Yixiao Zhang
31353282
yz3a19@soton.ac.uk

Multi-variate Gaussian
COMP6245 Lab 1

1 Linear Algebra

1.1 Task 1-9

In task 1-9 of linear algebra section, some functions of NumPy package related to matrix are used: function ***np.dot(array1,array2)*** is used to make dot product, function ***np.linalg.norm(array)*** is used to compute norm, function ***np.linalg.arccos(vector,vector)*** is used to compute arccos value, function ***np.trace(matrix)*** is used to compute the sum along diagonals of the matrix, function ***np.linalg.det(matrix)*** is used to compute the determinant of the matrix, function ***np.linalg.inv(matrix)*** is used to compute the inverse of the matrix, function ***np.linalg.eig(matrix)*** is used to compute the eigenvalues and right eigenvectors of the matrix.

1.2 Task 10: Singular Value Decomposition (SVD)

The result is $U@U.T \approx \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$, which is a unit matrix. The specific structure of matrix B is that matrix B is a symmetric. Using matrix $B = \begin{bmatrix} 3 & 2 & 1 \\ 2 & 6 & 5 \\ 1 & 5 & 9 \end{bmatrix}$ can calculate its normalized eigenvectors $U = \begin{bmatrix} -0.195 & -0.746 & 0.637 \\ -0.600 & -0.423 & -0.679 \\ -0.776 & 0.515 & 0.365 \end{bmatrix}$ and its eigenvalues $D = \begin{bmatrix} 13.11766327 & 3.44324229 & 1.43909444 \end{bmatrix}$. λ_1 and λ_2 are two eigenvalues extra from D, $\vec{v_1}$ and $\vec{v_2}$ are two eigenvectors extra from B.

$$B \cdot \vec{v_1} = \lambda_1 \vec{v_1}, B \cdot \vec{v_2} = \lambda_2 \vec{v_2}$$

$$\vec{v_2}^T \cdot B \cdot \vec{v_1} = \lambda_1 \vec{v_2}^T \cdot \vec{v_1}$$

As matrix B is a symmetric matrix, $B = B^T$

$$\vec{v_2}^T \cdot B \cdot \vec{v_1} = \vec{v_2}^T \cdot B^T \cdot \vec{v_1} = (B \cdot \vec{v_2})^T \cdot \vec{v_1} = \lambda_2 \vec{v_2}^T \cdot \vec{v_1}$$

As $\lambda 1 \neq \lambda 2 \neq 0$

$$\lambda_2 \vec{v_2}^T \cdot \vec{v_1} = \lambda_1 \vec{v_1}^T \cdot \vec{v_2}$$

$$\vec{v_2}^T \cdot \vec{v_1} = 0$$

That is why matrix U is an orthogonal matrix when matrix B is a symmetric matrix.

2 Random Numbers and Univariate Distributions

2.1 Using NumPy package to generate uniform random distribution and univariate distribution

Function ***rand()*** and ***randn()*** from the NumPy package ***np.random*** are used to generate uniform random numbers and Gaussian distributions respectively. Figure 1 shows that 1000 uniform random numbers were

generated by function ***np.random.rand(1000,1)*** and put in to 20 bins. A uniform random distribution can be seen from the Figure 1, with the increasing of sample number, the sum of each bin become averagely. The NumPy package also provide a function to generate univariate Gaussian distribution automatically. Figure 2 shows that 1000 numbers which obeys univariate Gaussian distribution were generated by function ***np.random.randn(1000,1)*** and put in to 20 bins.

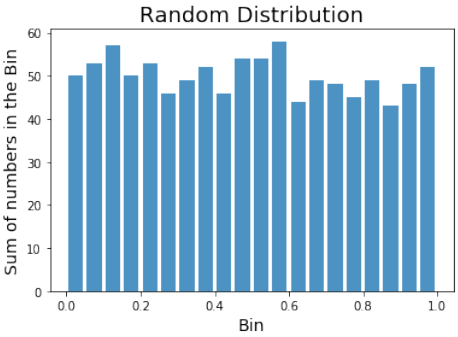

Figure 1: Random distribution

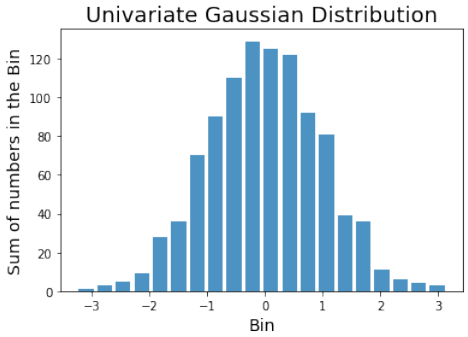

Figure 2: Univariate Gaussian distribution

2.2 Generating univariate Gaussian distribution manually and central limit theorem

Central limit theorem is a probability theory which gives a way to generate univariate Gaussian distribution from some random numbers. Central limit theorem describe that when group of random numbers are added together, the sum of each group obeys the Gaussian distribution. With the increasing of number of samples, the distribution obeys the Gaussian distribution better.

