Amirkabir University of Technology
(Tehran Polytechnic)
Department of Computer Engineering

Statistical Pattern Recognition

Assignment I

Soroush Mahdi
99131050

supervised by
Dr. Mohammad Rahmati

December 6, 2020

Contents

1 Designing Simple Pattern Analysis Systems

1.1 Predicting your final grade in this course

- This is a regression problem. Because we have labeled data, and we want to predict real valued labels.
- No sensors are needed for this problem.
- My training set is data from the previous 10 semesters that for each sample it contains the final score (our target), average study time per week, number of assignments that the student has done, the score for each assignment, number of sessions that the student has been in them, does the student take notes from class (yes or no), does the student has solved any sample questions for the exam? (yes or no) , the score of midterm, the score of the final exam, the score for each quiz.
- I will gather data from university information, the professor of the course records and I will ask questions from students(paper-based surveys).
- Selected features: average study time per week, the score for each assignment, does the student take notes from the class (yes or no), does the student has solved any sample questions for the exam? (yes or no) , the score of midterm, the score of the final exam, the score for each quiz.
- contacting all data together, handling the missing values, Encoding the categorical data, Splitting the dataset to train and test sets, Feature scaling
- the paper-based survey information may not be accurate.
- machines are faster and more accurate than humans. but a human observer is more flexible to changes than our system.

1.2 Predicting US Dollar to Iranian Rial exchange rate in the coming year

- This is a regression problem.
- I need a script that checks each day if there are any new tweets related to our work from specified accounts.
- My training set is data from the previous 5 years that for each sample it contains the exchange rate for each day (our target), oil sales per day, liquidity(cash flow) in the country, the debt of the government to the central bank and tweets from related persons to this field.
- I will gather data with scraping the related web pages like Currency Exchange web pages and twitter.
- Selected features: oil sales per day, liquidity(cash flow) in the country, the debt of the government to the central bank , related tweets.
- i have to clean the data that i had scraped and handle missing values. and i have to check the tweets with NLP methods to find useful information.
- It is a Complicated context so it is hard to select right features. the tweet feature is hard to Interpret.
- human expert may consider better features than us and has better ability to consider news and interpreting them.

1.3 Grouping students in a dorm by their personalities

- This is a classification problem.
- I need cameras in the dorm for detecting groups based on their friendship.
- My training set is this information for each person in the dorm: personality, age, sex, in the range 0-10 how much is he emotional and reasonable.

- I will gather data from cameras and paper-based surveys.
- Selected features: age, sex, group of friends , score of Emotional stability, score of reasoning.
- scaling the 2 scores , processing data from cameras and interpreting them.
- I need to use computer vision technics for interpreting camera data. the information of paper-based surveys may not be Accurate and emotional ability and reasoning are not measurable features so different humans may have different scales in their minds for these features and it is hard to scale them.
- this is a hard problem in its domain and needs expert knowledge for selecting features. but if it works well its so much faster then the human expert.

2 Getting More Familiar with the Art of Feature Extraction

2.1 a

- hair color
- skin color

2.2 b

- are teeth appearing?

2.3 c

- is the hair color white?
- Size of the face

2.4 d

- long hair or short hair?
- skin texture
- right eye corner and left eye corner
- nose center point

2.5 e

- hair color
- skin color
- lips center point
- right eye corner and left eye corner
- nose center point

3 Basic Statistics Warm-up

3.1 A - F

3.a

$$\int_{-\infty}^{+\infty} f_X(x)\,dx = 1 \Rightarrow \int_0^1 cx\,dx = 1 \Rightarrow \Big|_0^1 \frac{cx^2}{2} = 1 \Rightarrow c = 2$$

3.b

$$P(0 \leq X \leq 0.5) = \int_0^{0.5} 2x\,dx = 0,25$$

3.c

$$E[X] = \int_{-\infty}^{+\infty} x f_X(x)\,dx = \int_0^1 x.2x\,dx = \int_0^1 2x^2\,dx = \Big|_0^1 \frac{2x^3}{3} = \frac{2}{3}$$

3.d

$$Var(X) = E\left[(X-E(X))^2\right] = \int_{-\infty}^{+\infty} (x-E(X))^2 f_X(x)\,dx = \int_0^1 \left(x-\frac{2}{3}\right)^2 2x\,dx = \int_0^1 2x\left(x^2+\frac{4}{9}-\frac{4}{3}x\right)dx$$

$$= \int_0^1 2x^3+\frac{8}{9}x-\frac{8}{3}x^2 = \Big|_0^1 2\frac{x^4}{4}+\frac{8}{9}\frac{x^2}{2}-\frac{8}{3}\frac{x^3}{3} = \frac{1}{2}+\frac{4}{9}-\frac{8}{9} = \frac{1}{18}$$

3.e

$$\star\ E(aX+b) = aE(X)+b \Rightarrow E[2X-2] = 2E(X)-2 = \frac{4}{3}-2 = -\frac{2}{3}$$

3.f

$$\star\ Var(aX+b) = a^2\,Var(X) \Rightarrow Var[2X-2] = 4\,Var[X] = \frac{4}{18}$$

Figure 1: a-f

3.2 G - K

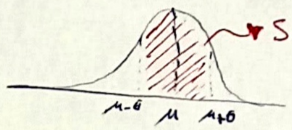
3.g ★ $Z \sim N(0,1)$

$$X = \sigma Z + \mu \Rightarrow X = 3Z+1 \Rightarrow P[-2 \leq X \leq 1] = P[-2 \leq 3Z+1 \leq 1] = P[-1 \leq Z \leq 0]$$

$$P[-1 \leq Z \leq 0] = \Phi(0) - \Phi(-1) = 0.5 - 0.16 = 0.34 = 34\%$$

$$\star \; -2 \leq X \leq 1 \equiv \mu - \sigma \leq X \leq \mu$$

↳



$S = 0.68 \Rightarrow \frac{S}{2} = 0.34$

3.h

$$X = \sigma Z + \mu = 3Z+1 \Rightarrow E[X] = E[3Z+1] = 3\overbrace{E[Z]}^{0} + 1 = 1 = \mu$$

$$Var[X] = Var[3Z+1] = 3^2 \overbrace{Var(Z)}^{1} = 9 = \sigma^2$$

3.i

$$X = 3Z+1 \Rightarrow Y = 2X-1 = 2(3Z+1)-1 = 6Z+2-1 = 6Z+1 \Rightarrow Y \sim N(1,36)$$

Y is a continuous random variable that has normal distribution with $\mu = 1$ & $\sigma^2 = 36$

