HERIOT WATT UNIVERSITY

DATA MINING AND MACHINE LEARNING

COURSEWORK 1A

Emotion Recognition Using KNN (Scipy Lib) in Python

Mohit VAISHNAV, *VIBOT*
Malav BATERIWALA, *VIBOT*
Makenzy ABASS GUMAH

Supervisor
Dr. Diana BENTAL

Submission Last Date 16th, Oct.

Contents

List of Tables

List of Figures

1 Introduction

Data mining is the process of discovering patterns in large data sets involving methods at the intersection of machine learning, statistics, and database systems. Data mining is an interdisciplinary sub field of computer science with an overall goal to extract information from a data set and transform the information into a comprehensible structure for further use. In this assignment, the task is to analysis the data given in form a dataset of emotions. The emotions file given are in form of: Anger, Neutral, Disgust, Fear, Happy, Sad and Surprise. The dataset consists of 35887 images and its 48X48 pixels in grey-scale. Looking at the image, most of the region is covered by the face based on a emotion described above. There are 7 more files given , which consists the same dataset but with the labels and classification of every emotion individually.

2 Data Conversion, Randomization and Reducing constraints

Data Prepossessing is a very important task in data mining and machine learning. If the data is not optimized and if the noise is too much the model generated might not be accurate and will lead to bad cases. It is said that , if the pre processing is done well than the results generated will vary a lot from the non-processed one in favor of processed data. Following are the steps taken for data prepossessing.

- **Reading data** : First Step is to read the data-set provided and the procedure to do that in Python is using Pandas library. As the given data set is in CSV format, *'read_csv'* Command is used to read the data set. Pandas library has many in-build function which allows the user to select the type of format of the data set and it converts it into a table format for easy visualization and accessibility for further processing. The shape of the dataset is (35887,2) which means that there are 35887 rows and 2 columns , with first column being the images and the second column consisting of the pixels respectively.

- **Randomization** : Data Randomization is required because the data set is unknown. Shuffling the data set takes care of the data being in a particular order. This step is very important due to the fact that data set will be split into 80-20% ratio for training and testing purposes and if the data set is not shuffled well the training will not be effective and will cause the model to give bad results. Here *'sample'* function of pandas is used to perform this task and this will shuffle the rows in the whole data set.

- **Reducing Constraints** : The attributes of each image are equal to 2304 and the total number of images are equal to 35887. So the total number of attributes in for the whole data set will be 82683648. Here in our cases the attributes are quite less for a good computational systems. But if the data set is 64X64 , then the attributes will be more and there will be need of reducing the constraints , otherwise the model will be extremely computational expensive.

3 Classification

For a data there can be 2 possibilities and they are Classification and Regression. Classification is a process to differentiate between classes for any given set. If the dataset contains 10 classes , then with the help of a model designed it is possible to get a label for any new data among those 10 classes. There are many ways of classification like : SVM (Support Vector Machines) , Decision Tress, Bayes Method, Random Forest, K-Nearest Neighbour, K-Means etc. All these techniques have their own methods and algorithms to reach the conclusion for any type of data. For this assignment, the method used for classification is K-Nearest Neighbour.

K-Nearest Neighbour is a classification or regression algorithm depending upon the algorithm progression. But the concept of both is the same. In this the first task to decide the value of K. K is the term defined that takes in account the neighbour that are surrounding it and adds them within it boundary. That boundary is defined by keeping a specific distance according to the user. The value of K is very important in this algorithm. With the help of the Euclidean distance , the nearest neighbours are decided. Then they are arranged in order. So if a new data comes they are compared with the features which were extracted with the algorithm. The closest matching features will be the answer and the new data will be given the label of that feature.

The process to find K-Nearest Neighbour is as follows:

1. The pre-processed data is taken in the shape of 2 dim and the image is flatten. So the input of the system will be of shape (35887,2). But before that, the data needs to split into test and training set using the *train_test_split* function of Sklearn library. Here the data is split into 80-20 ratio, training set being 80%.

2. So the input to the K-NN function will be the training set along with its label. For calculation the function *K_NearestNeighbour* of Sklearn module is used and the value of K is taken as 3.

