

## CS 214: Artificial Intelligence Lab

**Common Instructions:** You have to submit a code without any syntax errors. Don't copy from others; write your own piece of code with comments as much as possible. Try to Run with the test cases before submitting the code.

### Assignment 4

#### NOTE:

- Test your code and submit it on Moodle (<https://moodle.iitdh.ac.in/user/index.php?id=992>).
- Deadline for submission is **February 28 2024 by (11:59 PM)**
- We will run a plagiarism check for all the submissions; if found, a penalty will be applied

#### Lab Instructions:

- Construct a KNN and LwP classifier to separate data points sampled from 2 multivariate Gaussian distributions with a non-identity covariance matrix. Use the code below to generate the data.

```
num_samples = 100
# True means and covariance matrices of the Gaussians generating the data from two classes
class_1_mean = np.array([1.0, 1.0])
class_2_mean = np.array([-2.0, -2.0])
# Let's use non-spherical classes (non-identity covariance matrix for each Gaussian)
class_1_cov = np.array([[0.8, 0.4], [0.4, 0.8]])
class_2_cov = np.array([[0.8, -0.6], [-0.6, 0.8]])

X_class_1 = np.random.multivariate_normal(class_1_mean, class_1_cov, num_samples)
X_class_2 = np.random.multivariate_normal(class_2_mean, class_2_cov, num_samples)

X_train = np.vstack((X_class_1, X_class_2))
y_train = np.hstack((np.zeros(num_samples), np.ones(num_samples)))
```

- Draw the decision boundaries obtained from KNN and LwP and specify the prototypes obtained in LwP. Color the points belonging to the same class with the same color.

The code below is used to create the decision boundary. Several points are generated and classified with color change implying change in class.

```
x_min, x_max = X_train[:, 0].min() - 1, X_train[:, 0].max() + 1
y_min, y_max = X_train[:, 1].min() - 1, X_train[:, 1].max() + 1

xx, yy = np.meshgrid(np.arange(x_min, x_max, 0.02),
                     np.arange(y_min, y_max, 0.02))
Z = learning_with_prototype(X_train, y_train, np.c_[xx.ravel(), yy.ravel()])
Z = Z.reshape(xx.shape)

plt.contourf(xx, yy, Z, alpha=0.3)

plt.xlabel('Feature 1')
plt.ylabel('Feature 2')
plt.title('Learning with Prototype (LwP) Classifier')
plt.legend()
plt.grid(True)
plt.show()
```

- Repeat the above exercise for points sampled from two Chi-square distributions with different degrees of freedom. The x and y coordinates of the points can be sampled from the same distribution. Generate 100 points each. Here is some sample code to generate the relevant points.

```
import numpy as np

k1 = 7
k2 = 10
num_samples = 100
points_class_1 = np.random.chisquare(k1, (num_samples, 2))
points_class_2 = np.random.chisquare(k2, (num_samples, 2))

X_train = np.vstack((points_class_1, points_class_2))
y_train = np.hstack((np.zeros(num_samples), np.ones(num_samples)))
```

- Finally, compare the performance of LwP and KNN for the below dataset.  
<https://www.kaggle.com/datasets/akalyasubramanian/dataset-for-knn-classification>