Heriot Watt University

Data Mining and Machine Learning

Coursework 2

Bayesian Networks and Clustering

Mohit Vaishnav, *VIBOT*
Malav Bateriwala, *VIBOT*
Makenzy Abass Gumah

Supervisor
Dr Ekaterina Komendantskaya
Dr. Diana Bental

Submission Last Date 7^{th}, Nov.

Contents

List of Tables

List of Figures

1 Introduction

There are instances when a need appears to compute the probabilities of an uncertain cause given some observation. For an example, given the symptoms in any patient, disease is predicted by the doctor, similarly there can be numerous examples. With the increasing number of symptoms and disease this network will become more and more complex. In this kind of situations Bayesian approach is useful which deals with the uncertainty as well as complexity. This has to be implemented in the coursework.

This coursework is continuation of the last with the difference in using Bayes Network instead using K means algorithm. Bayes net is such a model which describes its states and the probabilities by which they are related. These probabilities are the result of occurrence of some states more often than the others present. In a Bayes net, links does not form cycles but may have loops which gives it an advantage to update quickly as there is no infinite loop because of cycles.

Bayesian Network provides the relationship between random variables and their conditional dependencies. Two of its parts are, *Conditional probability distribution* and *Directed Acyclic Graph (DAG)*. DAG represents an hierarchical structure so some of the node names common here are like parent, child, ancestor etc.

2 Data Conversion, Randomization and Reducing constraints

Following the same procedure as earlier, first the data is pre-processed to be used further. Python platform is used to implement this coursework using Pandas library which helps reading CSV format. The shape of the data set 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. Next part deals with the data randomization which is another essential step in any machine learning process. Later the data is split into two sets, training and validation in the ratio of 80:20.

Now many different combinations were tried to look for the analysis of the data set and performance. As the image is of size 48×48 total number of attribute becomes 2304 and number of samples are 35887 which leads to millions of attribute for the whole data. Hence to reduce the computational complexity and increasing speed, all the procedure is carried out over a limited number of attributes.

3 Implementation in Weka

The data is reduced to smaller data sets by using weka resample filter. This is to enable working with smaller dataset otherwise with a bigger datasets there is long hours of processing time. *Weka.filters.supervised.instance.Resample-B0.0-S1-Z10.0* is used for that purpose. In one of the learning in weka, changes are made to the number of seed and cross folds. For example, seed 1, 2, 5 and cross folds 2, 5, 10 are used for the resampling purpose to find out which one is suitable. It is found that using a smaller seed such as seed 2 and a small cross fold such as cross fold 5 algorithm is much faster to analyze.

Table. 1 indicates that when the same Weka filters is applied to the resample dataset emotion Disgust produces a higher classification as compared to the rest of the emotions. All these were run using cross validation with 10 folds.

Next, first 10 fields of each attribute emotions are selected for further analysis purpose. For this, Relief Attribute, Eval attribute and Ranker Evaluators are chosen be-

Classes	Angry	Disgust	Fear	Happy	Sad	Surprise	Neutral
Accuracy	61.6	98.49	66.75	58.99	62.82	66.50	58.91

Table 1: Precision value on resample dataset with 10 folds on Weka

cause it first calculates a feature score for each feature which can be applied to rank and select top scoring features. This feature is applied to determine weighs to guide downstream modeling. The later ranks attributes by their individually evaluating the attributes. This is done using full user training set attribute selection mode on weka. In Table. 2 different top fields can be seen. It also reflects a behaviour that for each kind of emotions, most of the fields appear near to each other.

Attributes	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10
Angry	2061	2109	2062	2013	2110	2112	2157	2060	2012	2208
Disgust	385	243	241	481	242	337	291	433	289	1923
Fear	2159	2017	480	2112	2111	479	2065	97	146	528
Happy	1697	1649	1698	1746	1650	1745	1648	1696	1744	1655
Sad	288	240	4	1	721	385	336	49	2	673
Surprise	1944	1943	1945	1947	1946	1850	1848	2038	2018	1846
Neutral	48	96	144	240	47	192	288	95	287	

Table 2: Pixel label for different emotion data set

4 Implementation in Python

In the first place, data is divided into train and test set using the Sklearn library function with the ratio as 80:20. After this Bayes algorithm is tried upon all the data set as a whole. As there are many variant of Naive Bayes algorithm, *GaussianNB* is chosen initially. Next the simulation is run over *BernoulliNB* with the input as whole data set again.

In the next variation, top fields from each of the emotion are found and taken into account for prediction value. Like in previous coursework, Pearson coefficient is used for finding out correlation which is imported from Scipy library. Over these attributes different classifiers are applied like *GaussianNB & RandomForest*

Now the data set is divided into a balanced ones where all the emotions are taken in equal proportion and then checked the performance accordingly.

