The analysis of SVM performance in comparison to LR upon the CIFAR-10 dataset, also considering the use of Dimensionality Reduction Techniques such as LDA & PCA

Bahawal Waraich

880114

Content

1 Introduction

As humans, we rely heavily on object recognition, although we may not notice this to be the case as it is a subconscious mechanism. Our brains are constantly categorising objects that we see so that in the future they can be easily and readily recalled whilst being correctly associated with a corresponding entity. As technology has evolved image data storage has increased exponentially, thus highlighting the need to transfer the notion of object recognition to machines as it can be utilised to offer various functionalities such as the example of using facial recognition for authorisation. To further explore the field of object recognition, an experiment was conducted upon the CIFAR-10 dataset. The dataset contains 10000 training and 1000 testing images that belong to 10 different categories, in which the supervised learning algorithms SVM (Support Vector Machines) and LR (Logistic Regression) would be applied to assess their performance. Pre-processing techniques such as PCA (Principal Component Analysis) and LDA (Linear Discriminant Analysis) were also applied to determine their effects when combined in use with SVM. An overview of the results can be seen in Figure 1.

2 Method

2.1 Preparation

Compression was carried out on the data, which entailed the extracting of features that could be used to differentiate between images so that less information would be required(1). This was done, to increase the performance of the learning algorithms, which would, in turn, reduce computational costs. A 'for loop' was implemented for the training data and testing data respectively. Within the 'for loop' itself, the 'computeFeatures' function was applied iteratively to each of the samples within the given datasets. The aforementioned function computes the histograms of gradients for an image, producing a vector containing the output. The extracted features are then saved into a two-dimensional numpy array, which comprised of samples and subsequent features. The array was set to [10000, 324] for training data and [1000, 324] for testing data, indicating the size of samples and features.

2.2 Pre-Processing

Once the features were extracted, they were pre-processed with the use of dimensionality reduction in the form of LDA and PCA. Further improving the performance of the algorithms and therefore reducing the computational costs, further to that, dimensionality reduction lowers the likelihood of overfitting; training an algorithm too well causing it to recognise an image to such a degree that it interferes with the performance of algorithms(2). Pre-processing data using this approach can be beneficial as a redundant feature can be removed with minimal effect on the accuracy of the algorithms. LDA computes a vector which best discriminates between classes, on the other hand, PCA computes vectors that have the greatest variance associated with them(3). both algorithms were implemented individually in coordination with supervised learning algorithms to investigate how each algorithm performs on the given data set.

2.3 Classifiers

SVM and LR were the classifiers that were used. Both algorithms sort images into their specific classes by using hyperplanes to distinguish between images. Depending on which side of the hyperplane the image is assigned to, it will be registered as belonging to a corresponding class. Multiple hyperplanes are used to extend the procedure to apply to higher dimensionality data(4).

2.3.1 SVM

SVM was chosen as it not sensitive to outliers, and in our image data, there may be many outliers when considering the variation that could be present between one airplane and another (e.g. shape, colour, size etc.), which could in turn skew classification(4). When implementing the SVM, an instance of the model was first created and then fitted to the extracted feature training sets and labels. The model would then be used to predict the class of the test images. An accuracy percentage would be calculated regarding the successful prediction of test images. A confusion matrix would be created and printed as well as a classification report to gain further insight into the performance of the model. The model would be executed individually with, raw feature extracted data, data that is pre-processed with PCA and data pre-processed with LDA. This was done to observe how PCA and LDA may individually impact the performance of the SVM. In both cases, the dimensionality of the feature data would be reduced to 2 dimensions from 324. This, in theory, should greatly reduce computational cost as the total number of calculations is greatly diminished.

2.3.2 LR

LR is considered a traditional method of machine learning, thus was chosen as a baseline for comparison to further assess the effectiveness of SVM. The implementation of LR was similar to that of SVM; an instance of the model was created, it was fitted to extracted feature data and a prediction was made regarding image classification. Pre-processing in the form of PCA and LDA was also applied so that it could be compared against the performance of SVM when utilising dimensionality reduction, also allowing for general conclusions to be formed regarding effects of PCA and LDA. Confusion matrices and classification reports were also created.

