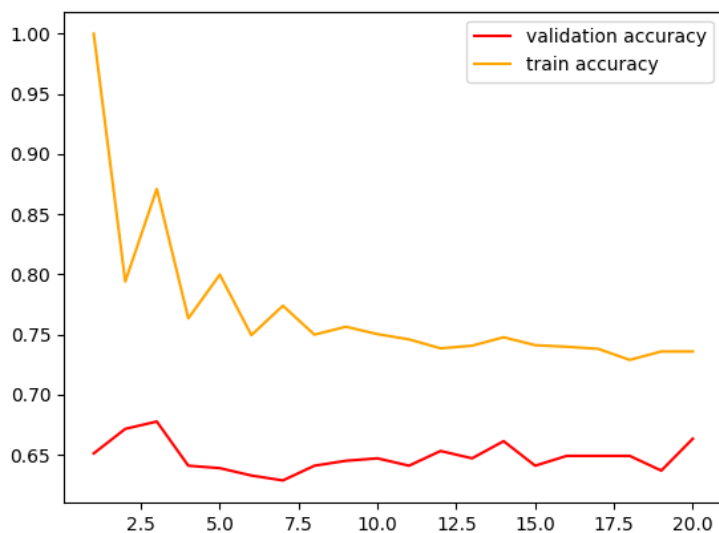
قسمت a,b)

موارد خواسته شده این دو قسمت، در کد پیاده‌سازی شده است.

به ازای k های ۱ تا ۲۰ خطای ارزیابی و آموزش در شکل ۳ رسم شده است.



شکل ۳ خطای ارزیابی و آموزش به ازای $k=1-20$



شکل ۴ دقت ارزیابی و آموزش به ازای $k=1-20$

با توجه به نمودار بالا بهترین k برابر با ۳ است. اگر دقت کنیم می‌بینیم که $k = 14, 17$ نیز دقت خوبی دارند. اما دقت در ۳ بهتر است.

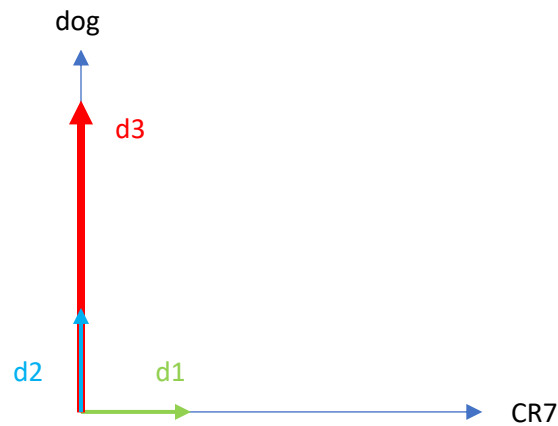
همانطور که انتظار می‌رود در $k=1$ خطای آموزش صفر و دقت ۱ است. چون اولین نزدیکترین همسایه به هر داده خود همان داده است و فاصله صفر می‌شود.

قسمت C)

خطای حاصل در این مدل از فاصله با $k=3$ برابر با 0.2408 است. در حالی که به ازای همین k در حالت قبل خطا حدود 0.3 بود.

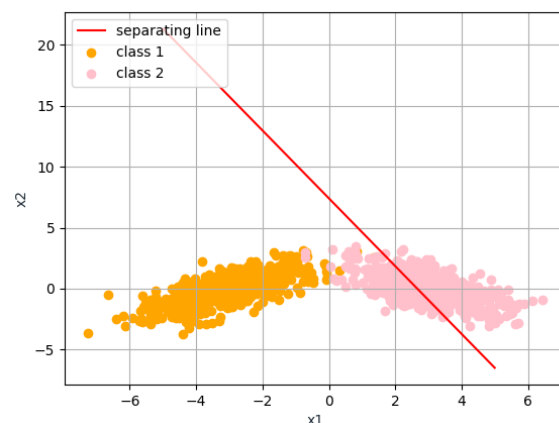
فاصله‌ی اقلیدسی تعداد کلمات را در محاسبه‌ی فاصله دخیل می‌کند در حالی که فاصله‌ی کسینوسی چون زاویه‌ی بین دو بردار را در نظر می‌گیرد تعداد تکرار کلمه و یا به طور کلی تر طول سند لحاظ نمی‌شود.

