IMAGE CLASSIFICATION PERFORMANCE COMPARISON OF NAÏVE BAYES AND LOGISTIC REGRESSION

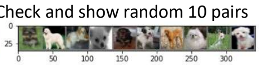
Student number: 200035575

Research Aim:

- This study starts from the **CIFAR-10 dataset[1]** which consists of 60000 32x32 colour images in 10 classes. To classify target in order to classes, two machine learning models have been used in a range of analysis.

Initial Investigation

- There are 50000 training images and 10000 test images. Check and show random 10 pairs or original images.



Baseline analysis

- For the Large dataset, an overview of runtime estimation(~12 sec) and classification model accuracy(~28%) has verified for NBC [Figure 1][Figure 2].
- To well-represent the classification task, the Sub-dataset training size(10000,3072) and test size(2000,3072) with target class 3(Cat) and 5(Dog) has considered for the combination of 2 classes would return the average result [Figure 3].

Naive Bayes Classifiers (NBC)

- Simplified assumption: variables are independent conditioned on the class label.
- For categorical variables, we can estimate their probability by counting its occurrence divided by the total number of related samples.
- Pros: Greatly reduce the parameters & computational cost[2].
- Cons: Naïve assumption can hardly happen in the real world, which details vulnerable to probability estimation without prior[3].

Logistic Regression Classifiers (LGC)

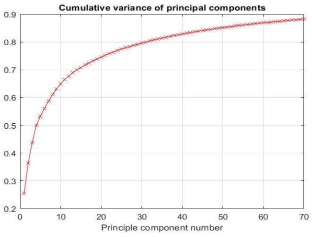
- As a discriminative classifier, it directly model the likelihood P(Y|X) or f: X→Y in ascending order.
- Parameters retains a probabilistic semantics.
- Pros: A simple linear classifier with a few parameter learned by iterative. optimization, and suppose less assumption
- Cons: Slow speed at the beginning[4].

Hypothesis Statement

- With linearly separable two classes, both Naïve Bayes and Logistic Regression classify well[4].
- For the PCA model, the hyper-parameter k is possibly determined by different k values be trained and compared[5]. This study expects the best k would be the dependent value for the max variances.
- NBC would be faster to converge but LGC would eventually catch up and overtake the speed performance[4].
- Accuracy would be saturated around 50~60% since the sample subset has chosen by the average range of classification.
- The performance would be able to do cross-verification using a different programming (MATLAB and Python language for this study).

Common Hyper-parameter k for the lower dimension

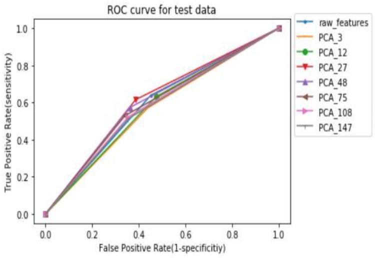
[Figure 6]



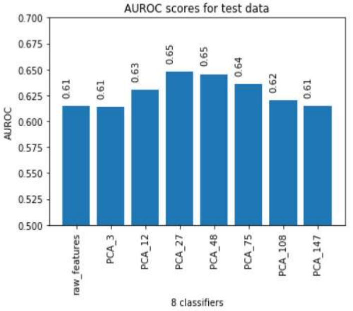
- Feature selection has been explained to pick random values[6] because principal components are for directions not any other measurements.
- However, the component whose value is bigger than 100 doesn't significantly increase the cumulative variance.
- Hence, 4 values **k1=30, k2=50, k3=80, k4=100** under 100 have been taken by convenience.

Performance Evaluation of eight different NBC with PCA

[Figure 7]

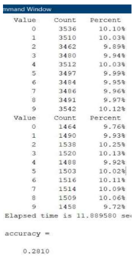


[Figure 8]



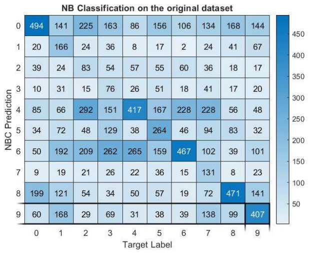
- Figure7 and Figure8 are performed in Python Programming.
- A Receiver Operator Characteristic(ROC) curve is a graphical plot used to show the diagnostic ability of binary classifiers. ROC curve shows the trade-off relationship between sensitivity(tpr) and specificity(1-fpr). When accuracy is not much high, for example PCA_3, the curve is closer to the 45 degree diagonal of the ROC space. In the opposite, the highest accuracy on test dataset which is on PCA_27 performs more closer curve to the top-left corner.
- With Area Under the Curve(AUROC) scores, we can see better accuracy do better discriminate among the classifiers. It results that good PCA components with Naive Classifiers return good output. Therefore, we can notice the two middle range of PCA components are doing well compared to others even better than raw features again.