3 Results

SVM used alone had the highest precision percentage (51.4%), however, was closely followed by LR (49.9%). LDA and PCA reduced the precision of both SVM and LR with PCA causing the more drastic reduction with SVM+PCA (66.6% less accurate) and LR+PCA (61.1% less accurate), as a comparison to SVM+LDA (21.2% less accurate) and LR+LDA (30.1% less accurate). LDA and PCA having a similar effect on both the Recall and F-Beta score percentages. On the other hand, LDA and PCA significantly decreased runtime on both classifier algorithms; SVM alone took 24.7s while with LDA it took 2.3s and 2.8s with PCA; LR alone taking 1.29s whilst with LDA and PCA reducing the time to under a second. Comparing SVM and LR directly, we see that in all cases LR is drastically faster than SVM with LR alone taking only 30% of the total runtime for LR. We see that SVM alone performed better amongst all classes when compared to the additional use of LDA and PCA. PCA and LDA would inhibit the performance amongst different classes. For example, PCA would lower correctly identified class 0 images to 2 (from 72 with SVM alone) whereas LDA would only lower correctly identified class 0 images to 61 (from 72). On the contrary, SVM+ LDA was more likely to wrongly classify images in some cases. For example, we see that the use of LDA identifies 17 images that belong to class 8 as incorrectly belonging to class 0 whilst PCA identifies only 5 images in the same nature. This is partially due to PCA having an overall lower precision percentage however, in terms of ratio, PCA+SVM is roughly half as accurate as LDA+SVM. Despite this, PCA+SVM only wrongly classifies roughly a third of the images LDA+SVM does. The classification reports clearly show the precision percentages of individual classes, we see that the classification reports further justify the results of the confusion matrix. For example, regarding the first example addressing the first row and the first column of matrices, we see that the results of the classification report concur; class 0 was initially completed to 59% precision, PCA+SVM was lowered to 8% whereas LDA+SVM was increased to 63% due to fewer images being wrongly categorised into class 0. We see that although LDA+SVM may wrongly classify images in some cases, although in general, it will have a higher precision across classes on average, with fewer cases of incorrect image classification.

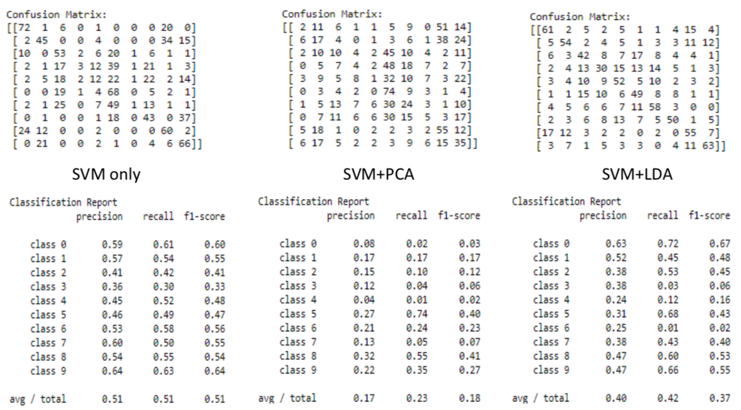
4 Conclusion

Various observation can be deduced from the conducted experiment involving the application of SVM and LR with and without the use of PCA and LDA for dimensionality reduction. We see that SVM is more accurate LR when performing image classification with no pre-processing, however, it is significantly slower thus making it difficult to assume which would be the better choice of an algorithm in this situation. With PCA and LDA pre-processing, the trend continues with SVM once again having greater precision at the cost of longer runtime on average. An interesting case to highlight is the comparison of LDA+SVM and LR+LDA. LDA+SVM could arguably be considered the best-case combination of algorithms in terms of satisfying high accuracy whilst maintaining low runtime; it offered a high accuracy (40.2%) whilst only having 2.3 seconds run time as opposed to LDA+LR which had a slightly lower accuracy (35.7%) with an extremely low runtime of less than a second. We see that LDA, in general, offered the best compromise in terms of high accuracy and low runtime when compared to, with or without PCA, for both SVM and LR. We learn that one has many different options when performing supervised machine learning, algorithms should be picked dependant on what the situation in question prioritises; speed or accuracy. If a general compromise of both factors was to be desired then, judging by the results of this experiment, one should consider the combination of LDA and SVM as this performed the best in our experiment. Although, this could be due to variables such as the specific dataset in question. To further the findings of this experiment we could have applied additional algorithms such as neural networks so that further comparisons could be made against SVM, we could have also used different datasets so that findings could be validated along with finding the optimal number of compnenets for both PCA and LDA.