Some of the other type of classifiers and methods tried are DecisionTreeClassifier, K Means and Keras Deep Learning model to find the best performance of this kind of data set.

5 Observation

With the use of GaussianNB variant of Naive Bayes algorithm, precision value obtained from confusion matrix is 27% which is very low from the last coursework where the average obtained was about 36%. ROC curve as shown in Fig. 1 illustrates the poor performance of this algorithm over the data set provided. In the Jupyternotebook provided along with this code, various other parameters are also evaluated like accuracy score (.26) and the area under the curve (.511).

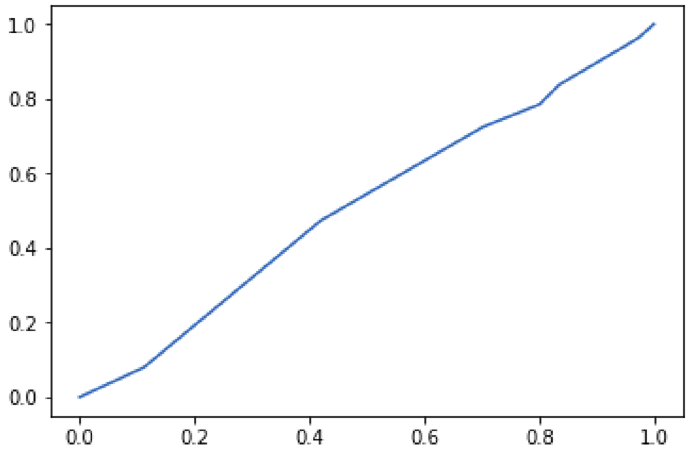

Figure 1: ROC curve for GaussianNB

In the next iteration with the *BernaulliNB* again the results were not very promising. Precision obtained is just 26% on an average and ROC as seen in Fig. 2.

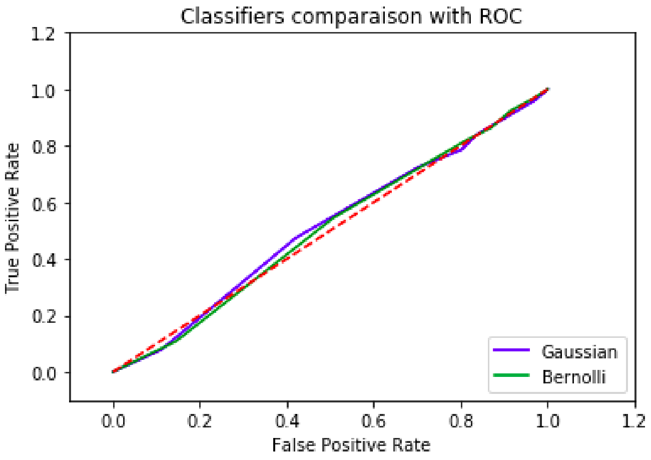

Figure 2: ROC curve for BernaulliNB

For the next analysis, different top level attributes are taken instead taking all 2304 attributes. Some of the combinations tried were top 2, 5, 10, 25 & 50 attributed obtained after calculating the Pearson coefficient. Over it classified applied is GaussianNB and the results are tabulated in Table. 3. Whereas when the same procedure is followed for a different type of classified named *RandomForest* better results are obtained.

Moving further, analysis was made with all the emotions having equal contribution in data set generated. For this purpose 500 emotions are selected from each of the seven classes and then a set is created of 3500 images. Now this data set is used for the analysis. Using the GaussianNB precision obtained is 22%.

Attributes	2	5	10	25	50
Precision (G)	0.25	.25	.26	.26	.28
Precision (RF)	0.32	.33	.32	.34	.36

Table 3: Precision value of GaussianNB (G) and RandonForest(RF) v/s number of attributes

Coming to the results of Random-Forest, it comes out to be better in all the cases in comparison to the normal NB methods. If the complete dataset is taken into account, accuracy of 38% is nearly obtained. As the best features are then considered for training the model, the output seems to increase when the best features are taken starting from 2 till 50 features increasing the accuracy from 33% to nearly 40%. But the thing to notice here is, when dataset of 500 images is taken, the accuracy comes to be around 24%. So here it can be said that, more data is better for this algorithm , and better if we just increase the best number of features and then train the model.