بردارهای متناظر با مثال داده شده در شکل زیر رسم شده اند. اگر از فاصله‌ی اقلیدسی استفاده کنیم شباهت سند یک و دو بیشتر است، درحالی که این نتیجه گیری غلط است. اما در فاصله‌ی کسینوسی چون زاویه‌ی بین دو سند مورد بررسی قرار می‌گیرد و زاویه با تعداد تکرار کلمات ارتباطی ندارد، این معیار بهتر عمل میکند.



سوال پنجم

قسمت d)

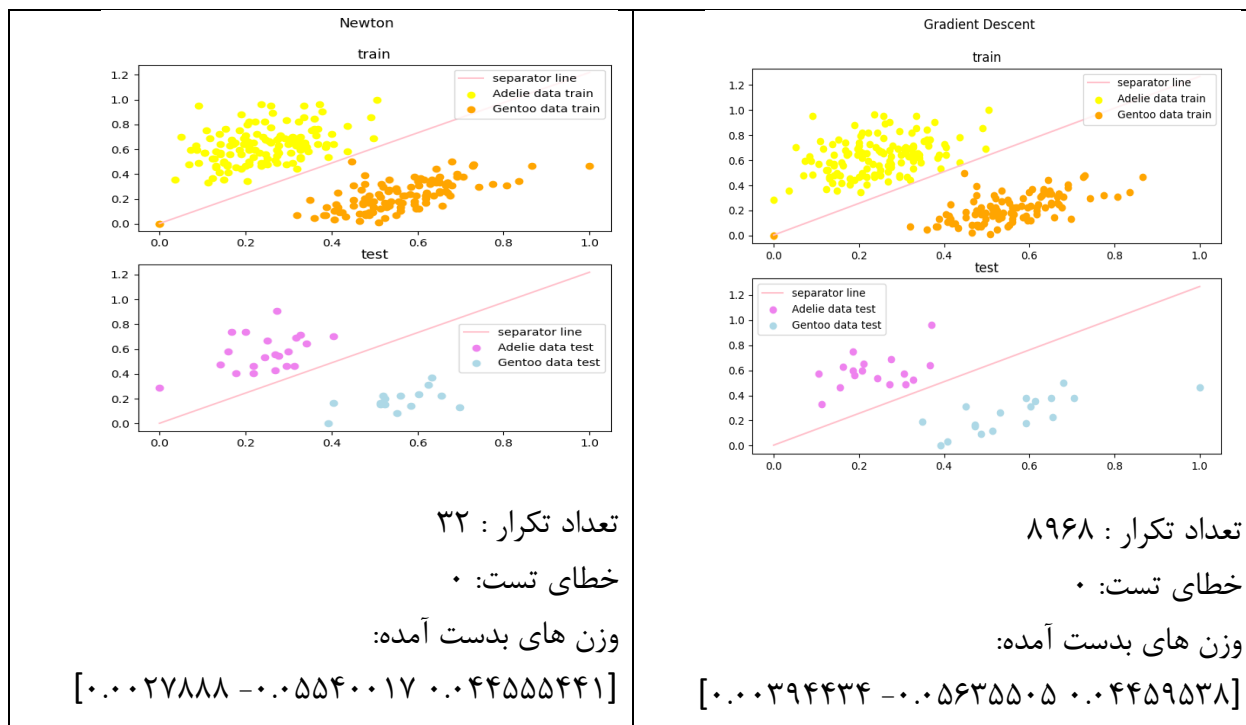


دیگر قسمت‌ها در فایل دستی نوشته شده است.