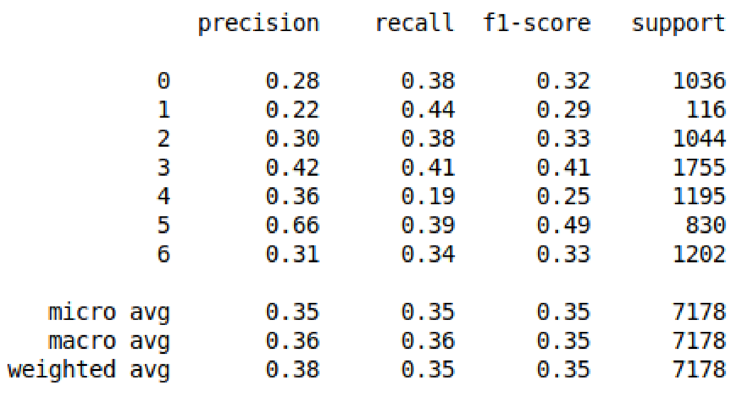
3. After calculation of the K-NN , we need to fit it with the data given to us. So it will apply the decision boundary according to the pixels given to it. This step is used to calculate the data set with the Euclidean distance along with the labels given.

4. Then the prediction step takes place, which is used to the predict the model with the test set taken from splitting the original set. For the precision and accuracy , confusion matrix is created using the function *Confusion_Matrix* and then creating a report for it. Fig.1 shows the confusion matrix.

5. With the help of classification report, the accuracy can be approximated. And in this case it comes around 36% on average of the given classes. For some classes it perform really well but for the some classes the precision is bad.

4 Deeper Analysis of data

Now as we have predicted the data for every class using the main CSV file, the next step is to have a deeper analysis of data and find different ways to have better results. So the other CSV files are loaded in the system, which has every classes differently and has the binary classification of each images. Now by taking the top 10 attributes of each image belonging to a class, K-NN will be carried out. By doing this, only the best features are taken in consideration and the accuracy of the model should improve. As removing the unwanted data helps in filtering out the extra features which do not



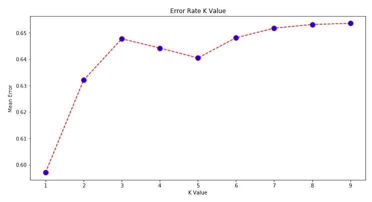
```
              precision    recall  f1-score   support

           0       0.28      0.38      0.32      1036
           1       0.22      0.44      0.29       116
           2       0.30      0.38      0.33      1044
           3       0.42      0.41      0.41      1755
           4       0.36      0.19      0.25      1195
           5       0.66      0.39      0.49       830
           6       0.31      0.34      0.33      1202

   micro avg       0.35      0.35      0.35      7178
   macro avg       0.36      0.36      0.35      7178
weighted avg       0.38      0.35      0.35      7178
```

Figure 1: Confusion Matrix Report with K=3

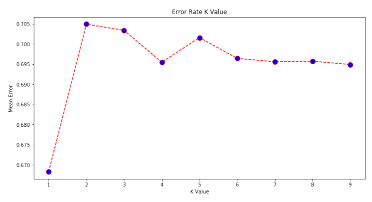
provide any weight in the classification. Not only that , the computational time of the system will also be reduced by some factor.

5 Improving Classification, based on Analysis

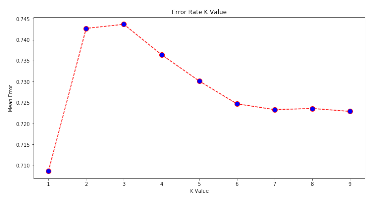
Now for improving the classification , the next step taken is only taking the attributes which corresponds to the best feature for every different emotions. So with the help of every dataset of every emotion, we take the 2,5, 10 top features from each emotions. This process is done calculating the correlation between the each attribute and emotion. Then the error is calculated for every attribute. Each pixel of an image is considered as an attribute. By getting the index of that pixel we get the attribute which corresponds to the best feature for that particular emotion. This is done for all the images for every emotions. Then by saving these pixel location , K-NN classification is carried out, which gives results with the test set. Figures for the case of 2 and 5 pixel are as below in Fig.2.