Next method tried is *DecisionTreeClassifier* which is also imported from Sklearn library which applies *T* functionality. Its a binary and multi class classification algorithm, which is used mostly for producing output using smaller subsets of each class and it predicts accordingly. This gave one of the best output of the whole results. Its performance is observed to be 32% in terms of precision when the complete dataset is being considered. When 50 top features are taken into account for training the model, the accuracy results drop down to 24%. When we use the data set of 500 images, it gave the results of 74% accuracy when it was predicted with the test set.

Now Unsupervised learning is implemented and observed. This is done for the K Means clustering method. When the clusters are defined to be 7, precision values is obtained to be 17%. If this model is run for the 3500 data set which is the balanced one then too there is slightly increase in the performance and it reaches to 19%. Another change made for this kind of algorithm is to remove the cluster while initializing and it worsens the performance to 12% precision value. This is because by default the number of clusters are taken to be 8 while this data set has 10 classes, so the result is expected.

Another unsupervised algorithm called Agglomerative clustering is also used for classification process. This links all the pixels of the same labels together and then tries to predict the output label according to the best match with its clusters. This clustering algorithm is mainly used when the dataset is very varid and also want to increase the data set with new input coming along the process to create a better model. By using this model 17% accuracy was achived.

At the end, comes the biggest and the best model which could be taken as the best predictor whose results could be bench marked with the above mentioned used algorithms. In this deep neural network, Keras is used with Tensorflow back end. 100 epochs were tried before deciding the final prediction values. More than 15 layers are included in this model including convolution layers, dropout layers etc. Total number of trainable parameters becomes 5,902,151. Learning rate is kept to be .01. The output of this model when run on separate training and testing data set is 64% which is by far the best achieved amongst all combination whereas if both of the training and testing are kept to be the same, it increases to 85%.

6 Conclusion

So finally we could say that again after making so many efforts on this data set, final output is far from the acceptable value of precision. Many different combinations were tried to make this model work better but all in vain. Using Bayes approach and implementing its different variants like Gaussian and Bernoulli performance achieved at max is 28% and 36% respectively which is not very great. One of the probable reason assumed last time was the balanced data set. So this time even this is taken care of by creating a new data set where all the emotions have the equal say. Over this kind of data set again different techniques were used which could be helpful in gaining insights but nothing significant was achieved. Finally unsupervised learning was implemented using K Means technique in Python which by default chose number of classes to be 8 which again led to bad accuracy. Hence for a change, deep learning model is used using Keras and tensorflow and it lead to a drastic gaining in terms of performance.

7 Research Question

Data set provided in this coursework has a very peculiar behaviour. Although many different techniques were tried but none seems to work fine. So there is a significant contribution that has to be made to prepare this data set. Upon observing many different images and their emotions, they do not represents the true behaviour of the emotions. For example, with the label as angry, the image is actually not angry, similarly this is true for other emotions too. So we need to have a properly labelled data set so that all features obtained are accurate leading to increase in the performance of any kind of model. Naive Bayes assumes that all the features in the data set are equally important and independent. Naives Bayes is believed to work more on word counting type of data where the probability map could easily be generated but here there are images that too not properly labelled. So theoretically performance has to decrease which was visible too. For understanding any kind of behaviour firstly there has to be proper input so that learning is true. So we could undoubtedly say that there is a strong demand of right data set as everything relies on this.

Research problem- Despite millions of image data available worldwide and many resourceful libraries, still we are lacking behind in prediction the emotions of any person. How can people determine user behaviours on social media using individual content? The main reason behind this research question is that there isn't a mechanism to determine the trustworthiness or behaviour of user content based on certain criteria. While doing the formulating the research question, several factors need to be known. Certain question become relevant to the question above and are the most relevant question with the problem stated above.

Answer to research question; In this of technological or disruptive age, social media networking sites produces voluminous big data people post of all sort of data such as text, video, audio etc. This has made social media to become one of the sources of news where people daily activities contributes to the massiveness of this big data. Knowing the factors that contribute to social media content and how it is analyzed is very important. Naïve Bayes or clustering algorithms can be used for analysis but not just this kind of algorithms. We have to move on higher order models and exploit the recently generated techniques for this purpose. To validate our point, here we tried implementing Deep learning model with ResNet Structure and the outcomes are incredible. There has been a magnificent gain in terms of performance. With the increasing processing capacity of normal cpu and gpu nothing seems impossible. Once the model understands the data properly, transfer learning can be further used to make full scale applications elsewhere. Other kind of alterations which can be made to process this fast and efficiently are using the techniques as mentioned above like finding the best attributes and training only those instead the whole set.

References

[1] https://www.bu.edu/sph/files/2014/05/bayesian-networks-final.pdf

[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://medium.com/deep-learning-turkey/deep-learning-lab-episode-3-fer2013-c38f2e052280