3.j

$$\binom{7}{4}(0.8)^4 \cdot \binom{3}{3}(0.2)^3$$

3.k

$$\int_{-\infty}^{m} f(x)\,dx = \frac{1}{2} \Rightarrow \int_0^m 1 - \frac{x^2}{4}\,dx = \frac{1}{2} \Rightarrow \Big|_0^m x - \frac{x^3}{12} = m - \frac{m^3}{12} = \frac{1}{2} \Rightarrow 12m - m^3 = 6$$

$$\Rightarrow m \simeq 0.52 \quad (0 \leq m \leq 2)$$

Figure 2: g-k

3.3 L - M

3.m

$$E(X) = \int_{-\infty}^{+\infty} x f_X(x)\, dx = \int_0^1 x \frac{4}{\pi(1+x^2)}\, dx = \frac{4}{\pi} \int_0^1 \frac{x}{1+x^2}\, dx \quad , u = x^2+1 \Rightarrow \frac{du}{dx} = 2x \Rightarrow dx = \frac{du}{2x}$$

$$\Longrightarrow E(X) = \frac{4}{\pi} \int_0^1 \frac{x}{u} \frac{du}{2x} = \frac{4}{\pi} \cdot \frac{1}{2} \int_0^1 \frac{1}{u}\, du = \left. \frac{2}{\pi} \ln(u) \right|_0^1 = \left. \frac{2}{\pi} \ln(x^2+1) \right|_0^1$$

$$= \frac{2}{\pi} \left[\ln(2) - \ln(1) \right] = \frac{2}{\pi} \ln(2)$$

3.l

$$X \sim N(10, 0.01) \Longrightarrow X = 0.1 Z + 10 \Longrightarrow P[0.1 Z + 10 < 9.8]$$

$$= P[Z + 100 < 98] = P[Z < -2] = \Phi(-2) = 0.0228$$

Figure 3: l-m

4 Mastering Eigenvalues and Eigenvectors and Their Properties

4.1 A - C

4.a

a1. $Av = \begin{bmatrix}1 & 4\\ 3 & 2\end{bmatrix}\begin{bmatrix}-3\\ -3\end{bmatrix} = \begin{bmatrix}-15\\ -15\end{bmatrix} \checkmark \longrightarrow \underset{\text{eigen value}}{\underline{5}}\begin{bmatrix}-3\\ -3\end{bmatrix} = \begin{bmatrix}-15\\ -15\end{bmatrix}$

$\underline{v}$ is an eigenvector of [A] and the corresponding eigenvalue is 5.

a2. $Av = \begin{bmatrix}5 & 6 & 2\\ 0 & -1 & -8\\ 1 & 0 & -2\end{bmatrix}\begin{bmatrix}1\\ 2\\ 1\end{bmatrix} = \begin{bmatrix}5+12+2\\ 0-2-8\\ 1+0-2\end{bmatrix} = \begin{bmatrix}19\\ -10\\ -1\end{bmatrix} \times$

$\underline{v}$ is not an eigenvector of [A].

4.b

b.1 $(A-\lambda I)X=0 \Rightarrow \begin{bmatrix}3 & -6\\ -3 & 6\end{bmatrix}X=0 \xRightarrow{\text{Add the top row to the bottom}} \begin{bmatrix}3 & -6\\ 0 & 0\end{bmatrix}X=0 \Rightarrow 3x_1-6x_2=0 \Rightarrow x_1=2x_2$, $X = \begin{bmatrix}x_1\\ x_2\end{bmatrix}$

$\Longrightarrow X = \begin{bmatrix}2\\ 1\end{bmatrix}$, The eigenspace corresponding to $\lambda=-2$ is the span of $\begin{bmatrix}2\\ 1\end{bmatrix}$

b.2 $(A-\lambda I)X=0 \Rightarrow \begin{bmatrix}-8 & 4 & 2\\ 2 & -5 & -2\\ 4 & -2 & -1\end{bmatrix}X=0$

1 Add 2 bottom row to the top row $\longrightarrow \begin{bmatrix}0 & 0 & 0\\ 2 & -5 & -2\\ 4 & -2 & -1\end{bmatrix}$

Figure 4: a-b

4.b.2

$$\begin{bmatrix} 0 & 0 & 0 \\ 2 & -5 & -2 \\ 4 & -2 & -1 \end{bmatrix} \xrightarrow{\text{subtract 2 times of second row from the bottom}} \begin{bmatrix} 0 & 0 & 0 \\ 2 & -5 & -2 \\ 0 & 8 & 3 \end{bmatrix}, X = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

$$\Rightarrow \left\{ \begin{array}{l} 2x_1 - 5x_2 - 2x_3 = 0 \\ 8x_2 + 3x_3 = 0 \Rightarrow x_3 = -\frac{8}{3}x_2 \end{array} \right\} \Rightarrow 2x_1 - 5x_2 + \frac{16}{3}x_2 = 0 \Rightarrow x_1 = -\frac{1}{6}x_2$$

$$\left\{ \begin{array}{l} x_1 = -\frac{t}{6} \\ x_2 = t \\ x_3 = -\frac{8}{3}t \end{array} \right. \Rightarrow X = \begin{bmatrix} -1 \\ 6 \\ -16 \end{bmatrix}$$

The eigenspace corresponding to $\lambda = 6$ is the span of $\begin{bmatrix} -1 \\ 6 \\ -16 \end{bmatrix}$

4.c

C.1 $|A - \lambda I| = 0 \Rightarrow \begin{vmatrix} 3-\lambda & 2 \\ -1 & 6-\lambda \end{vmatrix} = 0 \Rightarrow (3-\lambda)(6-\lambda) - (-2) = 0$

$$\Rightarrow 18 - 9\lambda + \lambda^2 + 2 = 0 \Rightarrow \underbrace{\lambda^2 - 9\lambda + 20}_{(\lambda-4)(\lambda-5)} = 0 \Rightarrow \lambda = \left\{ \begin{array}{l} 4 \\ 5 \end{array} \right.$$

$\lambda = 4$

$$\Rightarrow A - \lambda I = \begin{bmatrix} -1 & 2 \\ -1 & 2 \end{bmatrix} \equiv \begin{bmatrix} -1 & 2 \\ 0 & 0 \end{bmatrix} \xrightarrow{2x_2 = x_1} \text{eigenvectors} \begin{bmatrix} 2 \\ 1 \end{bmatrix}$$

Figure 5: b-c

4.C.1

$$\lambda = 5 \Rightarrow A - \lambda I = \begin{bmatrix} -2 & 2 \\ -1 & 1 \end{bmatrix} \equiv \begin{bmatrix} 0 & 0 \\ -1 & 1 \end{bmatrix} \xRightarrow{x_2 = x_1} \text{eigenvectors} \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

C.2

Zero matrix has only zero as it's eigenvalues.

C.3

$$|A - \lambda I| = 0 \Rightarrow \begin{vmatrix} 3-\lambda & 1 & -2 \\ 2 & 3-\lambda & -2 \\ 2 & 1 & -1-\lambda \end{vmatrix} = 0$$

$$\Rightarrow (3-\lambda)\Big[\overbrace{(3-\lambda)(-1-\lambda) - (-2)(1)}^{\lambda^2 - 3 - 2\lambda + 2}\Big] - \Big[\overbrace{2(-1-\lambda) - (-2)(2)}^{-2-2\lambda+4}\Big] +$$

$$(-2)\Big[\underbrace{(2)(1) - (3-\lambda)(2)}_{2-6+2\lambda}\Big] = -\lambda^3 + 5\lambda^2 - 11\lambda + 7 = 0$$

$$\Rightarrow (1-\lambda)(\lambda^2 + 7 - 4\lambda) = 0 \Rightarrow \lambda = \begin{cases} 1 \\ 2 + i\sqrt{3} \\ 2 - i\sqrt{3} \end{cases}$$

$$\xRightarrow{\lambda = 1} A - \lambda I = \begin{bmatrix} 2 & 1 & -2 \\ 2 & 2 & -2 \\ 2 & 1 & -2 \end{bmatrix} \equiv \begin{bmatrix} 0 & 0 & 0 \\ 2 & 2 & -2 \\ 2 & 1 & -2 \end{bmatrix} \equiv \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 2 & 1 & -2 \end{bmatrix}$$

$$\Rightarrow \left\{ \begin{array}{l} x_2 = 0 \\ 2x_1 + x_2 - 2x_3 = 0 \Rightarrow x_1 = x_3 \end{array} \right\} \Rightarrow \text{eigenvector} = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}$$

