SCC361 Artificial Intelligence

Coursework 1 – Data Preparation and Classification

Date Released: Friday, 22nd October 2021

Date Due: Friday 19th November 2021

This coursework is worth 20 marks

Introduction

In this coursework, you are required to solve a problem in Machine Learning and Data Preparation. You can choose to tackle a problem in either:

1. Computer Vision: Traffic Sign Recognition
2. Natural Language Processing: Sentiment Analysis

You may attempt both problems if you wish, but **please only submit one**. For each problem, the data will need to be prepared, feature and label arrays will need to created for training and testing, and models will need to be trained and evaluated.

Option 1: Traffic Sign Recognition from Images

This problem involves the automatic classification of images of speed signs. Your task in this project is to study the dataset, prepare it for machine learning, and to select the best classification model for automatically **determining the speed shown in the image**. This is a classification task and will require a supervised learning approach.

Dataset

The dataset for this problem is contained in the TrafficSignData.zip file. This zip file contains 2 folders:

1. "Train" contains the images that will be used to train your model.
2. "Test" contains the images that will be used to test and evaluate your model.

Both of these folders contain 4 folders with the names "40", "80", "90", "100", which contain the images showing speed limit signs for 40, 80, 90 100. The images are colour and 64x64 pixels in size.

Data Preparation

In order to use the images in your method efficiently, you will need to import each image into MATLAB, convert it to a double data type once imported, convert it to grayscale, and stack the columns to form a vector. I.e. your vector will take the form:

[column 1; column 2; column 3; …].

This will then need to be transposed to form a row vector.

Features and Labels

You will need to create one feature matrix for training with m_1 rows and n columns and one label vector for training with m_1 rows, where:

- m_1 is the number of training images
- n is the number of pixels the images, i.e. $n = 64 \times 64$

You will need to create one feature matrix for testing with m_2 rows and n columns and one label vector for training with m_2 rows, where:

- m_2 is the number of testing images
- n is the number of pixels the images, i.e. $n = 64 \times 64$

Model Training and Evaluation

Please see the Model Training and Evaluation section below.

Option 2: Sentiment Analysis from Text

This problem involves the automatic classification of sentiment from recorded tweets from Twitter. Your task in this project is to study the dataset, prepare it for machine learning, and to select the best classification model for automatically **determining the sentiment displayed by the tweet**. This is a classification task and will require a supervised learning approach.

Dataset

The dataset for this problem is contained in the SentimentAnalysisFata.zip file. This zip file contains 1 csv file: "text_emotion_data.csv". The csv file contains two columns:

1. "sentiment": this column lists one of four sentiments ("relief", "happiness", "surprise", "enthusiasm") that have been assigned to the corresponding tweet.
2. "Content": this column contains the text of the tweet

There are 8651 entries in this file.

Data Preparation

In order to use the tweets and their labels, you will need to import the csv file into MATLAB as a table, build a Bag of Words containing all of the tokenised tweets, remove stop words, remove any words with fewer than 100 occurrences in the bag (you can also try varying this number but it is not essential for this task), and build the full Term Frequency-Inverse Document Frequency matrix (tf-idf) for the resulting bag. You will also need to build a corresponding label vector from the column of sentiments.

Features and Labels

You will need to create one feature matrix for training by selecting the first 6921 rows of the tf-idf matrix and all columns. You will also need to create a corresponding label vector with the first 6921 labels.

You will need to create one feature matrix for testing by selecting all rows of the tf-idf matrix after row 6921 (i.e. the remaining rows). You will also need to create a corresponding label vector.

Model Training and Evaluation

Please see the Model Training and Evaluation section below.

Model Training and Evaluation

Model Training

You will train and compare **any three** classification algorithms implemented in MATLAB. It is sufficient for this to use their default parameters but you are welcome to optimise these parameters to improve performance.

Table 1 shows some examples of classification algorithms in MATLAB and their function names as implemented in MATLAB's Statistics and Machine Learning Toolbox™. You may find it useful to have a look at this MATLAB documentation page on Supervised Learning Workflow and Algorithms for some ideas on how to structure your solution and select your algorithms. You are encouraged to make use of the suggestions on this page but feel free to explore further.

Table 1: MATLAB Classification of Algorithms and their function Names

Algorithm	**Function**
K-Nearest Neighbour	fitcknn()
SVM for Multiclass	fitcecoc()
Decision Tree	fitctree()
Naïve Bayes	fitcnb()
Discriminant Analysis	fitcdiscr()
Ensembles	fitcensemble()

To train a machine learning model, you need to use the syntax:

```
>> model = function(features,label)
>> predictions = predict(model,features)
```

For example, you might try:

```
>> knnmodel = fitcknn(training_features,training_label)
>> predictions = predict(knnmodel,testing_features)
```

Evaluation

Accuracy is measured by comparing the predictions from your models to the test labels in the dataset. It will be sufficient to calculate the accuracy as

$$accuracy = \frac{number\ of\ correct\ predictions}{total\ number\ of\ labels}$$

You should also investigate the results more thoroughly by analysing the resulting confusion matrix. You can obtain this with the `confusionchart` function. Type `help confusionchart` in MATLAB if you need help with this.

Submission and Marking

Please submit the following on Moodle on or before the **deadline of 4pm on Friday, 19th November, 2021**.

1. Your MATLAB your code (10mks)
    a. running without any errors
    b. Structured
    c. Documentation/ detailed comments
2. A short (max 4 page) report describing (10mks):
    a. the content of the dataset
    b. the data preparation method
    c. the model training and evaluation methods used
    d. the selected model and criteria
    e. error analysis and/or other observations