سوال ششم

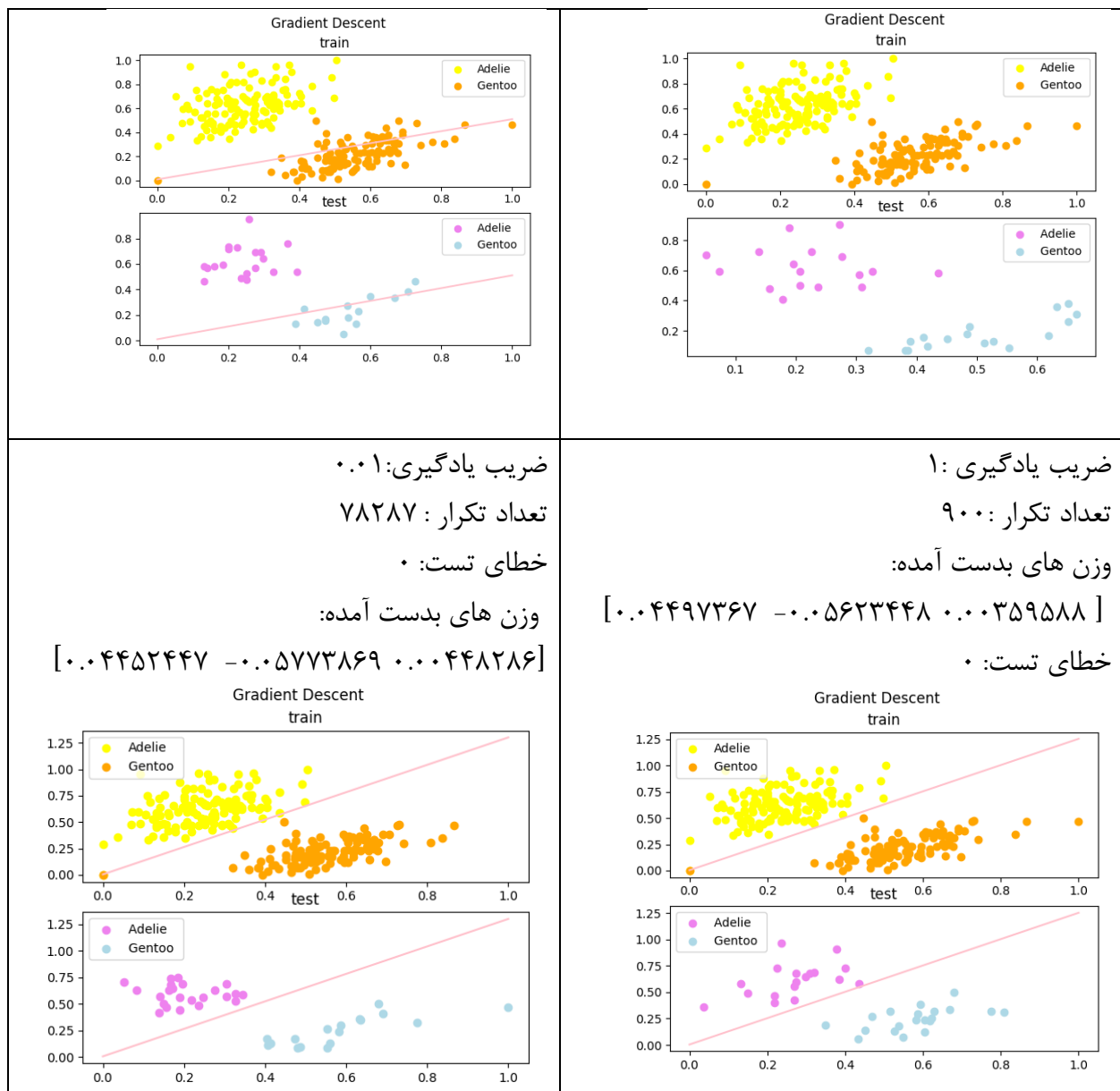
قسمت (a)

شرط اجرای حلقه این است که به ازای α های جدید مقدار تابع هدف بهتر از حالت قبل باشد.