Figure 6: c

4.2 D - E

4.d

$$A = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \qquad A v_1 = \lambda_1 v_1 \Rightarrow \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} -3 \\ 2 \end{bmatrix} = \begin{bmatrix} 3 \\ -2 \end{bmatrix} \Rightarrow \begin{cases} -3a + 2b = 3 \\ -3c + 2d = -2 \end{cases}$$

$$A v_2 = \lambda_2 v_2 \Rightarrow \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} 1 \\ 2 \end{bmatrix} = \begin{bmatrix} 7 \\ 14 \end{bmatrix} \Rightarrow \begin{cases} a + 2b = 7 \\ c + 2d = 14 \end{cases}$$

$$\Rightarrow a = 1,\ b = 3,\ c = 4,\ d = 5$$

$$\Rightarrow A = \begin{bmatrix} 1 & 3 \\ 4 & 5 \end{bmatrix}$$

4.e

$$Ax = \lambda x \Rightarrow \underbrace{A^{-1}A}_{I} x = \lambda A^{-1} x \Rightarrow I x = \lambda A^{-1} x \Rightarrow \frac{1}{\lambda} x = A^{-1} x \Rightarrow A^{-1} x = \lambda^{-1} x$$

Figure 7: d-e

4.3 F

Suppose A is an 8*8 matrix If A is diagonalizable, then the sum of the dimensions of the eigenspaces is n and vice versa.the sum of the dimensions of the eigenspaces is 2 so A is not diagonalizable.

4.4 G

4.g

$$Ax = 3x \Rightarrow \underbrace{A\,A}_{A^2} x = 3\underbrace{(Ax)}_{3x} \Rightarrow A^2 x = 9x$$

Figure 8: g

4.5 H

4.h

let $k_n = \begin{bmatrix} x_n \\ y_n \end{bmatrix} \Rightarrow k_{n+1} = \begin{bmatrix} 2x_n - 3y_n \\ -4x_n + y_n \end{bmatrix} = \begin{bmatrix} 2 & -3 \\ -4 & 1 \end{bmatrix} k_n \Rightarrow k_{n+1} = \begin{bmatrix} 2 & -3 \\ -4 & 1 \end{bmatrix}^{n+1} k_0$

, $k_0 = \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} = \begin{bmatrix} 2 \\ 3 \end{bmatrix}$

★ if A is diagonizable $\rightarrow S^{-1}AS = \Lambda \Rightarrow A = S\Lambda S^{-1} \Rightarrow A^n = (S\Lambda \overbrace{S^{-1})(S}^{I}\Lambda S^{-1}) \cdots (S\Lambda S^{-1})$

$\Rightarrow A^n = S\Lambda^n S^{-1}$

finding eigen values of $\begin{bmatrix} 2 & -3 \\ -4 & 1 \end{bmatrix} \Rightarrow (2-\lambda)(1-\lambda) - 12 = 0 \Rightarrow \lambda = \begin{cases} 5 \\ -2 \end{cases}$

This matrix has two distincts eigenvalues so it's diagonizable

$\Rightarrow \Lambda = \begin{bmatrix} 5 & 0 \\ 0 & -2 \end{bmatrix}$, eigenvectors $\Rightarrow x_1 = \begin{bmatrix} 1 \\ -1 \end{bmatrix}, x_2 = \begin{bmatrix} 3 \\ 4 \end{bmatrix}$

$\Rightarrow S = \begin{bmatrix} 1 & 3 \\ -1 & 4 \end{bmatrix}$

$\Rightarrow S^{-1} = \frac{1}{4+3} \begin{bmatrix} 4 & -3 \\ 1 & 1 \end{bmatrix}$

$= \frac{1}{7} \begin{bmatrix} 4 & -3 \\ 1 & 1 \end{bmatrix}$

$$\Rightarrow \begin{bmatrix} 2 & -3 \\ -4 & 1 \end{bmatrix}^n = \begin{bmatrix} 1 & 3 \\ -1 & 4 \end{bmatrix} \begin{bmatrix} 5^n & 0 \\ 0 & (-2)^n \end{bmatrix} \begin{bmatrix} \frac{4}{7} & \frac{-3}{7} \\ \frac{1}{7} & \frac{1}{7} \end{bmatrix}$$

$$\Rightarrow k_n = \begin{bmatrix} 1 & 3 \\ -1 & 4 \end{bmatrix} \begin{bmatrix} 5^n & 0 \\ 0 & (-2)^n \end{bmatrix} \begin{bmatrix} \frac{4}{7} & \frac{-3}{7} \\ \frac{1}{7} & \frac{1}{7} \end{bmatrix} \underbrace{\begin{bmatrix} 2 \\ 3 \end{bmatrix}}_{k_0}$$

Figure 9: h

5 Simple Sample Generation and Beyond

5.1

```
mu = 5
sigma = [1,2,3]

for i in sigma:
    s = np.random.normal(mu, i, 500) #generate samples
    plt.title('sigma= '+ str(i))
    plt.plot(s, np.zeros_like(s), 'o') #plotting samples
    plt.savefig('samples' + str(i) + '.png')
    plt.clf()

    plt.hist(s, 30, density=True) # plotting histogram
    plt.title('sigma= '+ str(i))
    plt.savefig('hist' + str(i) + '.png')
    plt.clf()
```

Listing 1: 1-D normal samples

we see that because mu is equal for all of the samples in each of the three histograms the data near the five are more than other places but for higher sigmas scattering of data becomes more and we have further tails to the mu. but for smaller sigmas we have higher peaks around five.

