Batch T4

Practical No. 4

Title of Assignment :

Design multivariate classifiers from first principles.

Student Name: Parshwa Herwade

Student PRN: 22510064

Assignment 4

Objec ve of this assignment is to design mul variate classifiers from first principles. You may choose to either extend the univariate methods from assignment 1, for the mul variate scenarios or design new approaches. However, do not use any off the shelf classifica on algorithms.

Please note that this assignment does not require you to submit the code. However, you may choose to implement your methods to explore concepts.

The task of your classifier is gender iden fica on, based on measured parameters. We have a toy data set of 1000 male and 1000 female (labeled) samples. We a empt a height based univariate classifier and realize that due to overlap in heights, there are limita on on improving accuracy meaningfully without exploring other features. So, we decide to iden fy and add new features to our learning algorithms to reduce the predic on error further.

Consider following scenarios of increasing complexity.

1. Uncorrelated input features (5 marks)

a. We have two input features say, Height(in cm) and Heamoglobin levels measured for all 2000 samples. Let's assume that both these features are normally distributed within each gender. These features are pre y much uncorrelated within each gender.

b. Can you design approaches to train a classifica on algorithms to predict gender?

2. Input features with non-zero correla on (3 marks)

a. In this scenario, we have two input features say, Height(in cm) and weight(in kg) measured for all 2000 samples. Both are normally distributed within each gender. The correla on between these features is 0.6 within each gender.

b. Which of the algorithms you designed for uncorrelated features would work as is ? If they don't, what changes can you make to your algorithms to accommodate correlations

3. How far can we go? (2 marks)

a. We observe that accuracy improves with addi on of one new feature in both of the above scenario. Can we reach a conclusion that accuracy can be improved further by adding mul ple such features to the input? How many such features would you add in your quest to improve accuracy? Would addi on of new features require any changes to the experimental set up?

ANSWERS:

In assign 1, I built a classifier using only height. Although that approach showed some promise, the overlap in height distributions between males and females limited its accuracy. I realized that by adding another feature or even more features I could capture additional differences between the genders. In assign 4, I have to design multivariate classifiers from first principles without using any off-the-shelf algorithms.

Task : One where the additional feature is uncorrelated with height (using Haemoglobin levels) and another where the features are correlated (using Weight). Also the implications are discussed further

1. Classifier for Uncorrelated Features (Height and Haemoglobin)

I chose Haemoglobin because it's a physiological measurement that might vary between genders and under the assumption it's independent from height can be combined with height straightforward .

(Parameter)

For each gender, I estimated the mean and standard deviation for Height,

The mean and standard deviation for haemoglobin,

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

$\mu =$ Mean

$\sigma =$ Standard Deviation

$\pi \approx 3.14159$

$e \approx 2.71828$

Source: Google

SOURCE: https://medium.com/analytics-vidhya/normal-distribution-and-machine-learning-ec9d3ca05070

(Calculation) since I assumed height and haemoglobin are independent, the joint likelihood for an observation.

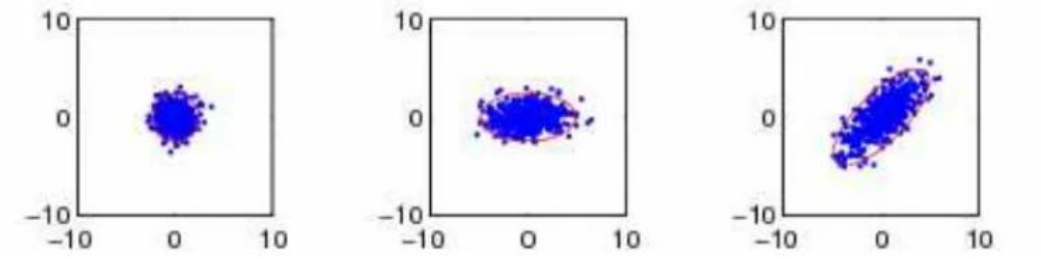
I assumed equal prior probabilities for both genders. For each new observation, I computed the likelihood for both male and female classes. I then assigned the observation to the gender with the higher likelihood. This simple maximum likelihood decision rule helped me make a clear classification based on the combined evidence from both features.

2. Classifier for Correlated Features (Height and Weight)

Height and weight are known to be positively correlated in most populations. In this scenario, the independence assumption does not hold, so I needed to account for their interaction.

(Parameter Estimation) for each gender:

SOURCE:
https://medium.com/swlh/understanding-gaussian-classifier-6c9f3452358f



$$\Sigma = \begin{pmatrix} \sigma^2 & 0 \\ 0 & \sigma^2 \end{pmatrix} \qquad \Sigma = \begin{pmatrix} \sigma_x^2 & 0 \\ 0 & \sigma_y^2 \end{pmatrix} \qquad \Sigma = \begin{pmatrix} \sigma_x^2 & \rho\sigma_x\sigma_y \\ \rho\sigma_x\sigma_y & \sigma_y^2 \end{pmatrix}$$

Kevin Murphy's slide

$$\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \Sigma) \stackrel{\text{def}}{=} \frac{1}{(2\pi)^{p/2}|\Sigma|^{1/2}} \exp[-\tfrac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T \Sigma^{-1}(\mathbf{x}-\boldsymbol{\mu})]$$

This formulation accounts for the covariance between height and weight. It's a bit more complex, but it accurately models the dependency between the two features.