قسمت (b)

<p>ضریب یادگیری : ۵</p> <p>تعداد تکرار : ۱</p> <p>وزن های بدست آمده:</p> <p>[۰.۰۲۳۹۲۲۲ -۰.۰۱۱۹۸۹۵۲ ۰.۰۰۹۴۶۵۰۲]</p> <p>خطای تست: ۰.۳۷</p>	<p>ضریب یادگیری: ۱۰</p> <p>تعداد تکرار : ۰</p> <p>خطای تست: ۰.۵</p> <p>وزن های بدست آمده:</p> <p>[۰ ۰ ۰]</p>
--	--



همانطور که مشاهده میکنیم با انتخاب نرخ یادگیری کوچک، تعداد تکرار حلقه افزایش می یابد. و با انتخاب های بزرگ حلقه اصلا اجرا نمیشود (چرا؟) چون شرط تکرار حلقه این است که مقدار تابع هدف به ازای a های جدید بهتر از حالت قبل باشد. در حالتی که نرخ یادگیری بزرگ باشد، تابع هدف مقداری بزرگتر از حالت صفر میگیرد، در نتیجه حلقه اصلا اجرا نمی شود.

قسمت C)

in this problem : $n = 344$, $d = 3$

Gradient Descent

$$\mathbf{a}_{k+1} = \mathbf{a}_k - \eta \times \mathbf{Y}^t (\mathbf{Y}\mathbf{a}_{(k)} - \mathbf{b})$$

$$\mathbf{Y}_{n \times d+1} \quad \mathbf{a}_{d+1 \times 1} \quad \mathbf{b}_{n \times 1}$$

of operations for expression $\mathbf{Y}\mathbf{a} - \mathbf{b}$: $n(2d+2)$

$$(\mathbf{Y}\mathbf{a} - \mathbf{b})_{n \times 1}$$

of operations for expression $\mathbf{Y}^t(\mathbf{Y}\mathbf{a} - \mathbf{b})$: $(d+1)(2n-1)$

of operations for expression $\eta \times \mathbf{Y}^t (\mathbf{Y}\mathbf{a} - \mathbf{b})$: $d+1$

of operations for expression $\mathbf{a} - \eta \times \mathbf{Y}^t (\mathbf{Y}\mathbf{a} - \mathbf{b})$: $d+1$

Total operations in one iteration : $2nd + 2n + 2nd + 2n - d - 1 + d + 1 + d + 1 =$
 $4nd + 4n + d + 1$

If we have m iterations then :

operations order is : $O(mnd)$

Newton

$$\mathbf{a}_{k+1} = \mathbf{a}_k - \mathbf{H}^{-1} \mathbf{Y}^t (\mathbf{Y}\mathbf{a}_k - \mathbf{b})$$

\mathbf{H} is Hessian matrix which is $d+1$ by $d+1$

Inverse of \mathbf{H} : $(d+1)^3 + (d+1)^2 - 3(d+1) + 2$

of operations for expression $\mathbf{H}^{-1} \mathbf{Y}^t (\mathbf{Y}\mathbf{a} - \mathbf{b})$: $(d+1)(2d+1)$

of operations for expression $\mathbf{a} - \mathbf{H}^{-1} \mathbf{Y}^t (\mathbf{Y}\mathbf{a} - \mathbf{b})$: $d+1$

total : $2nd + 2n + 2nd + 2n - d - 1 + 2d^2 + 3d + 1 + d + 1 + (d+1)^3 + d^2 + 2d + 1 - 3d$
 $-3 + 2 =$

