South Dakota School of Mines and Technology

Intro to Data Sci and Machine Learning, Fall 2022

CSC 592 - M01

Lab # 1

The aim of this project was to create a random dataset that met the following criterion:

- The dataset should have 4 labels tied to it.
- The dataset should be a 3 features corresponding to the 3 axes in 3D space.
- The points for each feature should be picked from randomly from a multivariate.normal distribution generated from a random covariance matrix and a random mean.
- The mean is to be randomly picked within a 20x20x20 cube centered at 0.
- The covariance matrix is supposed to be a diagonal covariance matrix with the diagonal elements randomly picked from the range [0,10)
- Each label should have a random number of points in the range [1,100].

The Covariance Matrix:

The diagonal covariance matrix was created by first picking 3 random values from the range [0, 10). These values were then used to construct the diagonal matrix using the *np.diag* function. An insight from this was that since the matrix was a diagonal matrix, the variance of each dimension solely depends on itself which implies that each dimension's normal distribution is independent from the other. The screenshot below shows the code that implemented this.

```
'''
Function that declares a random covariance matrix with
self variance between 0 and 10
'''

def cov_matrix_gen():
    cov_rand = 10*np.random.rand(3)
    cov_matrix = np.diag(cov_rand)
    return cov_matrix
```

Fig 1. Covariance Matrix Generator

The Plot:

The plot parameters were adjusted to have limits of [-10,10] for each dimension. They were also adjusted so that datapoints corresponding to certain label had a color tied to them. (0 - red, 1 - green, 2 - blue, 3 - black). The screenshot below shows this.

```
# setting plot parameters
fig = plt.figure()
ax = plt.axes(projection='3d')
ax.set_xlim(left=-10,right=10)
ax.set_ylim3d(-10,10)
ax.set_zlim3d(-10,10)
colors = ['red', 'green', 'blue', 'black']
```

Fig. 2. Plot Parameters

The Dataset:

The dataset was generate to meet the aforementioned conditions. A random mean was picked within a 20x20x20 box centered at (0,0,0). Using this and the randomly generated covariance matrix, a multivariate normal distribution of 100 points was created. The code snippet below shows that (Fig 3).

```
    cov_matrix = cov_matrix_gen()
    mean = 20*np.random.rand(3) - 10 #pick a random 3 dimensional mean

    # create an random x distribution using the cov_matrix
    x = np.random.multivariate_normal(mean, cov_matrix, 100)
```

Fig 3. MultiNorm Distribution Generation

A random number of elements were picked from distribution to contribute to the overall dataset. These were assigned a label value of ***i***. A loop iterated ***i*** in the range [0,3]. From a broader level, this code generated data corresponding to each label sequentially, while simultaneously plotting each label as soon as it was generated.
The figure below shows the final result.