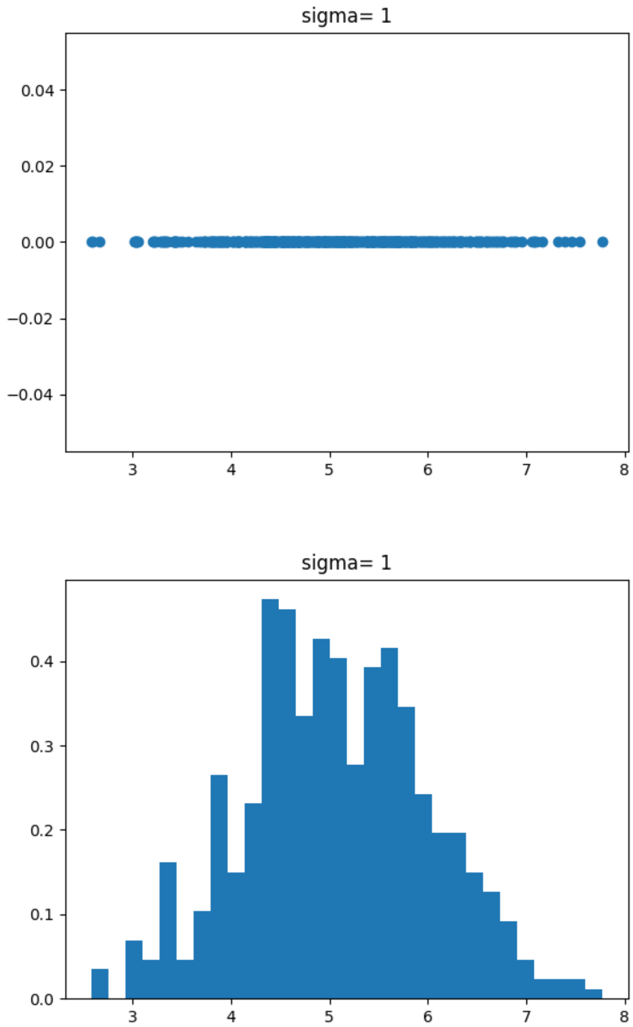


Figure 10: sigma = 1

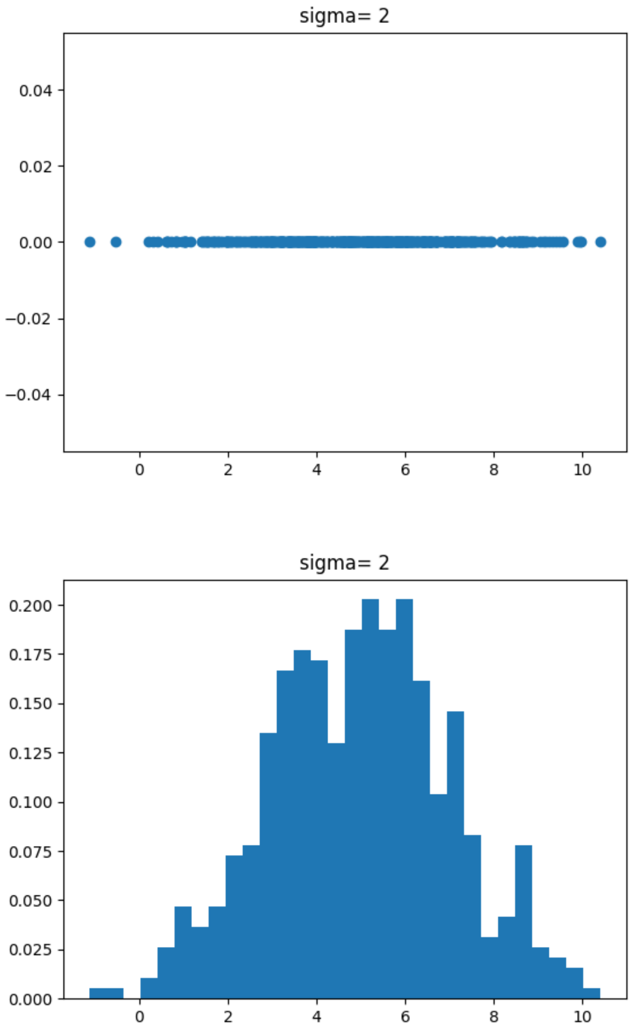


Figure 11: sigma = 2

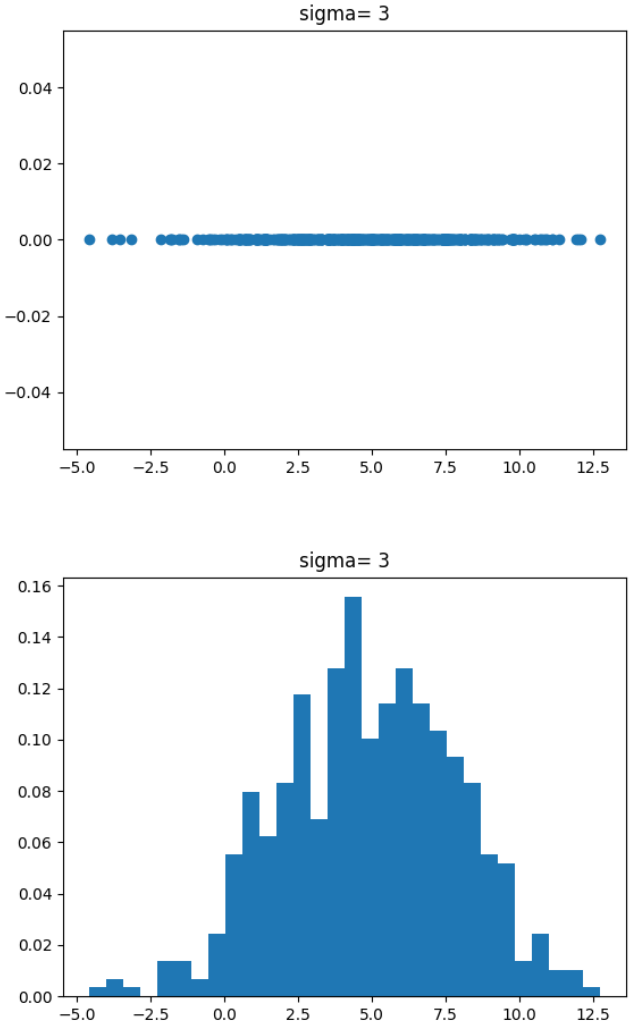
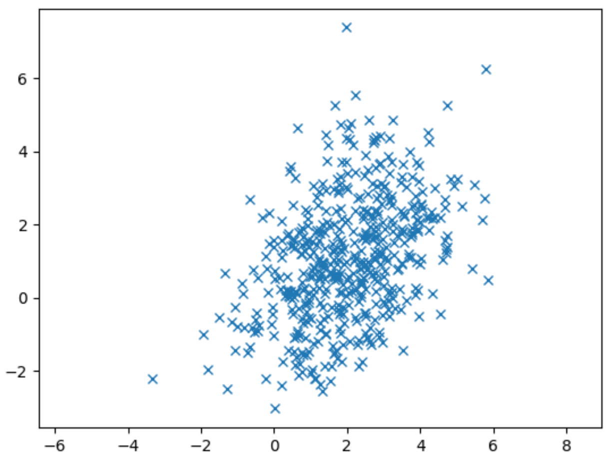


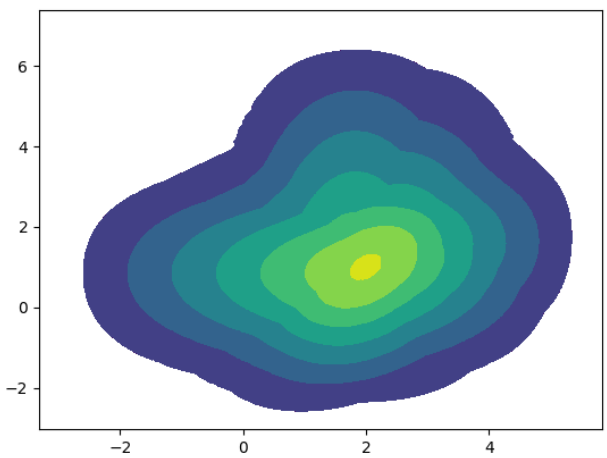
Figure 12: sigma = 3

5.2

```
from scipy.stats import multivariate_normal

mu = [2, 1]
cov = [[2, 1], [1, 3]]
#generating samples
x,y = np.random.multivariate_normal(mu, cov, 500).T
#plotting samples
plt.plot(x,y, 'x')
plt.axis('equal')
plt.savefig('5b1.png')

plt.clf()
# contour plot
xx, yy = np.meshgrid(x,y)
pos = np.dstack((xx,yy))
rv = multivariate_normal(mean=mu, cov=cov)
fig2 = plt.figure()
ax2 = fig2.add_subplot(111)
ax2.contourf(xx, yy, rv.pdf(pos))
plt.savefig('5b2.png')
```

Listing 2: 2-D normal samples





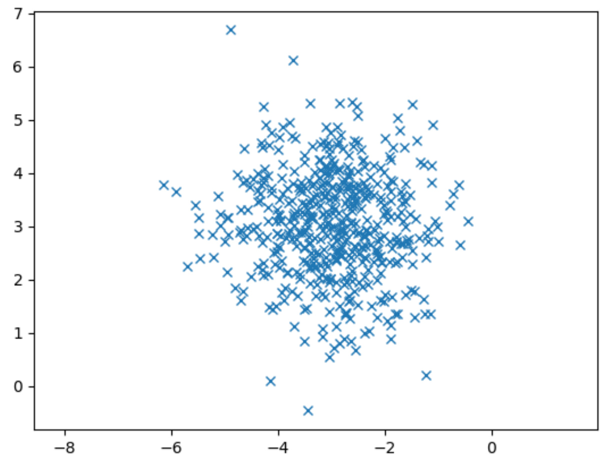
5.3

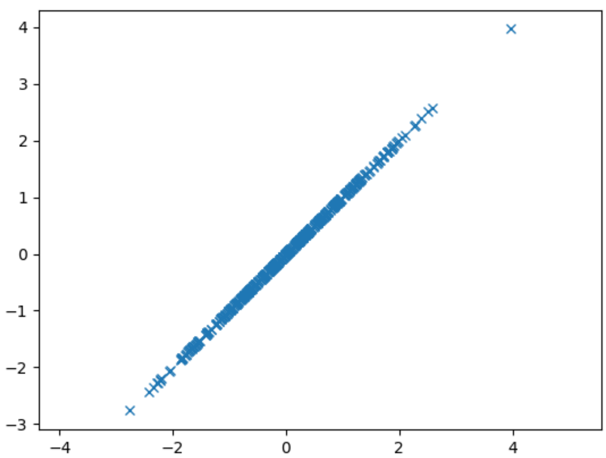
for a circle the correlation coefficient has to be zero. for a diagonal line the correlation coefficient has to be one or negative one. for a horizontal ellipsoid var sigma11 must be greater than sigma22. for each sample, M specifies the center of distribution.