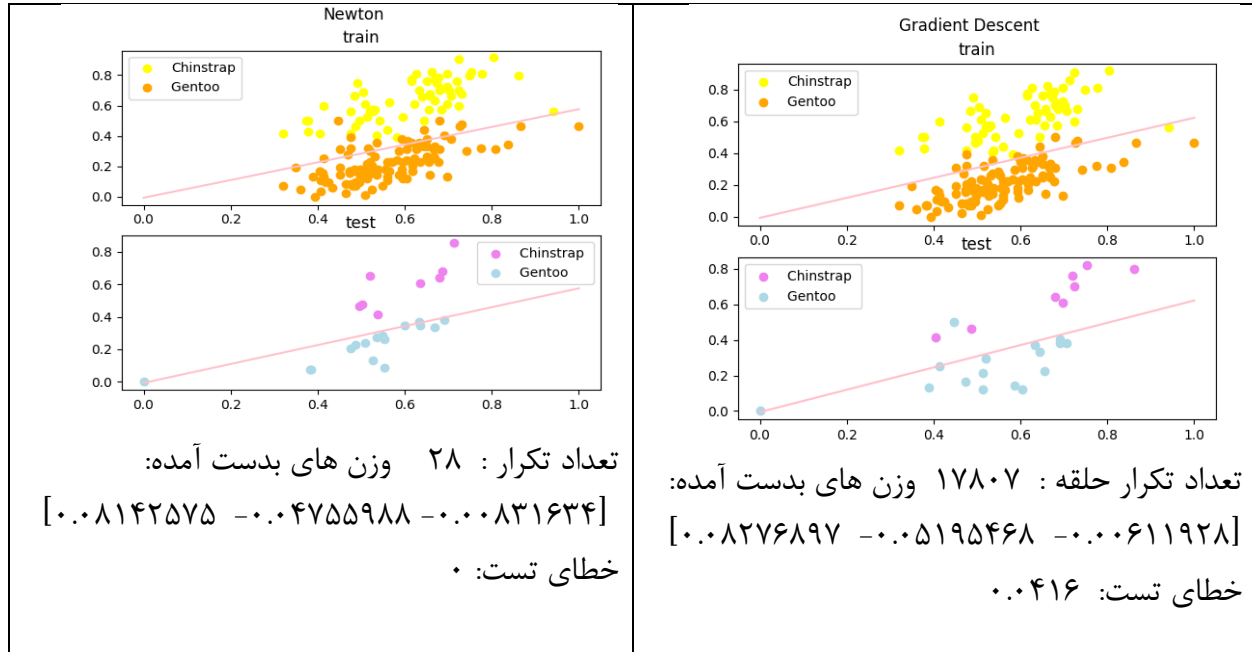
$$(d+1)^3 + 3d^2 + 4n(d+1) + 2d + 1$$

If we have m iterations then :