[Figure 1]



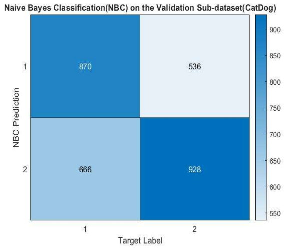
```
mmand Window
    Value    Count   Percent
        0     3536    10.10%
        1     3510    10.03%
        2     3462     9.89%
        3     3480     9.94%
        4     3512    10.03%
        5     3497     9.99%
        6     3484     9.95%
        7     3486     9.96%
        8     3491     9.97%
        9     3542    10.12%
    Value    Count   Percent
        0     1464     9.76%
        1     1490     9.93%
        2     1538    10.25%
        3     1520    10.13%
        4     1488     9.92%
        5     1503    10.02%
        6     1516    10.11%
        7     1514    10.09%
        8     1509    10.06%
        9     1458     9.72%
Elapsed time is 11.889580 se

accuracy =

    0.2810
```

[Figure 2]



	0	1	2	3	4	5	6	7	8	9
0	494	141	225	163	86	156	106	134	168	144
1	20	166	24	36	8	17	2	24	41	67
2	39	24	83	54	57	55	60	36	18	17
3	10	31	15	76	26	51	18	41	17	20
4	85	66	292	151	417	167	228	228	56	48
5	34	72	48	129	38	264	46	94	83	32
6	50	192	209	262	265	159	467	102	39	101
7	9	19	21	26	22	36	15	131	8	23
8	199	121	54	34	50	57	19	72	471	141
9	60	168	29	69	31	38	39	138	99	407

[Figure 3]



	1	2
1	870	536
2	666	928

Exceptional Hyper-parameters for NBC vs LGC

[Naïve Bayes Option]

- Normal Distribution / Kernel: Gaussian
- Prior probabilities: Empirical

[Logistic Regression Option]

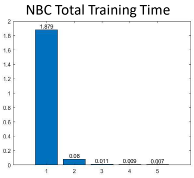
- Type of Model to Fit: Nominal
- Confidence Level: 0.95

Methodology:

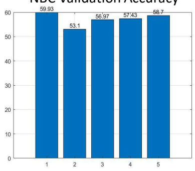
1. **Pre-processing in Python:** Create CatDog Sub-Dataset
2. **Data loading and Manipulation:**
   i. 70% Training data (7000, 3072) and target (7000, 1)
   ii. 30% Validation data (3000, 3072) and target (3000, 1)
   iii. Test data (2000, 3072) and target (2000, 1)
3. **Dimensionality Reduction:** applied a PCA classifier for two ML models. Every classifier is having different value k for the reduced dimensionality.
4. **Train and Validate** CatDog Sub-Dataset with NBC and LGC model.

CatDog Dataset	NBC					LGC				
	Base	k1	k2	k3	k4	Base	k1	k2	k3	K4
Time (sec)	1.879	0.08	0.011	0.009	0.007	N/A	0.193	0.19	0.198	0.279
Accuracy (%)	60	53	57	57	59	N/A	51	56	57	57
Best Model	-				✓	-				✓

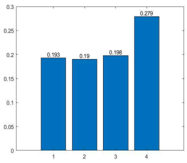
[Figure 4]
NBC Total Training Time



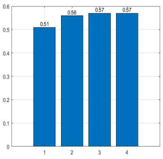
[Figure 5]
NBC Validation Accuracy



[Figure 9]
LRC Total Training Time



[Figure 10]
LRC Validation Accuracy



Analysis and Evaluation of Results

[NBC Baseline]

- Train dataset Accuracy: 60.1%

[NBC Best Model]

- Training Time: 0.0566 sec
- Test dataset Accuracy: 57.9%

[LGC Baseline]

- Train dataset Accuracy: N/A as intractable and optimization yields complex coefficients

[LGC Best Model]

- Training Time: 1.5779 sec
- Test dataset Accuracy: 62.1%

Total Training Time: NBC < LGC. But this differences are not significantly big.

Classification Accuracy: NBC > LGC

- Gaussian Naive Bayes is better for this CIFAR-10 dataset classification because these color images are consisted of continuous components which are pixel values.
- Additionally, it is also good to use Gaussian Naive Bayes Classifier because each independent conditioned variables are following Gaussian distribution(i.e. P(x_1|C_i),P(x_2|C_i),…,P(x_p|C_i) when k=1 to p are following Gaussian distribution in this case).
- If it was Multinomial, it will take every 0-255 pixels for each feature so it most probably takes much longer time and more memories to apply classifier on PCA features.
- No matter the size of PCA components, the accuracy rate does not increase in proportion to its size.

Lessons learned and Future work

- For later work, it would be worth to split training dataset into selections of validation target labels.
- Training the logistic regression model on the full dataset was time intensive and yielded complex coefficients. Either implementing a bespoke algorithm or using different software in the future for implementing the logistic model would be best.
- Learn how to image process in MATLAB and CNN machine learning algorithm to compare performances.

References:

1) Collected by Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton from the University of Toronto. With 6000 images per class, each image is a 3-channel colour images of 32x32 pixels in size.
2) VanderPlas J., 2018. 05.05-Naive-Bayes. https://github.com/jakevdp/PythonDataScienceHandbook/blob/master/notebooks/05.05-Naive-Bayes.ipynb
3) Moon, I., 2010. Week3 Naïve Bayes Classifier, p.18. https://github.com/aailabkaist/Introduction-to-Artificial-Intelligence-Machine-Learning/blob/master/Week03/IE661-Edu3.0-Week%203-icmoon-ver-2.pdf
4) Ng, A. and Jordan, M., 2001. On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. *Advances in neural information processing systems*, 14, pp.841-848.
5) Christopher M. Bishop. 2006. Pattern Recognition and Machine Learning (Information Science and Statistics). Springer-Verlag, Berlin, Heidelberg. p.484
6) Xing, E.P., Jordan, M.I. and Karp, R.M., Feature Selection for High-Dimensional Genomic Microarray Data.