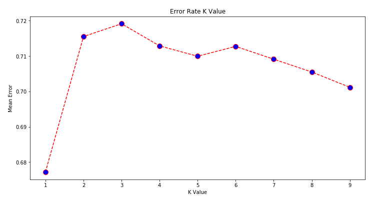
(a) Error rate with change in value of K with all attributes



(b) Error rate with change in value of K with top 10 attributes



(c) Error rate with change in value of K with top 2 attributes



(d) Error rate with change in value of K with top 5 attributes

Figure 2: Comparison of Error rate with change in Number of attributes used

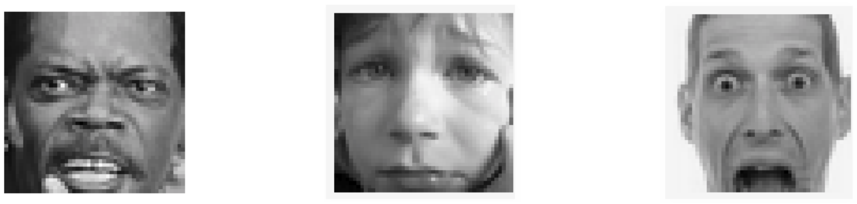

(a) Gray Image 1 (b) Gray Image 2 (c) Gray Image 3

Figure 3: Visualizing different Emotions

6 Conclusion

This assignment aim to help understand that basics of data mining and machine learning. Here a basic algorithm is implemented to see how the data can be manipulated and it can affect the result. Everything affects in some way in the algorithm and the model. Starting from randomization to how the parameters are shifted and adjusted. With running the randomization of data again we get some changes in the final accuracy. It makes lot of difference in some cases. Then the error in all the figure calculated shows how the value of K in K-NN affects the results. Not only that, with the change in attribute taken into consideration also changes the overall accuracy of the algorithm which can be seen in the confusion matrix created in the Ipython notebook.

7 Research Question

In this coursework biggest learning is opting for the Python instead using the Weka Platform. Conceptual understanding took a bit of time to know what to do and how to go ahead but after having few meetings with the faculty advisor this problem could be tackled. Data received is in the form of *csv* file where all the attributes are stored in the form of object. But for processing the data in python it has to be converted to the desired format. In this implementation, data has to be converted in different formats as per the requirements. Like for visualization of the image, data is converted in numpy readable format, for passing the data in knn learning algorithm it requires the data to be as a list. Hence all the conversion are done in the required places.

Next task is to implement *correlation* function. For the implementation of correlation amongst various available correlation functions available in literature, pearson's [7] coefficient was chosen which works on the linear relation. This has a advantage of reduced complexity and faster implementation along side other. For the clearer visualization of data, *seaborn* library is used. All the pixels and their location plays an important role in deciding which attribute has to be selected and which are to be rejected. Hence, we decided to consider each pixel as an attribute and find its correlation with the emotion. Similarly this is done for each pixel and from where top correlating values are searched along with their location. Each pixel of that location is considered for all the rows (images) and just those values are used for KNN classification purpose. For the experimental purpose, number of attributes selected are in range of 2,5, 10, 15 & 25 from each emotions. Important observation doing this is the precision value is same as that obtained using all the attributes. Hence it can be said that there is no use of all the attributes for the classification because even with 175 (25 $\times 7$ same precision is obtained. Another observation can be quoted is about the error analysis varying with number of cluster size (ie. from 1 - 9). It can be seen that not always cluster size 3 works the best way. Even from the end precision (Table. 1 value this observation is strengthened.

Moving forward, analysis of the dataset is done. It is very uneven with *Disgust* having the least count and *Happy* fifteen times more than the least as seen in Fig.4. Probably this is one aspect where improvement has to be done by managing to obtain the equal dataset for all kind of emotions. Here an effort was also made to visualize those images for which python program is created and the output can be seen as in Fig.3.