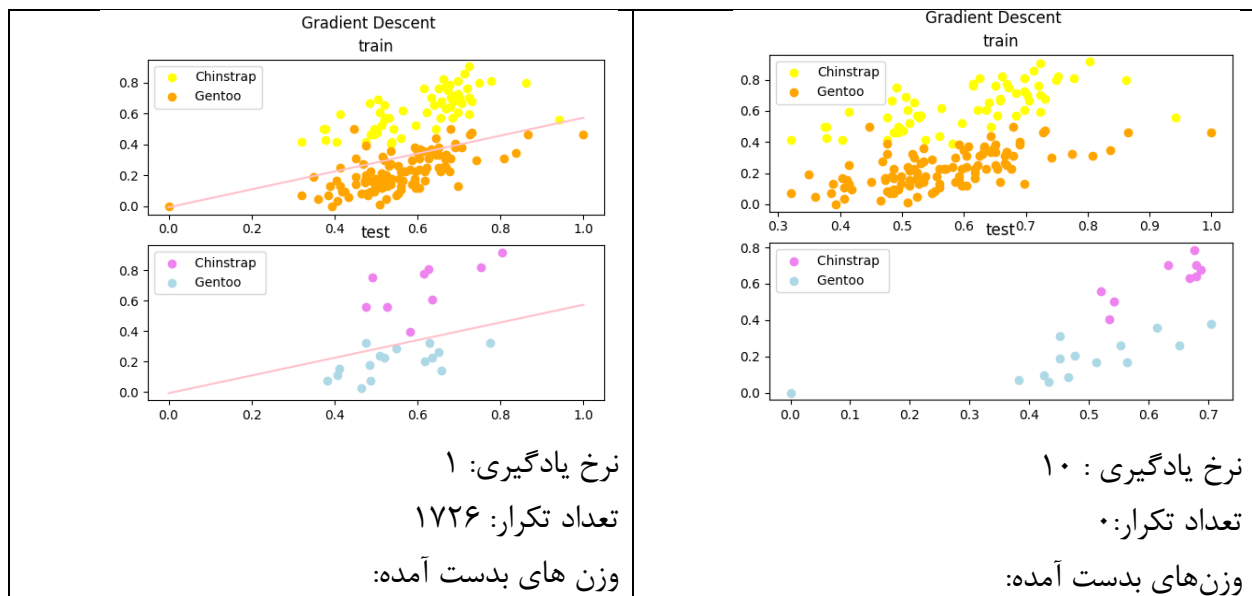
operations order is : $O(mnd + md^3)$

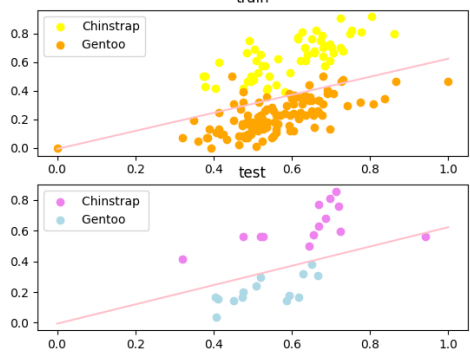
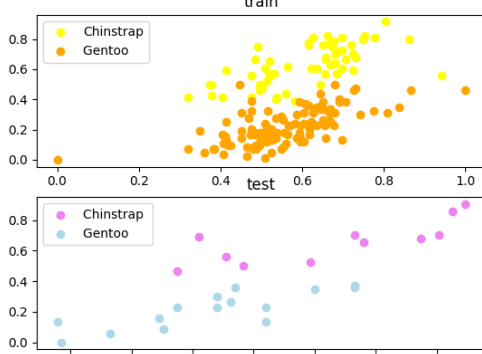
قسمت (d)

بخش اول:



بخش دوم:

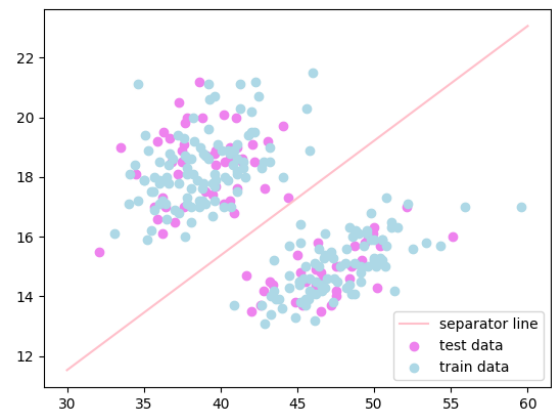


<p>[۰.۰۸۴۶۳۳۴۲ - ۰.۰۴۹۱۶۹۰۴ - ۰.۰۰۸۶۲۹۰۴]</p> <p>خطای تست: ۰.۰۴</p>	<p>[۰ ۰ ۰]</p> <p>خطای تست: ۰.۶</p>
<p>Gradient Descent train</p>  <p>نرخ یادگیری: ۰.۰۱ تعداد تکرار: ۱۴۷۷۴۱ وزن‌های بدست آمده: [۰.۰۸۱۷۷۹۶۷ - ۰.۰۵۱۴۸۶۴۳ - ۰.۰۰۶۷۱۰۰۴] خطای تست: ۰.۰۳</p>	<p>Gradient Descent train</p>  <p>نرخ یادگیری: ۵ تعداد تکرار: ۰ وزن‌های بدست آمده: [۰ ۰ ۰] خطای تست: ۰.۷۳</p>