In the experiment, function ***np.random.rand(size,1)*** which has been used in section above is used to generate random numbers and function ***np.sum(listOfNumbers)*** is used to calculate the sum of each group. 100 samples and 10000 samples are generated and put in to bins respectively as Figure 3 and Figure 4 shows. Comparing Figure 3 with Figure 4, the increasing of number of samples makes the distribution obeys the Gaussian distribution better which verify central limit theorem as well.

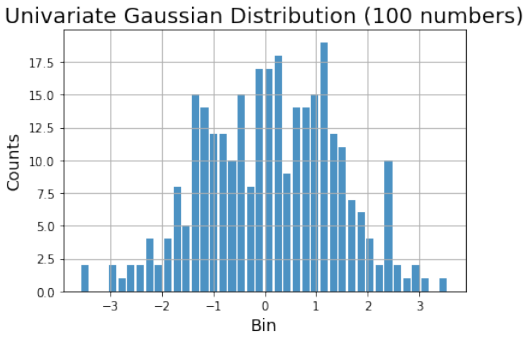

Figure 3: Gaussian distribution (100 numbers)

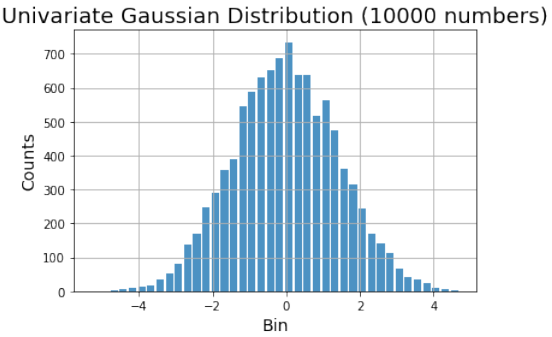

Figure 4: Gaussian distribution (10000 numbers)

3 Uncertainty in Estimation

Uncertainty in the estimation of variance could lower with the number of samples increasing. This is why a bigger dataset could help machines to get a better performance when training a model in machine learning process. In the experiment, function ***np.linspace(100,1500,100)*** is used to generate 100 numbers between 100 and 1500 and have same space between each number. These numbers are used as sample size in 100 times estimation. In each estimation, function ***np.random.randn(sampleSize,1)*** is used to generate univariate

Gaussian distribution and function ***np.var(listOfNumbers)*** is used to calculate variance of these group of numbers.

Figure 5 shows the trend of variance during 100 times estimation. The lower variance indicates the better precise of data generated which obeys univariate Gaussian distribution. As Figure 5 shows, variance decrease dramatically when sample size smaller than 600 and become steady when sample size bigger than 600 which means it obeys univariate Gaussian distribution better when improve the sample size.

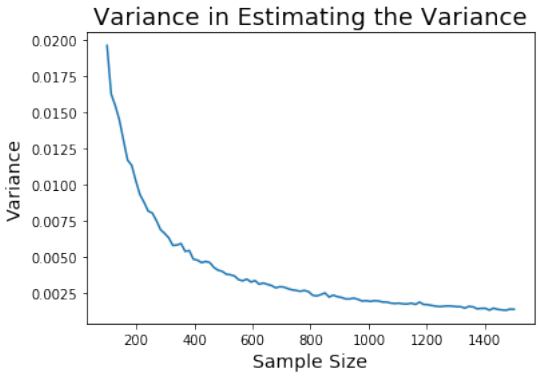

Figure 5: Variance in 100 times estimating the variance

4 Bivariate Gaussian Distributions

The bivariate Gaussian distribution density can be calculate by Formula (1). The function ***np.linalg.inv(matrix)***, ***np.linalg.det(matrix)***, ***np.exp(value)***, ***np.dot(matrix,matrix)*** can be used to compute Formula (1).

$$P(x) = \frac{1}{(2\pi)^{\frac{1}{2}}\,|C|^{\frac{1}{2}}} \exp\left\{-\frac{1}{2}(x-m)^T C^{-1}(x-m))\right\} \tag{1}$$

Three different values of means and covariance matrices m1,C1 and m2,C2 and m3,C3 are used to generate figures of bivariate Gaussian distribution. C is covariance matrice and m is the mean of the distribution as well as the centre of scatter.

$$m1 = \begin{bmatrix} 0 \\ 2 \end{bmatrix}, C1 = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} \quad and \quad m2 = \begin{bmatrix} 2 \\ 0 \end{bmatrix}, C2 = \begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix} \quad and \quad m3 = \begin{bmatrix} -2 \\ -2 \end{bmatrix}, C3 = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$$

The results are shown as Figure 6 and Figure 7. The red scatter is generated by m1 and C1, the blue scatter is generated by m2 and C2, the blue scatter is generated by m3 and C3.

