M248QuizWeek2

January 28, 2021

1 1) List five things you learned in the first two weeks of classes. Explain.

2 2) What is the difference between the uniform distribution and the gaussian distribution?

3 3) Let's plot the two dimensional bivariate Gaussian distribution

The general formula for the two dimensional bivariate Gaussian distribution is:

$$g(x,y) = \frac{1}{\sqrt{(2\pi)^2 \det\begin{pmatrix} \sigma_1^2 & \rho\sigma_1\sigma_2 \\ \rho\sigma_1\sigma_2 & \sigma_2^2 \end{pmatrix}}} e^{-\left\{\frac{1}{2}\begin{pmatrix} x-\mu_1 & y-\mu_2 \end{pmatrix}\begin{pmatrix} \sigma_1^2 & \rho\sigma_1\sigma_2 \\ \rho\sigma_1\sigma_2 & \sigma_2^2 \end{pmatrix}^{-1}\begin{pmatrix} x-\mu_1 \\ y-\mu_2 \end{pmatrix}\right\}}$$

This can be compacted into

$$g(x,y) = \frac{1}{\sqrt{(2\pi)^2 \det(\Sigma)}} e^{-\left\{\frac{1}{2}(u-\mu)^T\Sigma^{-1}(u-\mu)\right\}}$$

Compare the above formula to the one dimensional Gaussian distribution function:

$$g(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

- When there is only one random variable you only have one mean μ and one standard deviation σ.
- When there are two random variables, you have two means $\begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix}$, two standard deviations $\begin{pmatrix} \sigma_1 \\ \sigma_2 \end{pmatrix}$, and the product σ^2 will be replaced by the covariance matrix $\Sigma = \begin{pmatrix} \sigma_1^2 & \rho\sigma_1\sigma_2 \\ \rho\sigma_1\sigma_2 & \sigma_2^2 \end{pmatrix}$ and its determinant. Here, ρ is the correlation between the two random variables.

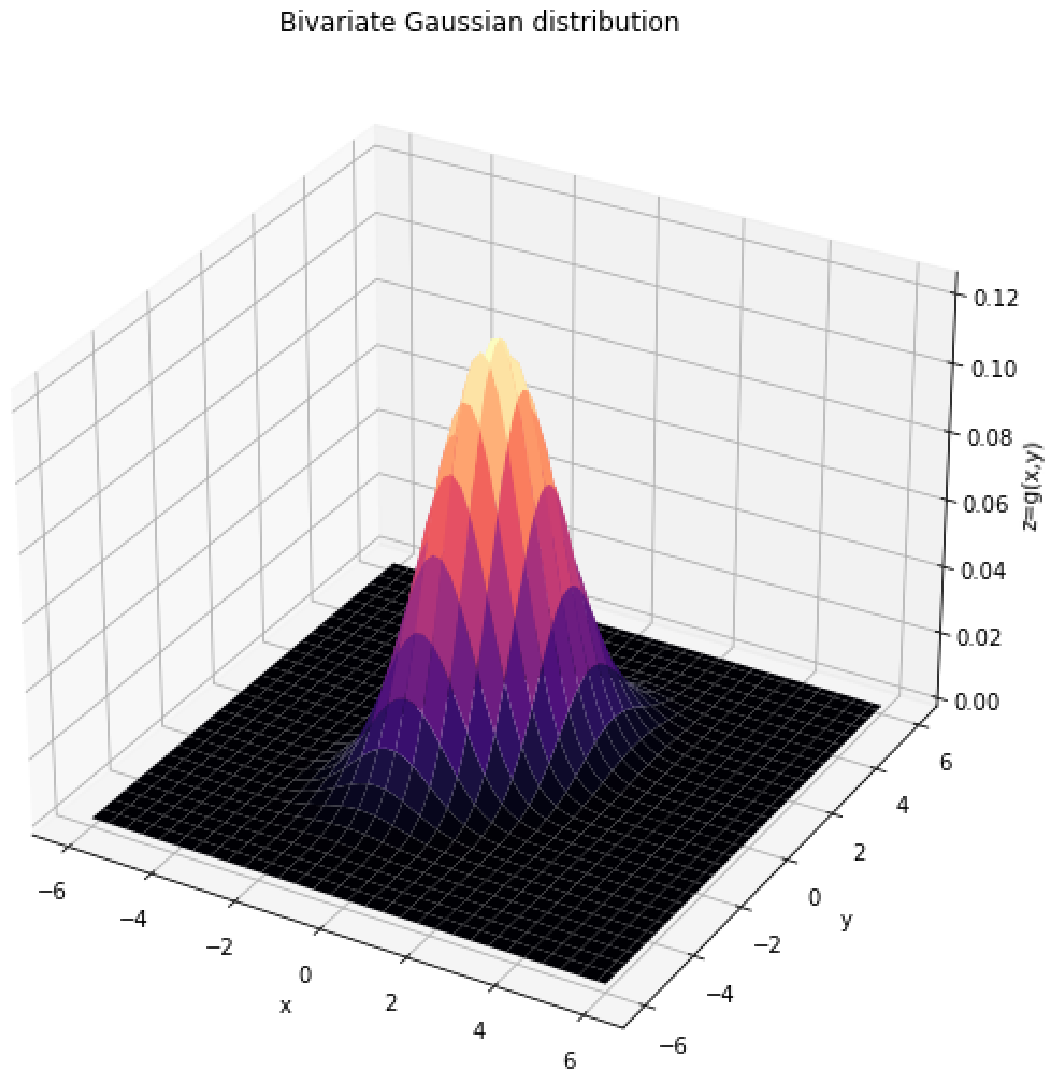
The following code plots the bivariate normal distribution. Read the code carefully then explain it step by step in the follwing cell.