همانطور که مشاهده میکنیم با انتخاب نرخ یادگیری کوچک، تعداد تکرار حلقه افزایش می‌یابد. و با انتخاب‌های بزرگ حلقه اصلاً اجرا نمیشود (چرا؟) چون شرط تکرار حلقه این است که مقدار تابع هدف به ازای a های جدید بهتر از حالت قبل باشد. در حالتی که نرخ یادگیری بزرگ باشد، تابع هدف مقداری بزرگتر از حالت صفر میگیرد، در نتیجه حلقه اصلاً اجرا نمیشود.

قسمت e)

نحوه ی توزیع دو کلاس کاملاً مشخص است. این دو کلاس خطی جدا پذیر هستند.

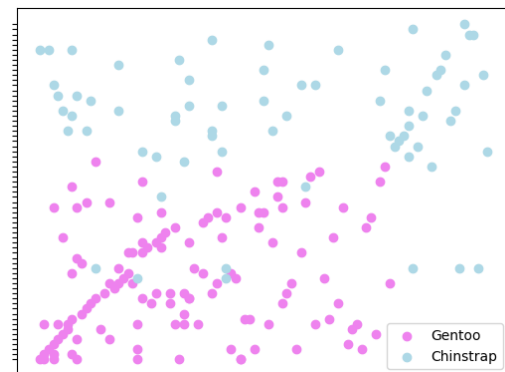


تعداد تکرار ۷۷ ، میزان خطای تست: ۰.۰۱۲

قسمت (f,g)

هیچگاه برنامه پایان نمی‌یافت.

برای درک اینکه چه اتفاقی در کد می‌افتد توزیع داده‌ها را رسم کردم که در شکل زیر قابل مشاهده است.



همانطور که می‌بینیم داده‌های این دو توزیع به صورت خطی جداپذیر نیست بنابراین این برنامه تا بی‌نهایت ادامه می‌یابد و هیچوقت همگرا نمی‌شود.

سوال هفت

قسمت (a) بله

در کلاس سه روش مطرح شد که به شرح زیر است:

روش اول

$$h_{opt} = 1.06\sigma N^{-1/5}$$

یعنی ۱.۰۶ * انحراف معیار داده ها * تعداد نمونه ها به توان ۱/۵ -

روش دوم

$$h_{opt} = 0.9AN^{-1/5} \quad \text{where } A = \min\left(\sigma, \frac{IQR}{1.34}\right)$$

که IQR همان رنج داده‌ها در چارک دوم و سوم است.

روش سوم

$$h_{opt} = \sqrt[2]{n}$$

قسمت (b) بله

در الگوریتم کمترین فاصله، اگر از فاصله‌ی اقلیدسی استفاده کنیم در واقع داریم فاصله‌ی تست را از نماینده‌ی یک توزیع محاسبه می‌کنیم و اگر از فاصله‌ی مانهالانوبیس استفاده کنیم داریم فاصله‌ی تست را از توزیع یک کلاس محاسبه می‌کنیم و سپس تصمیم‌گیری را انجام می‌دهیم. در هر دو حالت سعی می‌شود که فاصله با کل داده‌های یک کلاس در نظر گرفته شود (نماینده‌ی یک کلاس یا کل توزیع یک کلاس).

در الگوریتم k نزدیکترین همسایه فاصله با تک تک داده‌های آموزش محاسبه می‌شود و سپس بین k نزدیکترین همسایه‌ها یک رای‌گیری انجام می‌شود و برچسب داده‌ی تست مشخص می‌شود. در این الگوریتم نیز میتوان از فاصله‌ی اقلیدسی استفاده کرد.