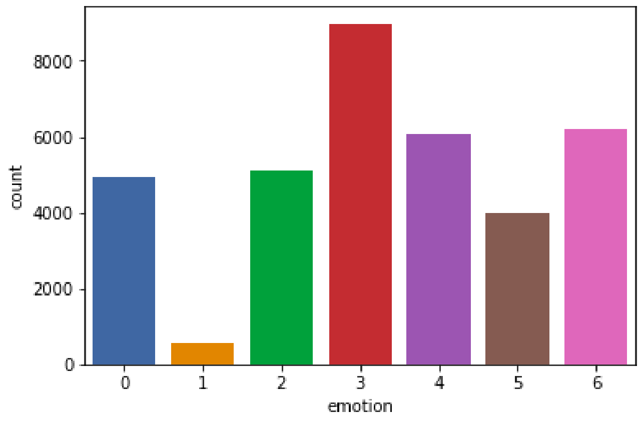
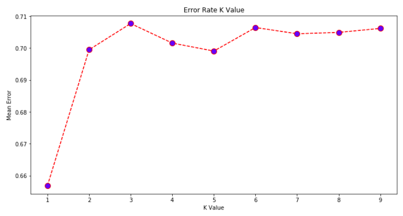


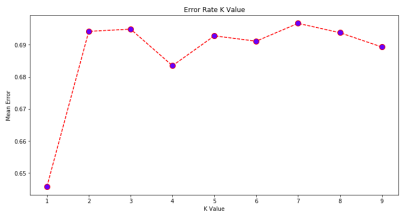
Figure 4: Data Divisibility between emotions

In this implementation, results are analyzed using the confusion matrix which also provides values of precision and recall. When all the attributes are used for the KNN prediction, greatest accuracy of 37% is achieved with 3 as nearest neighbours. Further the analysis is done by changing the number of nearest neighbours as seen in Graphs represented in Fig.5.

In this experiment different number of attributes are used for the purpose of classification using KNN. Various values used are 2, 5, 10, 15 & 25. Though this course work requires only till 10 which means from each emotion data, top 10 attributes are selected amounting to total of 70 instead of 2304 for the prediction purpose. Here the precision achieved on an average is about 32% (K = 3) and 33% in maximum. Similarly we thought of moving ahead to find that threshold where the average precision of top attributes moves as close to using all 2304 attributes. Table.1 shows the comparative analysis of all the simulations performed. So it can be safely said that there is no need to use all the data available but using just a few variables accuracy achieved is same as that of using all the data available.



(a) Error analysis for Top 15 attributes



(b) Error analysis for Top 25 attributes

Figure 5: Further Features

Attributes	K = 1	K = 2	K = 3	K = 4	K = 5	K = 6	K = 7	K = 8	K = 9
2	0.29	0.31	0.27	0.27	0.27	0.27	0.27	0.27	0.27
5	0.32	0.31	0.30	0.29	0.29	0.28	0.29	0.29	0.29
10	0.33	0.31	0.31	0.31	0.30	0.30	0.30	0.30	0.30
15	0.34	0.32	0.30	0.30	0.29	0.29	0.29	0.29	0.29
25	0.35	0.32	0.32	0.32	0.30	0.31	0.30	0.30	0.30

Table 1: Mean Accuracy with different Vocab Size

References

[1] `https://stackoverflow.com/questions/13070461/get-index-of-the-top-n-values-of-a-list-in-python`

[2] `https://stackoverflow.com/questions/16597265/appending-to-an-empty-data-frame-in-pandas`

[3] `https://www.quora.com/How-is-a-Pandas-DataFrame-different-from-a-2D-NumPy-array`

[4] `https://stackoverflow.com/questions/3989016/how-to-find-all-positions-of-the-maximum-value-in-a-list`

[5] `https://stackoverflow.com/questions/15868512/list-to-array-conversion`

[6] `https://stats.stackexchange.com/questions/64676/statistical-meaning-of-pearsonr-output-in-python`

[7] `https://docs.scipy.org/doc/scipy-0.14.0/reference/generated/scipy.stats.pearsonr.html`

[8] `https://stackoverflow.com/questions/42579908/use-corr-to-get-the-correlation-between-two-columns`

[9] `https://stackabuse.com/k-nearest-neighbors-algorithm-in-python-and-scikit-learn/`

[10] `https://www.ritchieng.com/machine-learning-k-nearest-neighbors-knn/`

[11] `https://www.datascience.com/learn-data-science/fundamentals/introduction-to-correlation-python-data-science`

[12] `https://stackoverflow.com/questions/42579908/use-corr-to-get-the-correlation-between-two-columns`