HW2 SPR Fall 2022

Deadline 26 Aban

In this homework you have to implement Multiclass Classification, Softmax and Naïve Bayes.

Multiclass Classification

Dataset: 'seed.txt'

- The Seed dataset consists of 7 features and 3 classes. Use all features and classes for this part of homework.
- In before, you used logistic regression for binary classification. In this part you should use whole seed dataset for multiclass classification (one-vs.-one and one-vs.-all) by logistic regression.

- Your task is to train your model on the given dataset and test your implementation on the test part.
- For implementing this part read the ‘seed.txt’ file, Consider the first 80% of the data for train and 20% for test (You can use libraries for splitting your data).
- Train multiclass classification (one-vs.-one and one-vs.-all) by logistic regression and report train and test accuracy for both of method.
- Plot cost function for enough iteration for one-vs.-all and one-vs.-one method and report convergence iteration in this method.
- Describe how many calls to binary classification are made in each technique.

- Use ***softmax*** regression and compare it with the previous part.
- Train multiclass classification by softmax regression and report train and test accuracy.
- What method (one-vs.-one, one-vs.-all or softmax) has worked best?

Naïve Bayes

Dataset: 'digitdata.zip'

- Handwriting recognition is the ability of a computer to interpret hand written text as the characters. In this assignment we will be trying to recognize numbers from images. To accomplish this task, we will be using a Naïve Bayes classifier.
- Naïve Bayes classifiers are a family of probabilistic classifiers that are based on Bayes' theorem. These algorithms work by combining the probabilities that an instance belongs to a class based on the value of a set of features. In this case we will be testing if images belong to the class of the digits 0 to 9 based on the state of the pixels in the images.

- We are also providing data files representing handwriting images along with the correct classification of these images. These are provided as a zip file.
- This zip file contains the following:
  - readme.txt - description of the files in the zip
  - testimages - 1000 images to test
  - testlabels - the correct class for each test image
  - trainingimages - 5000 images for training
  - traininglabels - the correct class for each training image
- Your task is to train your model on the given dataset and execute your implementation on the test part.
- The image data for this consist of 28 × 28 pixel images stored as **text**. Each image has 28 lines of text with each line containing 28 characters. The values of the pixels are encoded as ' ' for white and '+' for gray and finally '#' for black. We will be classifying images of the digits 0-9, meaning that given a picture of a number we should be able to accurately label which number it is.
- We will be using these images as a set of binary features to be used by our classifier. Since the image is 28 × 28 pixels we will have 28 × 28 = 784 features for each input image. To produce binary features from this data we will treat pixel i, j as a feature and say that F(i,j) has a value of 0 if it is a white(' ') pixel, and 1 if it is grey or black('#' or '+'). In this case, we are generalizing the two values of gray and black as the same to simplify our task.
- From this, we get two phases of the algorithm.
  - **Training** - compute the conditional probability that for each feature an image is part of each class.
  - **Classifying** - for each image compute the class that it is most likely to be a member of given the assumed independent probabilities computed in the training stage. To classify the unknown images using the trained model you will perform maximum a posteriori (MAP) classification of the test data using the trained model.

Continue

- If you need, you should use **Laplace Smoothing** as a part of your job. Play around with the Laplace smoothing factor to figure out how to get the best performance out of your classifier.
- If you multiply many small probabilities you may run into problems with numeric precision, what is the problem? To handle it, we recommend that you compute the logarithms of the probabilities instead of the probabilities. Explain why this approach can help?
- Report your confusion matrix. This is a 10 × 10 matrix whose entry in row r and column c is the percentage of test images from class r that are classified as class c.
- Report the accuracy, precision, recall, and F1 score of the model on the test part.
- What is the base assumption of the Naïve Bayes classifier? Why is it important?

- Bonus:

- Use ternary features (by taking into account the two foreground values to see if classification accuracy improves.

✓ Pay extra attention to the due date. It will not extend.

✓ Be advised that submissions after the deadline would not grade.

✓ Prepare your full report in PDF format and include the figures and results.

✓ Do not use sklearn or any similar library and write your own code.

✓ Use only the python programming languages.

✓ Submit your assignment using a zipped file with the name of:
“FirstName_LastName_ TeammateFirstName _Teammate LastName.zip”

✓ Email your zip file to ‘melikaaa9896@gmail.com’.

✓ Using other students’ codes or the codes available on the internet will lead to zero grades.