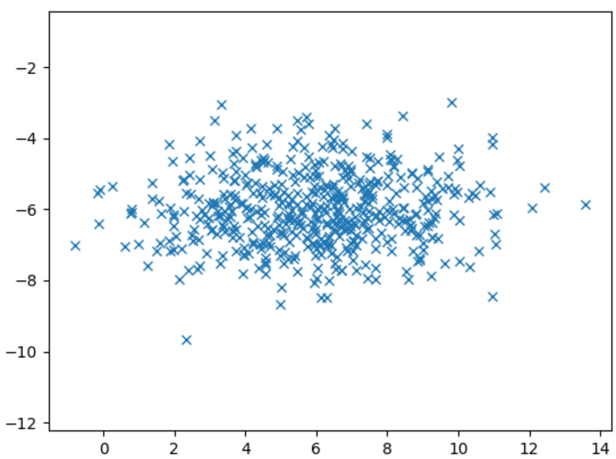
```
mus = [[-3, 3],[0,0],[6,-6]]
covs =[[[1, 0], [0, 1]],[[1,1],[1,1]],[[6,0],[0,1]] ]
#generating samples
for i in range(3):
    mu = mus[i]
    cov = covs[i]
    x,y = np.random.multivariate_normal(mu, cov, 500).T
    #plotting samples
    plt.plot(x,y, 'x')
    plt.axis('equal')
    plt.savefig('5c'+ str(i+1)+'.png')
    plt.clf()
```

Listing 3: 5-C

plots are in the next page in wanted order.


7
6
5
4
3
2
1
0
−8
−6
−4
−2
0


4
3
2
1
0
−1
−2
−3
−4
−2
0
2
4


−2
−4
−6
−8
−10
−12
0
2
4
6
8
10
12
14

5.4

$$d^2(x) = (x-\mu)^T \Sigma^{-1}(x-\mu) = \begin{bmatrix} x_1-2 \\ x_2-3\end{bmatrix}^T \times \left(\frac{1}{4-4\rho^2}\begin{bmatrix}4 & -2\rho \\ -2\rho & 1\end{bmatrix}\right) \times \begin{bmatrix} x_1-2 \\ x_2-3\end{bmatrix}$$

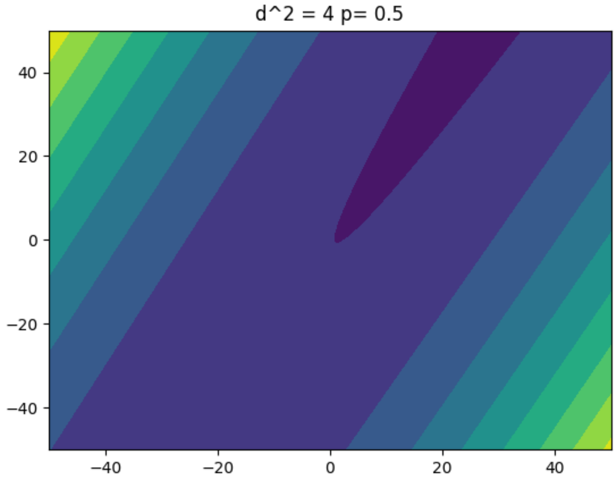
$$= \frac{1}{4-4\rho^2}\begin{bmatrix}4x_1-2\rho x_2+6\rho-8 \\ -2\rho x_1+x_2-3+4\rho\end{bmatrix}^T \times \begin{bmatrix}x_1-2\\x_2-3\end{bmatrix} = \left[(4x_1-2\rho x_2+6\rho-8)(x_1-2)+(-2\rho x_1+x_2-3+4\rho)(x_2-3)\right] \times \frac{1}{4-4\rho^2}$$

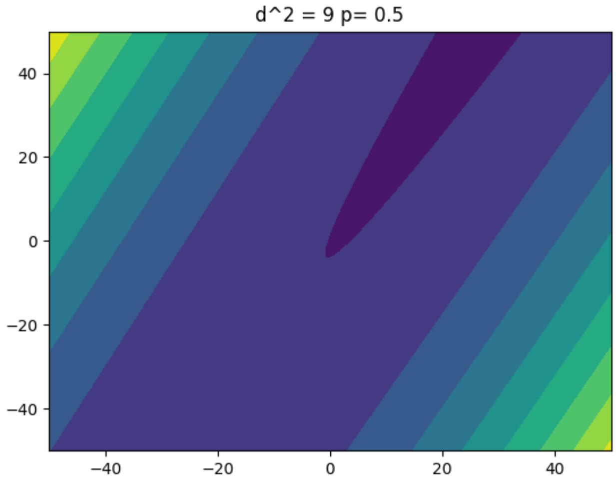
5.5

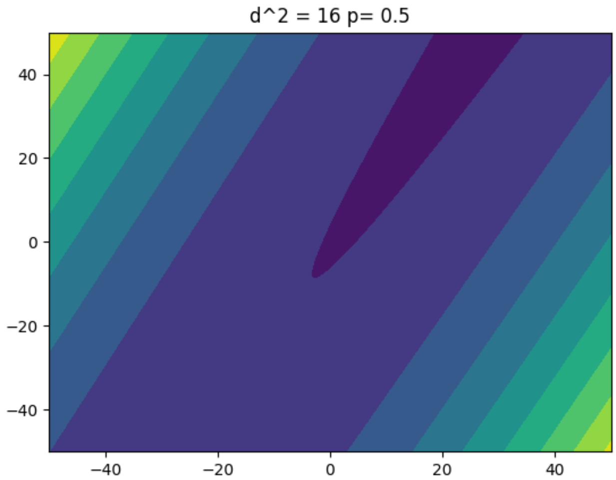
```
p = [-0.99,-0.5,0.5,0.99]
d2 = [4,9,16]

x1 = np.linspace(-50, 50, 5000)
x2 = np.linspace(-50, 50, 5000)

x_1, x_2 = np.meshgrid(x1, x2)

for i in p:
    for d in d2:
        #contour plot
        d22 = (1/(4-4*i**2))*((4*x_1 - 2*i*x_2 * 6 * i - 8)*(x_1 - 2) + (-2*i*x_1+x_2
    -3+4*i)*(x_2 - 3)) - d
        plt.contourf(x_1, x_2, d22)
        plt.title("d^2 = " + str(d) + ' p= ' + str(i))
        plt.savefig('5e'+str(i)+str(d)+'.png')
        plt.clf()
```