[160]:
```
import numpy as np
import matplotlib.pyplot as plt
from matplotlib import cm
```

```
from mpl_toolkits import mplot3d

# Our 2-dimensional distribution will be over variables X and Y
n = 60 # number of grid points on each axis
X = np.linspace(-6, 6, n)
Y = np.linspace(-6, 6, n)
X, Y = np.meshgrid(X, Y)


# Mean vector and covariance matrix
mu_1=0
mu_2=0

mu = np.array([mu_1, mu_2])
Sigma = np.array([[ 1 , 3/5], [3/5,  2]])

# Pack X and Y into a single 3-dimensional array
pos = np.empty(X.shape + (2,))
pos[:, :, 0] = X
pos[:, :, 1] = Y

def multivariate_gaussian(pos, mu, Sigma):
    """This function returns the bivariate Gaussian distribution on array pos.
       pos is an array constructed by packing the meshed arrays of variables x␣
 ↪and y into its last dimension.
    """

    n = 2 # this is the dimension we are in
    Sigma_det = np.linalg.det(Sigma)
    Sigma_inv = np.linalg.inv(Sigma)
    N = np.sqrt((2*np.pi)**n * Sigma_det) # this is the normalizing constant
    # This einsum or Einstein sum calculates (x-mu)^T.Sigma^-1.(x-mu) in a␣
 ↪vectorized
    # way across all the input variables.
    fac = np.einsum('...k,kl,...l->...', pos-mu, Sigma_inv, pos-mu)

    return np.exp(-fac / 2) / N

# The distribution on the variables X, Y packed into pos.
Z = multivariate_gaussian(pos, mu, Sigma)

# Create a surface plot
fig = plt.figure(figsize=(9,9))
picture = fig.gca(projection='3d')
picture.plot_surface(X, Y, Z, cmap=cm.magma)
# set labels for axes
picture.set_xlabel('x')
```

```
picture.set_ylabel('y')
picture.set_zlabel('z=g(x,y)')
# set image title
picture.set_title('Bivariate Gaussian distribution')
```

[160]: Text(0.5, 0.92, 'Bivariate Gaussian distribution')



