COMP5318 – Machine Learning and Data Mining

Assignment 1

"Classification with dimensionality reduction using Principal Component Analysis and Logistic Regression"

Mohammad Danial Azam(460241140)

Muhammad Arshad (460337384)

Introduction

The aim of this study is to design, implement and evaluate a suitable classification algorithm that can learn from the training dataset and be able to predict the labels for the test data with a high confidence. We have gone through the process of dimensionality reduction using PCA and then implemented multinomial logistic regression as our classifier.

The training data set contains of 20,104 samples with around 13,626 variables contained in each sample.

Methods

Exploratory Data Analysis:

One of the first steps we did in order to get a feel of the data was to explore the distribution of training labels attached to the training set. We found that almost all the labels were roughly equal in number, the exception being the 'Comics' label with a surprisingly low number of samples. Secondly on calculating the total count of non-zero values of each variable. We found two variables containing all zeros, which meant that they contributed no information towards the labels and were redundant.

Pre-processing methods: Since the size of the data set was considerably large, it was highly desirable to use effective dimensionality reduction technique. So decided to use the Principal Component analysis (PCA). Applying conventional PCA was not feasible with the computational power at hand. Our approach was to sample the dataset to a smaller set, then compute the PCA bases and use these bases to project the rest of the data in the smaller dimensional space. We used Incremental PCA function from the sklearn library to ensure that the process was more memory efficient. Since Incremental PCA applies the analysis on batches of data, we set

the batch size to be around 5,000 samples. Incremental PCA is convenient to use in our scenario since the memory available is too small to fit the entire data set.

After receiving the principal components we fitted and transformed our original data into a smaller matrix with the 200 most influential principal components covered.

Classifier:

One of the key benefits of using PCA is that it ensures that multicollinearity will not happen, which is one of the key assumptions of regression based models. Our classification algorithm is based on the logistic regression and is generalizing the logistic regression model for a multiclass problem. Our approach is to learn one logistic regression for each class. i.e. One vs all classes.

$$P(Y = 1|X) = \frac{e^{\beta_0+\beta_1 x1+\beta_2 x2+\cdots}}{1 + e^{\beta_0+\beta_1 x1+\beta_2 x2+\cdots}}$$

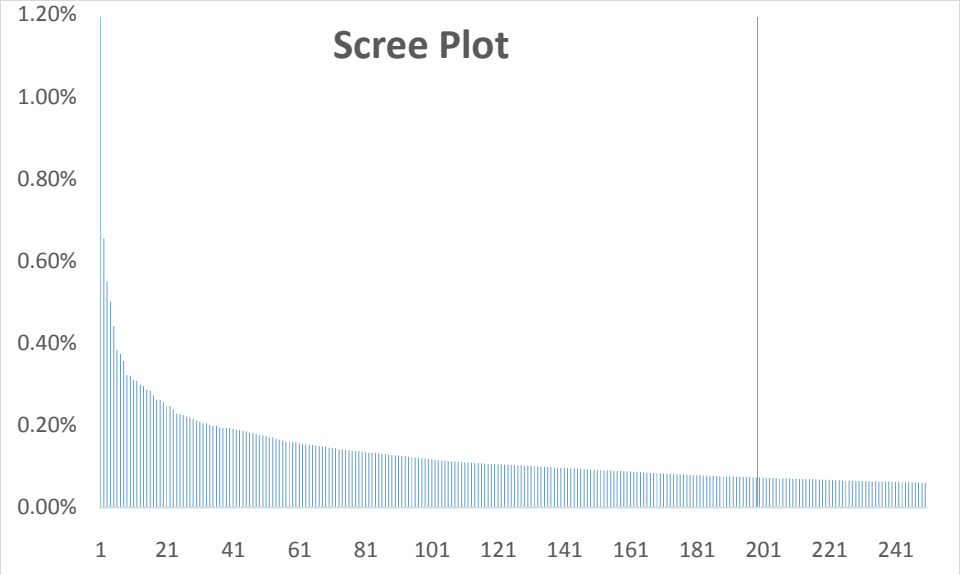
Logistic regression models parameters are optimised using the BFGS algorithm .i.e. β's are estimated using BFGS. To estimate the class of a new observation, we first fit the PCA to the new observation and then find the probabilities of new observation belonging to each class and assign the new record to class for which it has the highest probability. This algorithm is scalable and can handle any number of classes.

Experiments and Results

The Experiments and performance analysis were run on an HP Probook G2 core i7-5500u @2.5Ghz with 8GB RAM and windows 7 64bit OS.