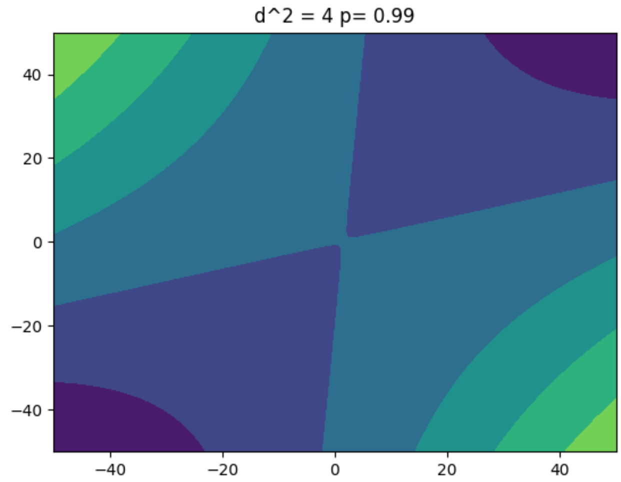
Listing 4: 5-e

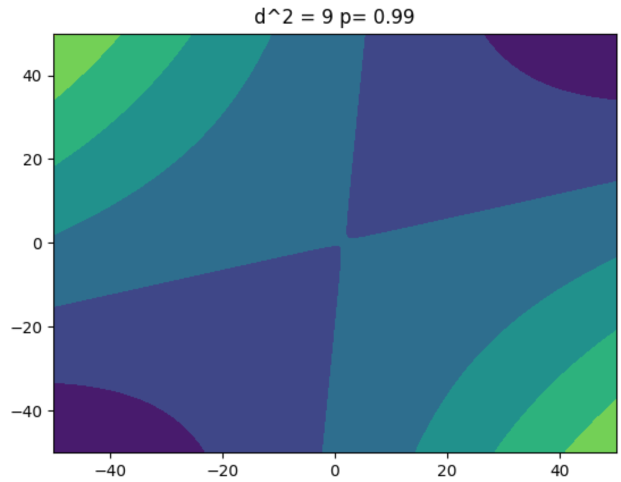
for p = 0.5 and p = -0.5 the plot is an ellipsoid and for p = 0.99 and p = -0.99 plot is a hyperbola