As before, I computed the likelihoods for both male and female classes and assigned the observation to the gender with the higher likelihood. This approach directly incorporates the interdependency of height and weight into the classification decision.

In assign 1, I only had to worry about one feature, so the math was simpler. Here, by including a covariance matrix, I learned how to handle more realistic data where features are not independent. This extra step should lead to better performance in scenarios where height and weight are naturally correlated.

3. Extending to More Features

Once I saw improvements with two features, the next logical step was to consider adding even more features, such as blood pressure, body mass index, or other biological markers.

(Pros) increased Accuracy: Each additional, informative feature can provide extra evidence, improving classification accuracy.

Better Representation: More features can capture more aspects of the differences between genders.

(Cons)

Overfitting: With too many features and limited data, the model might start to learn noise rather than the actual underlying patterns.

Curse of Dimensionality: As the number of features increases, the data becomes sparser. This means that distances between points become less meaningful, making classification harder.

Mitigation Strategies:

If I add too many features, I'd consider using dimensionality reduction techniques like PCA to reduce the number of variables while keeping the essential information. This step is necessary to keep the model robust and to avoid the pitfalls of overfitting and sparsity.

In assign 1, I experimented with threshold-based, probability-based, and quantized methods using just height. Those methods were simple and provided a baseline. With assign 4, I'm extending those ideas to multivariate settings. The challenges I encountered with a single feature (like overlap and limited discrimination) are tackled by incorporating more dimensions. However, I also learned that more isn't always better unless managed correctly (hence, the need for techniques like PCA when facing high-dimensional data).

SOURCE: https://www.datacamp.com/blog/curse-of-dimensionality-machine-learning

CODE:

```
import numpy as np

from scipy.stats import norm, multivariate_normal


np.random.seed(42)

n = 1000


# Uncorrelated Features (Height and Haemoglobin)

height_f = np.random.normal(152, 5, n)

hb_f = np.random.normal(13, 1, n)

height_m = np.random.normal(166, 5, n)

hb_m = np.random.normal(14, 1, n)


mean_height_f = np.mean(height_f)

std_height_f = np.std(height_f)

mean_hb_f = np.mean(hb_f)

std_hb_f = np.std(hb_f)


mean_height_m = np.mean(height_m)

std_height_m = np.std(height_m)

mean_hb_m = np.mean(hb_m)

std_hb_m = np.std(hb_m)
```

```
def likelihood_uncorrelated(h, hb, gender):
    if gender == 'F':
        lh = norm.pdf(h, loc=mean_height_f, scale=std_height_f)
        lhb = norm.pdf(hb, loc=mean_hb_f, scale=std_hb_f)
    else:
        lh = norm.pdf(h, loc=mean_height_m, scale=std_height_m)
        lhb = norm.pdf(hb, loc=mean_hb_m, scale=std_hb_m)
    return lh * lhb

correct_uncorr = 0
for i in range(n):
    if likelihood_uncorrelated(height_f[i], hb_f[i], 'F') > likelihood_uncorrelated(height_f[i], hb_f[i], 'M'):
        correct_uncorr += 1
for i in range(n):
    if likelihood_uncorrelated(height_m[i], hb_m[i], 'M') > likelihood_uncorrelated(height_m[i], hb_m[i], 'F'):
        correct_uncorr += 1
accuracy_uncorr = correct_uncorr / (2*n)
print("Uncorrelated Features Accuracy: {:.2f}%".format(accuracy_uncorr*100))

# Correlated Features (Height and Weight)
def generate_correlated(mean_h, std_h, mean_w, std_w, corr, n):
    cov_hw = corr * std_h * std_w
    cov_matrix = [[std_h**2, cov_hw], [cov_hw, std_w**2]]
    means = [mean_h, mean_w]
    return np.random.multivariate_normal(means, cov_matrix, n)
```

```
female_corr = generate_correlated(152, 5, 60, 7, 0.6, n)
male_corr = generate_correlated(166, 5, 70, 7, 0.6, n)

mean_vec_f = np.mean(female_corr, axis=0)
cov_f = np.cov(female_corr, rowvar=False)
mean_vec_m = np.mean(male_corr, axis=0)
cov_m = np.cov(male_corr, rowvar=False)

def likelihood_correlated(x, gender):
    if gender == 'F':
        return multivariate_normal.pdf(x, mean=mean_vec_f, cov=cov_f)
    else:
        return multivariate_normal.pdf(x, mean=mean_vec_m, cov=cov_m)

correct_corr = 0
for i in range(n):
    x = female_corr[i]
    if likelihood_correlated(x, 'F') > likelihood_correlated(x, 'M'):
        correct_corr += 1
for i in range(n):
    x = male_corr[i]
    if likelihood_correlated(x, 'M') > likelihood_correlated(x, 'F'):
        correct_corr += 1
accuracy_corr = correct_corr / (2*n)
print("Correlated Features Accuracy: {:.2f}%".format(accuracy_corr*100))
```