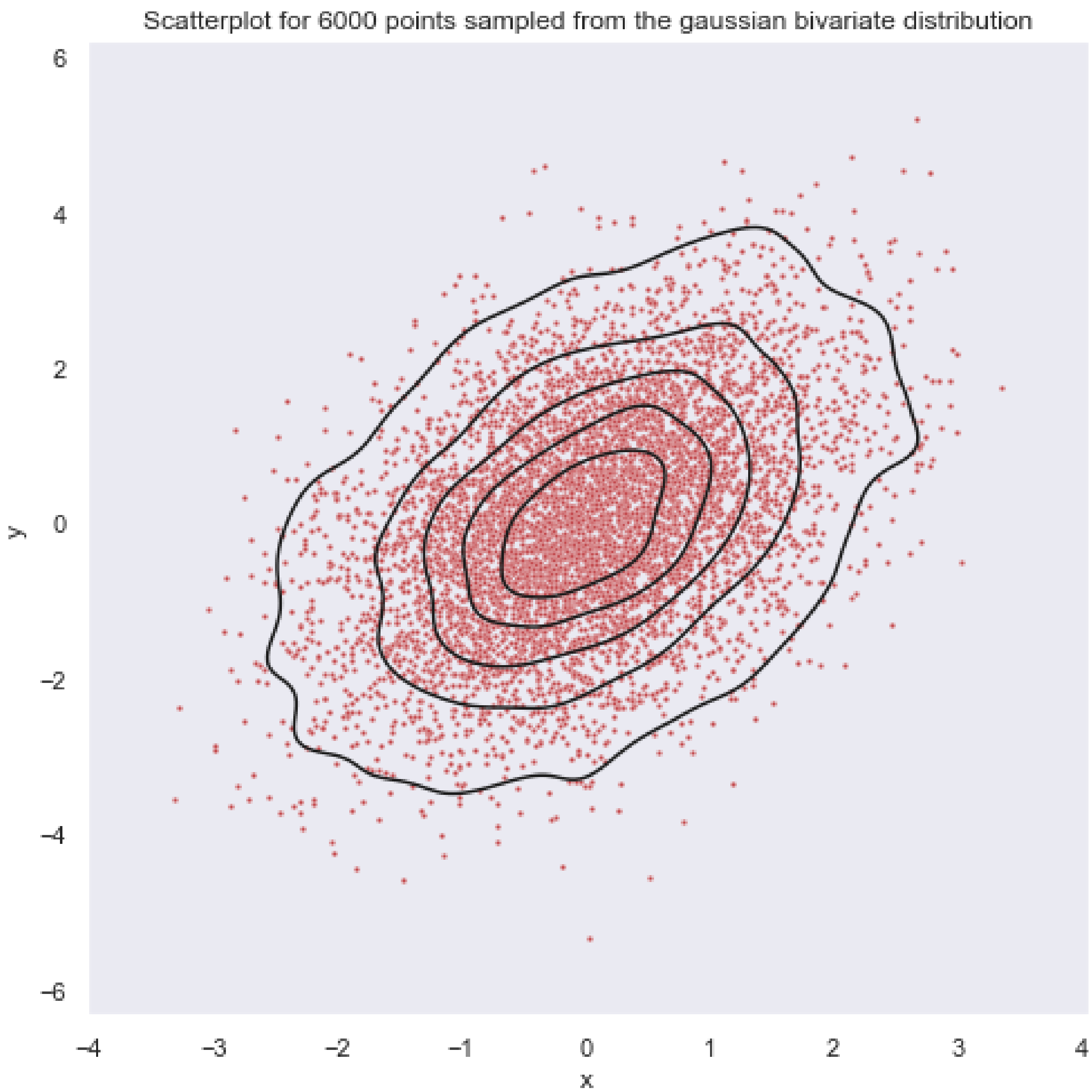
4 Explain the above code line by line.

5 4) Now let's sample n points (x_i, y_i) from the bivariate Gaussian distribution and use seaborn library to plot the sampled points in the x-y plane. These should look like the shadow in the x-y plane of the surface plot of $g(x, y)$ above.

Read the following code then explain it line by line in the next cell.

[182]:
```python
import seaborn as sns

# Simulate data from a bivariate Gaussian
n = 6000
mean = [0, 0]
cov = [(1, 3/5), (3/5, 2)]
x, y = np.random.multivariate_normal(mean, cov, n).T

# Draw a scatterplot with density contours
fig = plt.figure(figsize=(8,8))
sns.scatterplot(x=x, y=y, s=6, color="r")
sns.kdeplot(x=x, y=y, levels=6, color="k", linewidths=1.5)
# set labels for axes
plt.xlabel('x')
plt.ylabel('y')
# set image title
plt.title('Scatterplot for {} points sampled from the gaussian bivariate␣
 ↪distribution'.format(n))
```

[182]:
```
Text(0.5, 1.0, 'Scatterplot for 6000 points sampled from the gaussian bivariate
distribution')
```



6 Explain the above code line by line.

7 5) Create a class for a car dealership called cars. The objects belonging to the class should have 5 attributes: make, model, year, color, and VIN number. Then specify five instances belonging to this class, and call certain attributes for three of these instances.

[ ]:
```
# Your code here
```