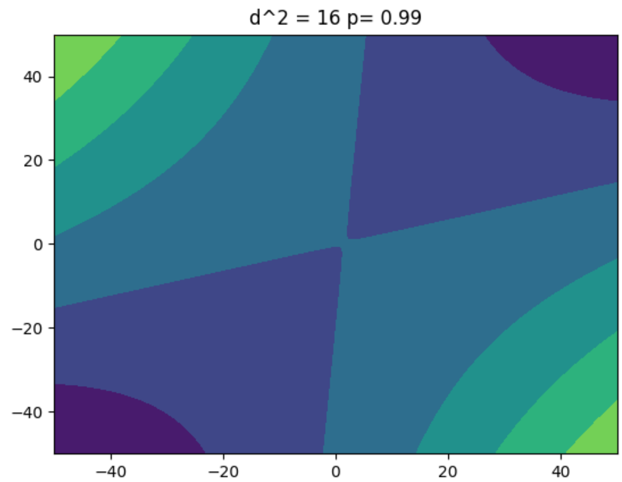

d^2 = 4 p= 0.5
40
20
0
−20
−40
−40
−20
0
20
40

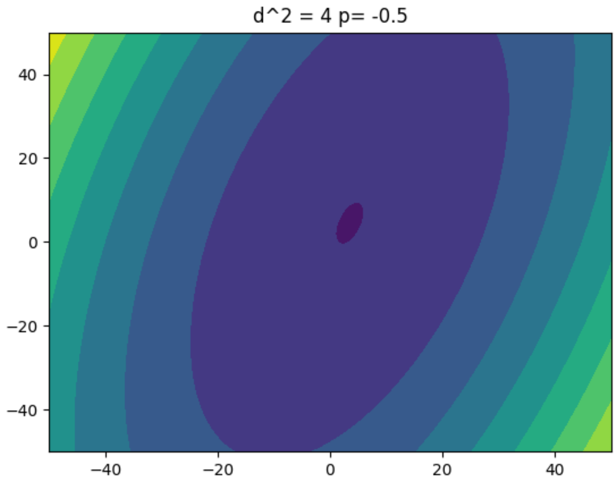

d^2 = 9 p= 0.5
40
20
0
−20
−40
−40
−20
0
20
40

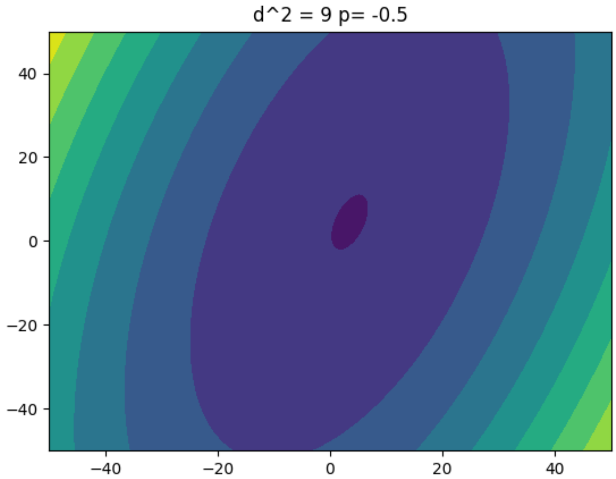

d^2 = 16 p= 0.5
40
20
0
−20
−40
−40
−20
0
20
40

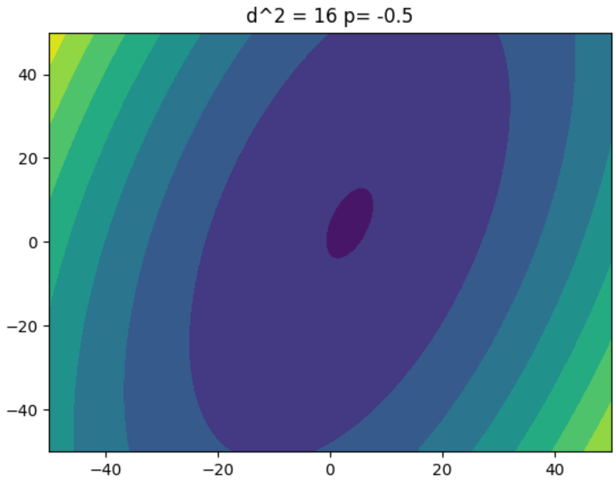

d^2 = 4 p= 0.99
40
20
0
−20
−40
−40
−20
0
20
40

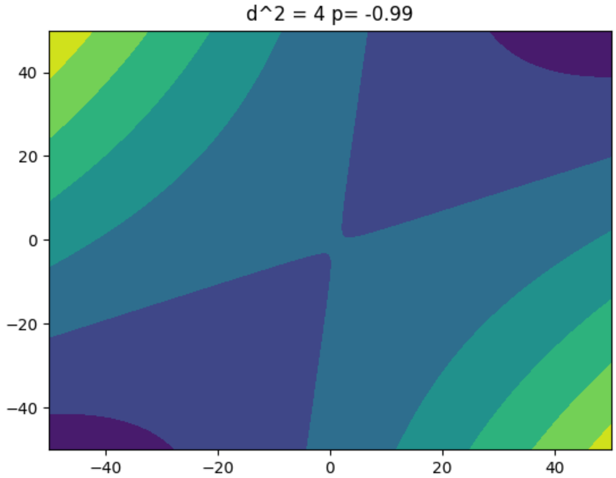

d^2 = 9 p= 0.99
40
20
0
−20
−40
−40
−20
0
20
40

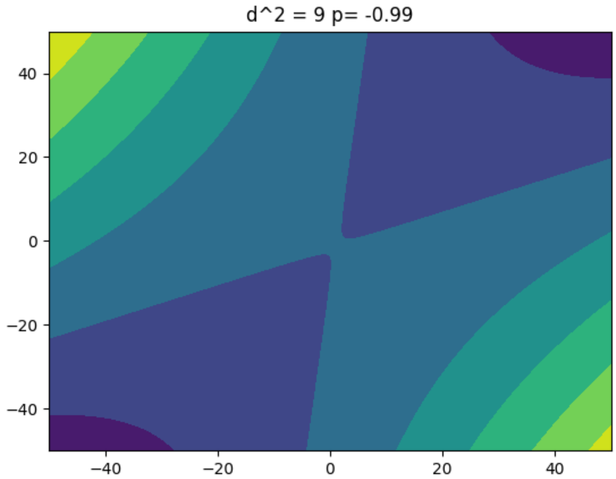

d^2 = 16 p= 0.99
40
20
0
−20
−40
−40
−20
0
20
40

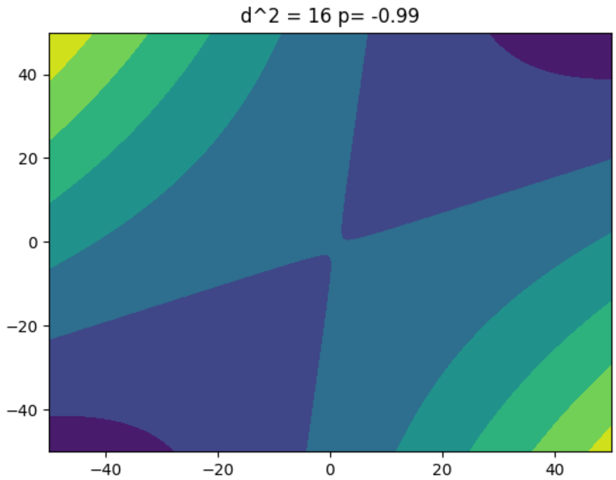

d^2 = 4 p= -0.5
40
20
0
−20
−40
−40
−20
0
20
40


d^2 = 9 p= -0.5
40
20
0
−20
−40
−40
−20
0
20
40


d^2 = 16 p= -0.5
40
20
0
−20
−40
−40
−20
0
20
40


d^2 = 4 p= -0.99
40
20
0
−20
−40
−40
−20
0
20
40


d^2 = 9 p= -0.99
40
20
0
−20
−40
−40
−20
0
20
40


d^2 = 16 p= -0.99
40
20
0
−20
−40
−40
−20
0
20
40

5.6

```
data = pd.read_csv('b.csv')

x1 = data['X'].to_numpy().reshape(len(data),1)
x2 = data['Y'].to_numpy().reshape(len(data),1)

#calculating mu
m1 = np.sum(x1) / 500
m2 = np.sum(x2) / 500

print('m1= ',m1,' m2=' , m2)

#calculating sigma
cov11 = np.sum(np.power((x1 - m1),2))/500
cov22 = np.sum(np.power((x2 - m2),2))/500
cov12 = np.dot((x1 - m1).T , (x2 - m2))/500

print('cov11= ',cov11,' cov22 = ' , cov22 ,' cov12=cov21= ', float(cov12))
```

Listing 5: 5-f

results are: m1= 1.995273636044089 m2= 1.0766133898598453

cov11= 1.9934111431614752 cov22 = 2.804900535041718 cov12=cov21= 0.9611637174990503

we see that M hat and sigma hat values are near the M and sigma values but not exactly the same if we choose a a bigger n these values become nearer to M and sigma

5.7

```
sigma = np.array([[2, 1], [1, 3]])
sigmahat = np.array([[1.9934, 0.96116], [0.96116, 2.8049]])

eigvals, eigvecs = np.linalg.eig(sigma)
eigvals2, eigvecs2 = np.linalg.eig(sigmahat)
v = eigvals
vhat = eigvals2

print(v)
print(np.diag(v))
print(np.diag(vhat))
```

Listing 6: 5-g

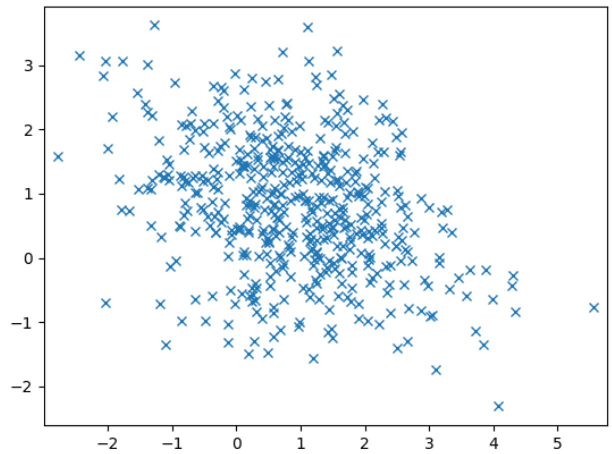
V = [1.38196601 3.61803399]

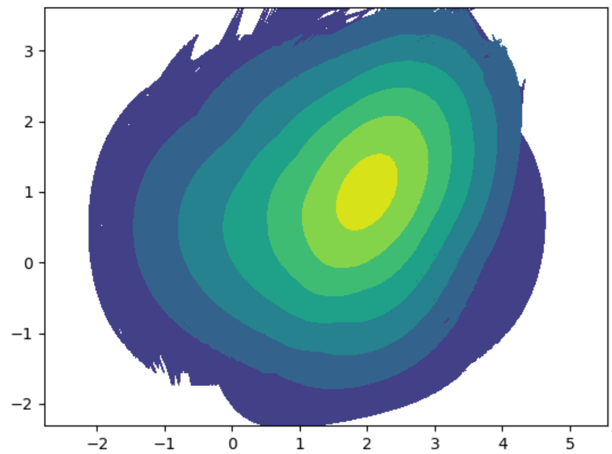
5.8

i will use whitening transformation in this section

```
mu = [2,1]
sigma = np.array([[1.9934, 0.9611], [0.9611, 2.8049]] )
data = pd.read_csv('b.csv')

x1 = data['X'].to_numpy().reshape(len(data),1)
x2 = data['Y'].to_numpy().reshape(len(data),1)
x = np.concatenate([x1,x2]).reshape([500,2])

#calculating transformation matrix
eigvals, eigvecs = np.linalg.eig(sigma)
v = np.power(eigvals , -1/2)
t = np.diag(v)
t = np.dot(eigvecs , t)
t = np.dot(t , eigvecs.T)


#transforming data
x = np.dot(x , t)


plt.plot(x[:,0],x[:,1], 'x')
plt.axis('equal')
plt.savefig('5h1.png')

plt.clf()
# contour plot
xx, yy = np.meshgrid(x[:,0],x[:,1])
pos = np.dstack((xx,yy))
rv = multivariate_normal(mean=mu, cov=list(sigma))
fig2 = plt.figure()
ax2 = fig2.add_subplot(111)
ax2.contourf(xx, yy, rv.pdf(pos))
plt.savefig('5h2.png')
```