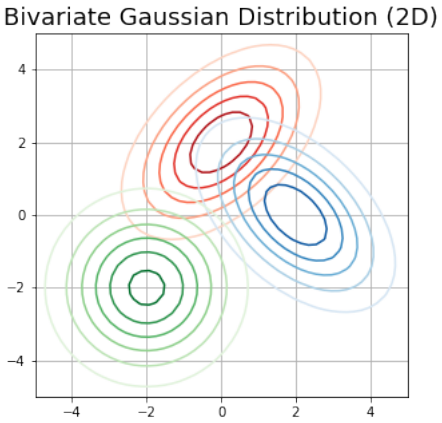

Figure 6: 2D bivariate Gaussian distribution

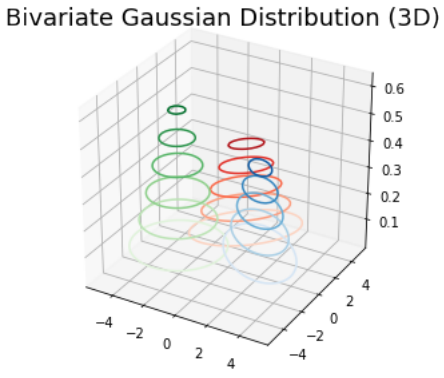

Figure 7: 3D bivariate Gaussian distribution

5 Sampling from a Multivariate Gaussian Distributions

A 1000-sample bivariate Gaussian distribution is generated by function ***np.random.randn(1000,2)*** as orange points in Figure 8 shows. These points are also a scatter of isotropic Gaussian densities and could be seen as $X \sim N(0, 1)$.

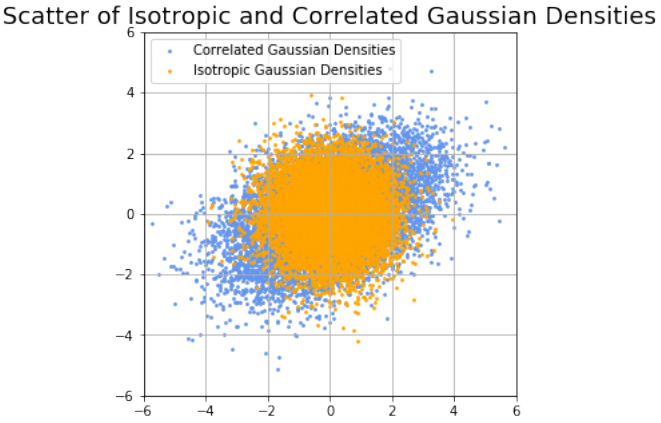

Figure 8: Sampling from a multivariate Gaussian distributions

A covariance matrix $C = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}$ is given and $A^T A = C$ is estimated, so that $A = \begin{bmatrix} 1.41 & 0.00 \\ 0.71 & 1.22 \end{bmatrix}$. A correlated Gaussian distribution is generated by $Y = AX$ as blue points in Figure 8 shows. As Figure 8 shows, isotropic Gaussian distribution could linear transform to correlated Gaussian distribution using a covariance matrix C. This correlated Gaussian distribution can be seen as $Y \sim N(A0, A1A^T)$.

6 Distribution of Projection

The covariance matrix $C1 = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}$ and covariance matrix $C2 = \begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix}$ are given. Two correlated Gaussian distributions are generated using $C1$ and $C2$ as Figure 9 shows. The projected variance of two distributions are showed in Figure 10.

These correlated Gaussian distributions are linear transform from isotropic Gaussian distributions which is round. So that the shape of correlated Gaussian distributions are oval of which distance between origin and boundary is Sin. That is why the shape of projections are Sin.

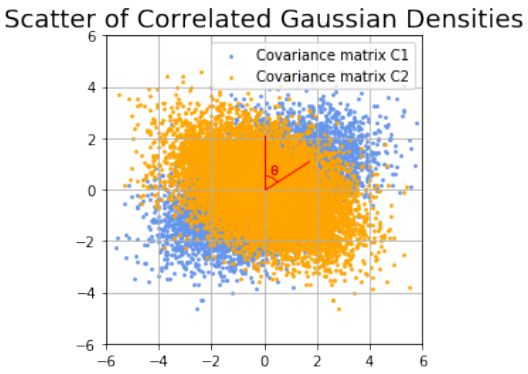

Figure 9: Scatter of correlated Gaussian densities

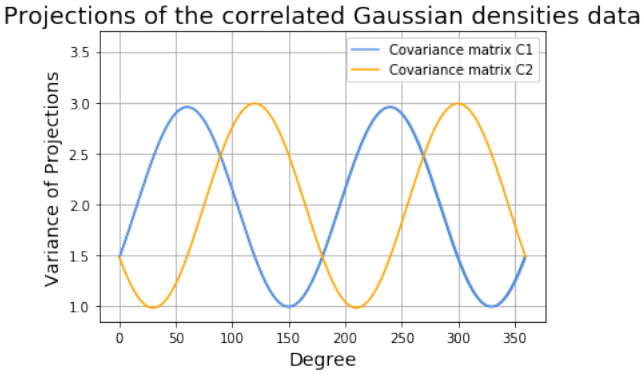

Figure 10: Projections of correlated Gaussian densities