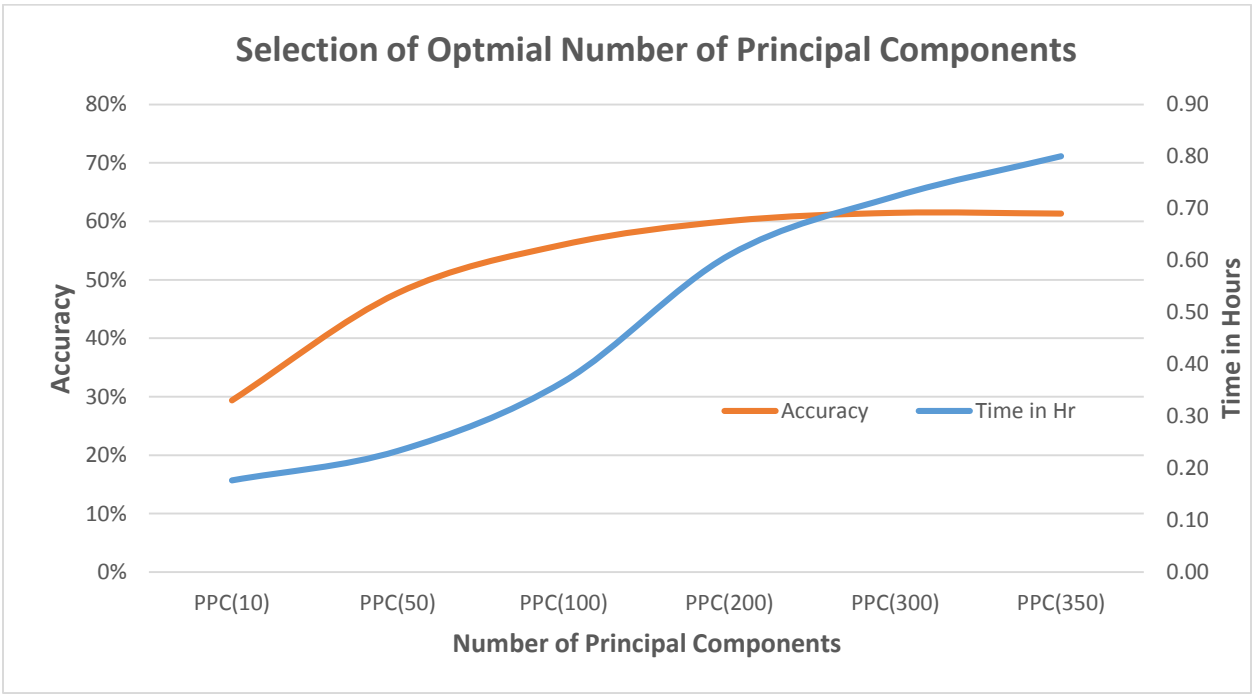
For testing and evaluating purposes we split the training data set randomly into two sets of 15,000 samples for training and 3000 for testing purposes. The run-time on these sets came out to be around 20 minutes for Principal Component analysis (PCA), while our Logistic Regression function can successfully predict labels for the test data in 7-8 minutes after receiving the input file produced by PCA.

We found that our accuracy fluctuated greatly with our choice of the number of components set for PCA. We started from a minimum of 10 principal components which gave us an accuracy of 29.877 %. Gradually increasing the number of principal components led to an increase in accuracy to up to 63.74% with 300 components selected. Though with 300 components our model was taking very long time to run and the rate of increase in model accuracy was very low, so we decided to use the first 200 principal components. Which effectively means that we reduced the data dimensionality by approximately 68 times.



Scree plots show that if we consider first 200 components out of 250, we will retain more than 90% of the variation in the dataset.

Also we tested our model with 500 principal components but the time to run the algorithm increase at much faster rate than as compared to accuracy. With 500 components our algorithm takes about 50-70 minutes. This time is inclusive of both PCA and multinomial regression and the final predictions on the test dataset.



The figure here illustrates our accuracy vs time and we chose 200 components to be the optimal state with the test taking 35 minutes to complete.

Model Results and Accuracy:

Table below shows the model accuracy measures for classifiers prediction function.

Labels	precision	recall	f1-score	support
Arcade and Action	0.61	0.63	0.62	104
Books and Reference	0.5	0.53	0.51	111
Brain and Puzzle	0.67	0.78	0.72	103
Business	0.49	0.36	0.41	97
Cards and Casino	0.83	0.89	0.86	106
Casual	0.45	0.51	0.48	86
Comics	0.78	0.5	0.61	42
Communication	0.55	0.58	0.57	113
Education	0.59	0.49	0.53	115
Entertainment	0.28	0.14	0.19	99
Finance	0.75	0.86	0.8	105
Health and Fitness	0.6	0.63	0.62	101
Libraries and Demo	0.74	0.57	0.64	68
Lifestyle	0.32	0.19	0.24	110
Media and Video	0.48	0.5	0.49	92
Medical	0.75	0.73	0.74	116
Music and Audio	0.65	0.79	0.71	106
News and Magazines	0.7	0.77	0.74	111
Personalization	0.53	0.79	0.64	94
Photography	0.61	0.79	0.69	107
Productivity	0.41	0.35	0.38	104
Racing	0.8	0.84	0.82	89
Shopping	0.66	0.71	0.68	125
Social	0.52	0.61	0.56	118
Sports	0.78	0.55	0.65	94
Sports Games	0.83	0.42	0.56	69
Tools	0.32	0.43	0.37	97
Transportation	0.59	0.66	0.62	116
Travel and Local	0.57	0.49	0.53	118
Weather	0.84	0.76	0.8	84

It is critical to highlight that the classifier for following classes performed very well

- Cards and Casino
- Finance
- Racing
- Weather

Whereas the classifiers for the following classes performed poorly

- Entertainment
- Lifestyle
- Tools
- Productivity

Upon close examination of the classifiers performing poorly we found that the classes Life Style, Entertainment and Media and Video were very hard to differentiate.

Our overall accuracy of predicting a correct class on a new dataset is 61.3%, i.e. on average this classifier will correctly predict the class with 61.3% accuracy.

Conclusion and Future Work

Considering 30 classes, the probability of correctly predicting a label for a given example is approximately 3.34%, and considering our algorithm accuracy of approximately 61%, that is 18 times better than a random guess so we can suggest that our algorithm is performing reasonably well, although there are areas of improvements for example our algorithm is susceptible to outliers. Future work should investigate the anomaly detection, the venue of reinforcement learning, deep learning and other supervised and unsupervised learning techniques.

Appendix

Instructions to run the code

First file PCA.py is for dimensionality reduction. It will create two files PCA_train_result.csv and PCA_test_result.csv. These two files are used by LogisticReg.py to run the classifier.

STEP 1: Run PCA.py

STEP 2: Run LogisticReg.py