Listing 7: 5-h

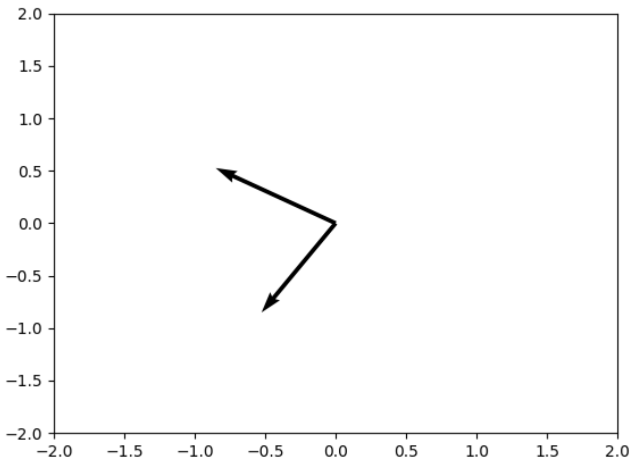

3
2
1
0
−1
−2
−2
−1
0
1
2
3
4
5


3
2
1
0
−1
−2
−2
−1
0
1
2
3
4
5

5.9

```python
sigma = np.array([[2, 1], [1, 3]] )

eigvals, eigvecs = np.linalg.eig(sigma)

print('eigenvalues: ' , eigvals)

print('eigenvectors: ',eigvecs)


soa = np.array([[0, 0, eigvecs[0,0], eigvecs[1,0]], [0, 0, eigvecs[0,1], eigvecs
    [1,1]]])
X, Y, U, V = zip(*soa)
plt.figure()
ax = plt.gca()
ax.quiver(X, Y, U, V, angles='xy', scale_units='xy', scale=1)
ax.set_xlim([-2, 2])
ax.set_ylim([-2, 2])
plt.draw()
plt.savefig('5i.png')
```

Listing 8: 5-i

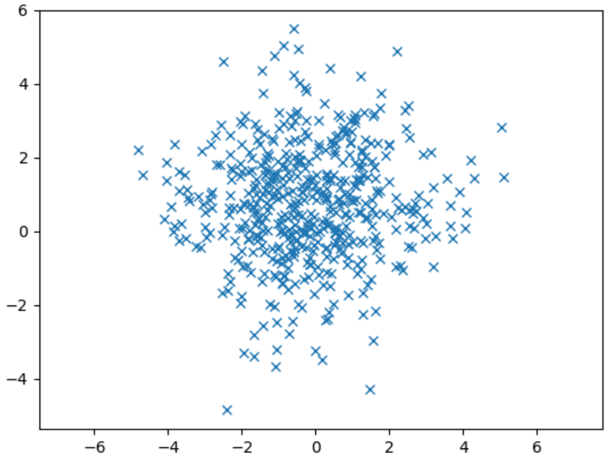


eigenvalues: [1.38196601 3.61803399] eigenvectors: [[-0.85065081 -0.52573111] [0.52573111 -0.85065081]]

5.10

```python
m = np.array([2,1])
sigma = np.array([[2, 1], [1, 3]] )
data = pd.read_csv('b.csv')
x1 = data['X'].to_numpy().reshape(len(data),1)
x2 = data['Y'].to_numpy().reshape(len(data),1)
x = np.concatenate([x1,x2]).reshape([500,2])
eigvals, eigvecs = np.linalg.eig(sigma)


p = np.array([[eigvecs[0][1] , eigvecs[0][0]],[eigvecs[1][1] , eigvecs[1][0]]])

y = np.dot((x-m) , p)

plt.plot(y[:,0],y[:,1], 'x')
plt.axis('equal')
plt.savefig('5j.png')
```

Listing 9: 5-j



samples seem to be more correlated than before and it seems that the mu vector has changed and its at (0,0)

5.11

```
sigma = np.array([[1.9934, 0.9611], [0.9611, 2.8049]] )
data = pd.read_csv('b.csv')

x1 = data['X'].to_numpy().reshape(len(data),1)
x2 = data['Y'].to_numpy().reshape(len(data),1)
x = np.concatenate([x1,x2]).reshape([500,2])

eigvals, eigvecs = np.linalg.eig(sigma)
v = np.power(eigvals , -1/2)
t = np.diag(v)
t = np.dot(eigvecs , t)
t = np.dot(t , eigvecs.T)

x = np.dot(x , t)

m1 = np.sum(x[:,0]) / 500
m2 = np.sum(x[:,1]) / 500

print('m1= ',m1,' m2=' , m2)

cov11 = np.sum(np.power((x[:,0] - m1),2))/500
cov22 = np.sum(np.power((x[:,1] - m2),2))/500
cov12 = np.dot((x[:,0] - m1).T , (x[:,1] - m2))/500

print('cov11= ',cov11,' cov22 = ' , cov22 ,' cov12=cov21= ', float(cov12))

cov = np.array([[cov11 , cov12],[cov12 , cov22]])

eigvals, eigvecs = np.linalg.eig(cov)

print('eigenvalues: ' , eigvals)

print('eigenvectors: ',eigvecs)
```

Listing 10: 5-k

cov11= 1.5295345656650166 cov22 = 1.0635059068174582 cov12=cov21= -0.42602735128304425 eigenvalues: [1.78210749 0.81093298] eigenvectors: [[0.8601921 0.50997015] [-0.50997015 0.8601921]] covariance matrix is near the identity matrix but not exactly the same.

6 Some Explanatory Questions

6.1 Central Limit Theorem

The importance of the central limit theorem stems from the fact that, in many real applications, a certain random variable of interest is a sum of a large number of independent random variables. In these situations, we are often able to use the CLT to justify using the normal distribution. Examples of such random variables are found in almost every discipline. The CLT is also very useful in the sense that it can simplify our computations significantly. If you have a problem in which you are interested in a sum of one thousand i.i.d. random variables, it might be extremely difficult, if not impossible, to find the distribution of the sum by direct calculation. Using the CLT we can immediately write the distribution, if we know the mean and variance of the Xi's. it is often stated that if n(number of iid random variables) is larger than or equal to 30, then the normal approximation is very good.

HOw to apply CLT

1. write the random variable of interest, Y, as the sum of n i.i.d random variables (X_i's):

$$Y = X_1 + X_2 + \dots + X_n$$

2. Find $E[Y]$ and $Var[Y]$ by noting that:

$$E[Y] = n\mu \quad , \quad Var(Y) = n\sigma^2$$

where $\mu = E[X_i]$ and $\sigma^2 = Var(X_i)$

3. According to the CLT, conclude that $\frac{Y - E[Y]}{\sqrt{Var[Y]}} = \frac{Y - \mu n}{\sqrt{n}\,\sigma}$ is approximately standard normal(Z) thus:

$$P(y_1 \leq Y \leq y_2) = P\left(\frac{y_1 - n\mu}{\sqrt{n}\,\sigma} \leq \frac{Y - n\mu}{\sqrt{n}\,\sigma} \leq \frac{y_2 - n\mu}{\sqrt{n}\,\sigma}\right)$$

$$\simeq \Phi\left(\frac{y_2 - n\mu}{\sqrt{n}\,\sigma}\right) - \Phi\left(\frac{y_1 - n\mu}{\sqrt{n}\,\sigma}\right)$$

Figure 13: CLT

6.2 difference between a feature and a measurement

6.3 Does a covariance matrix need to be symmetric?

6.c

$$Var[X]^T = E[(X-E[X])(X-E[X])^T]^T$$

$$= E[((X-E[X])(X-E[X])^T)^T]$$

$$= E[(X-E[X])(X-E[X])^T]$$

$$= Var[X] \implies Var[X] \text{ is Symmetric}$$

Figure 14: 6-c

6.4 zero eigenvalue

Geometrically, having one or more eigenvalues of zero simply means the nullspace is nontrivial. It means that some nonzero vector is mapped to zero times itself - that is, to the zero vector. By linearity, every scalar multiple of this vector is also mapped to the zero vector. As a consequence the matrix is not invertible, because you have an infinite number of vectors being mapped to the zero vector, and an infinity-to-one mapping cannot be run backwards.

6.5 whitening transformation

The goal of whitening is to make the input less redundant; more formally, our desiderata are that our learning algorithms sees a training input where (i) the features are less correlated with each other, and (ii) the features all have the same variance. so we use whitening transformation in preprocessing step.