باوجود اینکه هردو الگوریتم از معیار فاصله برای تصمیم‌گیری استفاده می‌کنند ولی عملکرد و خروجی‌شان متفاوت است.

الگوریتم knn یک نسخه از الگوریتم‌های کمترین فاصله هست بدین صورت که فاصله با توزیع در نظر گرفته نمی‌شود و بجای آن فاصله با تمام داده‌های آموزش محاسبه می‌شود و سپس تصمیم‌گیری در یک ناحیه اطراف داده‌ی تست انجام می‌شود.

قسمت (c) خیر.

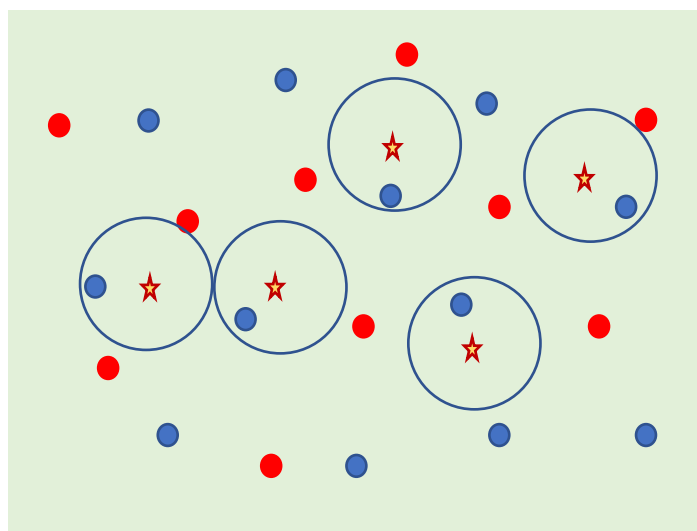
روش‌هایی خطی هستند که مرز تصمیم بتواند یک ابرصفحه باشد. در الگوریتم knn اگر k کوچک باشد مرز تصمیم یک ابرصفحه نیست و بیشتر شبیه یک پیچ هست. در k های بزرگ نیز مرز تصمیم یک پیچ هموار است ولی ابر صفحه نیست و انحنا دارد.

قسمت (d) بله.

هرچه k بزرگ‌تر باشد مرز تصمیم هموارتر است و هرچه k کوچک‌تر باشد این مرز دارای شکستگی‌های بیشتری است.

قسمت (e) اگر ویژگی‌های مناسبی انتخاب شده باشند و کلاس‌ها از هم مجزا باشند، خیر ممکن نیست. چون در هر صورت هر داده‌ای تست به توزیع خودش نزدیک تر است تا توزیع کلاس دیگر. مگر اینکه تمام داده‌های تست نویز و داده‌ی پرت باشند. پس در حالت معمول این اتفاق ممکن نیست. اما اگر ویژگی‌های مناسب انتخاب نشده باشند و توزیع داده‌ها از هم مجزا نباشد؛ یعنی کاملاً در هم رفته باشند ممکن است این حالت پیش بیاید.

در تصویر زیر توزیع داده‌های آموزش در دو کلاس با رنگ قرمز و آبی و داده‌های تست با شکل ستاره مشخص شده‌اند.



همانطور که مشاهده می‌کنیم در این حالت (که احتمال رخداد آن خیلی کم است) تمام داده‌های تست، یک برچسب می‌گیرند (اینجا آبی).

قسمت f) به داده‌های پرت و نویز حساس است.

حل: قبل از استفاده از این الگوریتم داده‌ها را پیش پردازش کنیم و با استفاده از روش‌های شناسایی نویز و داده‌ی پرت، این داده‌ها را حذف کنیم که عملکرد الگوریتم در فاز تست خوب باشد.

قسمت g,h) فکر می‌کنم این دو مورد آخر را استاد درس نداده‌اند.