CS 4780/6780 Final Project

Qianyun Chen, Hsiu Yuan Fan, Mike Fanor

***Abstract*—This project is to find a method to predict whether the bacteria has high resistance or low resistance against two medicines - carbenicillin and tobramycin based on bacteria's images. In the training data, the resistance of carbenicillin and tobramycin and the bacteria's image are given. In this project, we also testify which feature of the image could make the prediction more accurate.**

I. Introduction

The procedure of our experiment is:

1) Label the Data: We analyzed the data and labeled the data which are positive as 1 and the data which are zero or negative as 0. In our settings, the positive numbers means that bacterium have strong resistance and non-positive numbers means that bacterium have weak resistance.
2) Image Processing: We cropped images so that they could have the same size and the bacteria is located at the center. We also set images gray-scale when it is necessary.
3) Extract Features: In order to classify the images, we decided to extract the features from images and compare them with different method. In this experiment, we extract the images' RGB values and symmetry values as their features.
4) Test different methods of classifying: In this experiment, we tried K-Nearest Neighbors algorithm, Linear Regression algorithm, and Logistic Regression algorithm.

II. Data

- We used 80% data as training data and 20% as out testing data.
- The training data filename is $Training\_Data.xlsx$
- The filename of the training data for carbenicillin is $carb\_data.csv$
- The filename of the training data for tobramycin is $toby\_data.csv$
- The training data with features' values filename is $Training\_Data\ symmetryFan.xlsx$

III. Methods

Here is the method to extract RGB values:

- There are two steps of this method. First of all, we crop all images to be a rectangle which size is as big as the bacterial, so we can reduce the background noise. Secondly, after we crop the images, we have to resize the image to the same pixels(5000*5000) and calculate the total RGB value of this image by dividing the total number of the total pixels. Finally, we can get the mean RGB value to be our feature.
  In short: 1. Cropping the images. 2. Resizing the images to the same pixels. 3. Calculating the average RGB value.

Here is the method to extract symmetry values:

- There are three steps of this method. In the beginning, we still need to pre-execute these images like getting RGB value method. Then, we transfer all RGB value to gray value, so we can simplify next step by only calculating one value instead of three. Secondly, we select the down part image or pixels and flip it up to subtract the up part pixels to get the symmetry values. Finally, we summarize the each pixels value and dividing the total number of the total pixels number.

IV. Experiments

- Experiment I:
  Test with K-Nearest Neighbor algorithm. We separated the data set into two files with different medicines.
  Code By Matlab.
  Code can be changed with different inputs. We changed the inputs for different medicines and control features such as testing with RGB values only, Symmetry values only, or both values.
  Code Filename: $KNN.m$
  Result Filename: $KNN\_toby.xls$, $KNN\_carb.xls$
  Trial 1: We use training images to clustering in two groups based on the features: RGB values and Symmetry values. The algorithm will calculate the distance between each images according to RGB values and Symmetry values and classify.
  Trail 2: We use training images to clustering in two groups based on the feature: RGB values only.
  Trial 3: We use training images to clustering in two groups based on the feature: Symmetry values only.
- Experiment II:
  Test with Linear Regression algorithm. We separated the data set into two files with different medicines. Code can be changed with different inputs. We changed the inputs for different medicines and control features such as testing with RGB values only, Symmetry values only, or both values.
  Code By Matlab.
  Code Filename: $Linear\_R.m$
  Result Filename: Result with RGB values and symmetry values as features: $Linear\_grayandRGB\_toby.xls$ $Linear\_grayandRGB\_carb.xlsx$

Result with RGB values only $Linear\_RGB\_carb.xlsx$ $Linear\_RGBonly\_toby.xls$

Result with symmetry values only: $Linear\_grayonly\_toby.xls$ $Linear\_gray\_carb.xlsx$

Here is the equations for the algorithm:

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} \tag{1}$$

$$\boldsymbol{\beta} \;=\; (\mathbf{X}^{\mathsf{T}}\mathbf{X})^{-1}\mathbf{X}^{\mathsf{T}}\mathbf{y} \tag{2}$$

$$\mathbf{y}_{new} = X_{new}\boldsymbol{\beta} \tag{3}$$

The algorithm will calculate $\boldsymbol{\beta}$ based on the training data and it will use the $\boldsymbol{\beta}$ to calculate $\mathbf{y}\_new$. From step 3, we got non-integer numbers. So we choose different standards to test data to be labeled as zero or one. If the result greater than the standard, it is labeled as 1. If the result smaller than the standard, it is labeled as 0.

Trial 1: We calculate $\boldsymbol{\beta}$ based on the features: RGB values and Symmetry values.

Trail 2: We calculate $\boldsymbol{\beta}$ based on the features: RGB values only.

Trial 3:We calculate $\boldsymbol{\beta}$ based on the features: Symmetry values only.

- Experiment III:

  Test with Logistic Regression algorithm. We separated the data set into two files with different medicines. Code can be changed with different inputs. We changed the inputs for different medicines and control features such as testing with RGB values only, Symmetry values only, or both values.

  Code By Python.

  Code Filename: $Logistic\_Regression.txt$ $predictions.py$

  Result Filename: Result with RGB values and symmetry values as features: $carb\_logistic.xlsx$ $toby\_logistic.xlsx$

  Here is the equations for the algorithm:

$$\mathrm{P}(Result = 1 \mid \mathbf{X}) = \frac{1}{1 + e^{-\boldsymbol{\beta}^{\mathsf{T}}\mathbf{X}}} \tag{4}$$

$$\boldsymbol{\beta}^{*} = argmax\ \ell(\boldsymbol{\beta} \mid Y, \mathbf{X}) \tag{5}$$

The algorithm will calculate $\boldsymbol{\beta}$ based on the training data and it will use the $\boldsymbol{\beta}$ to calculate the probability of the result is one. This algorithm needs to use likelihood to determine the probability that a parameter was the one that generated a sample $\mathbf{X}$.

V. Summerize

Summarize with Features as RGB values and Symmetry Values:

Method	Antibiotic 1	Antibiotic 2
Linear Regression	0.94	**1**
K Nearest Neighbor	0.82	**1**
Logistic Regression	0.88	**1**

TABLE I: Accuracy with different algorithms

Method	RGB only	Symmetry only	RGB and Symmetry
Linear Regression	**1**	0.75	0.94
K Nearest Neighbor	0.8	**0.85**	0.82

TABLE II: Accuracy with different features in Carb

Method	RGB only	Symmetry only	RGB and Symmetry
Linear Regression	**1**	0.88	**1**
K Nearest Neighbor	0.94	0.98	**1**

TABLE III: Accuracy with different features in Toby

VI. Conclusions

In this experiment, We compared linear regression algorithm, K Nearest Neighbor algorithm, and logistic regression algorithm. From the result, linear regression and logistic regression work better for the accuracy. We also used control variate method to examine which features has strong influence. From the result, we conclude that RGB values has a slightly stronger influence than the features for classifying. We think that there may be some errors while testing algorithms such as rounding error. From the training data set, We think that the accuracy is not very useful for tobramycin since 97% of the data is classified as strong resistance. For this experiment, we only found that RGB values and Symmetry values are important features. For the future, we think that there could be more features extracted from images and we could testify more algorithms with larger data-set to find the best algorithm to predict images' information more accurately.

References

[1] Daniel L. Pimentel-Alarcon
https://danielpimentel.github.io/teaching/CS6780/lectures
Topic 6: Linear Regression Topic 7: Nearest Neighbors Topic 9: Logistic Regression