

Lab: AI & ML Lab (23A31403)

College: Andhra Engineering College

12. Demonstrate Naïve Bayes Classification algorithm.

Math

4.5.1 Multi-Class Classification ✓

We have discussed the use of the Bayes classifier in the two-class case. It can be easily used to deal with multi-class cases, that is, when the number of classes is more than 2. It may be described as follows:

- Let the classes be C_1, C_2, \dots, C_q , where $q \geq 2$.
- Let the prior probabilities be $P(C_1), P(C_2), \dots, P(C_q)$.
- Let x be the test pattern to be classified as belonging to one of these q classes.
- Compute the posterior probabilities using Bayes' rule

$$P(C_i|x) = \frac{P(x|C_i)P(C_i)}{\sum_{j=1}^q P(x|C_j)P(C_j)}, \text{ for } i = 1, 2, \dots, q$$

- Assign the test pattern x to class C_l if

$$P(C_l|x) \geq P(C_i|x), \text{ for } i = 1, 2, \dots, q$$

- In the case of a tie (two or more of the largest-valued posteriors are equal), assign arbitrarily to any one of the corresponding classes. In practice, breaking the tie arbitrarily is the prescription suggested for any ML model.
- In this case, the probability of error is the sum of the posterior probabilities of the remaining $q-1$. We know that the posteriors across all the q classes add up to 1, that is, $\sum_{i=1}^q P(C_i|x) = 1$. So, if x is assigned to class C_l , then the probability of error is $1 - P(C_l|x)$.
- In this case also, we have average probability of error as

$$\int_x P(\text{error}|x)P(x)dx = \int_x (1 - P(C_l|x))P(x)dx$$

Its Changes Based On Problem Statement

Code

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.naive_bayes import GaussianNB

X = np.array([
    [0.3,1],[0.5,4.5],[1,2.3],[1.4,1.9],[1.7,8.9],[2,4.1],
    [3.3,7],[3.5,1.5],[4,6.3],[4.4,1.9],[5.7,2.9],[6,7.1]
])
y = np.array([0]*6 + [1]*6)
```

```

model = GaussianNB()
model.fit(X, y)

test_point = np.array([[2, 2]])
pred = model.predict(test_point)[0]

# Plot
plt.scatter(X[y==0][:,0], X[y==0][:,1], color='blue', label="Class 0")
plt.scatter(X[y==1][:,0], X[y==1][:,1], color='red', label="Class 1")
plt.scatter(2, 2, color='green', s=200, label="Test Point")

plt.legend()
plt.title(f"Gaussian Naive Bayes Prediction → Class {pred}")
plt.xlabel("X")
plt.ylabel("Y")
plt.show()

```

Output