Figure 1: Overview of Results

	SVM only	SVM + PCA	SVM + LDA	LR only	LR + PCA	LR + LDA
Precision (%)	51.4	17.2	40.2	49.9	20.2	35.7
Speed (seconds)	24.7	2.8	2.3	7.4	<1	<1
Recall (%)	51	23	42	51	23	41
F-Beta Score (%)	51	18	37	50	18	35

Precision: Of the images that are placed within a class, how many are correctly assigned.
Runtime: How quickly an algorithm carried out image classification.
Recall: How many images are identified as pertaining to the corresponding class.
F-Beta Score: An average of precision and recall.
Confusion Matrix Interpretation: Taking the first row and column as an example; the first row indicates class 0, looking at the first column we see 72 images that belong to class 0 were correctly classified whilst looking at the next row, in the same column, 2 images belonging to class 1 were classified as class 0.
Classification Report Interpretation: Each individual class is assigned a precision, recall and f-beta score.



```
Confusion Matrix:
[[72  1  6  0  1  0  0  0 20  0]
 [ 2 45  0  0  4  0  0  0 34 15]
 [10  0 53  2  6 20  1  6  1  1]
 [ 2  1 17  3 12 39  1 21  1  3]
 [ 2  5 18  2 12 22  1 22  2 14]
 [ 0  0 19  1  4 68  0  5  2  1]
 [ 2  1 25  0  7 49  1 13  1  1]
 [ 0  1  0  0  1 18  0 43  0 37]
 [24 12  0  0  2  0  0  0 60  2]
 [ 0 21  0  0  2  1  0  4  6 66]]
```

SVM only

```
Classification Report
             precision    recall  f1-score

    class 0       0.59      0.61      0.60
    class 1       0.57      0.54      0.55
    class 2       0.41      0.42      0.41
    class 3       0.36      0.30      0.33
    class 4       0.45      0.52      0.48
    class 5       0.46      0.49      0.47
    class 6       0.53      0.58      0.56
    class 7       0.60      0.50      0.55
    class 8       0.54      0.55      0.54
    class 9       0.64      0.63      0.64

avg / total       0.51      0.51      0.51
```

```
Confusion Matrix:
[[ 2 11  6  1  1  5  9  0 51 14]
 [ 6 17  4  0  1  3  6  1 38 24]
 [ 2 10 10  4  2 45 10  4  2 11]
 [ 0  5  7  4  2 48 18  7  2  7]
 [ 3  9  5  8  1 32 10  7  3 22]
 [ 0  3  4  2  0 74  9  3  1  4]
 [ 1  5 13  7  6 30 24  3  1 10]
 [ 0  7 11  6  6 30 15  5  3 17]
 [ 5 18  1  0  2  2  3  2 55 12]
 [ 6 17  5  2  2  3  9  6 15 35]]
```

SVM+PCA

```
Classification Report
             precision    recall  f1-score

    class 0       0.08      0.02      0.03
    class 1       0.17      0.17      0.17
    class 2       0.15      0.10      0.12
    class 3       0.12      0.04      0.06
    class 4       0.04      0.01      0.02
    class 5       0.27      0.74      0.40
    class 6       0.21      0.24      0.23
    class 7       0.13      0.05      0.07
    class 8       0.32      0.55      0.41
    class 9       0.22      0.35      0.27

avg / total       0.17      0.23      0.18
```

```
Confusion Matrix:
[[61  2  5  2  5  1  1  4 15  4]
 [ 5 54  2  4  5  1  3  3 11 12]
 [ 6  3 42  8  7 17  8  4  4  1]
 [ 2  4 13 30 15 13 14  5  1  3]
 [ 3  4 10  9 52  5 10  2  3  2]
 [ 1  1 15 10  6 49  8  8  1  1]
 [ 4  5  6  6  7 11 58  3  0  0]
 [ 2  3  6  8 13  7  5 50  1  5]
 [17 12  3  2  2  0  2  0 55  7]
 [ 3  7  1  5  3  3  0  4 11 63]]
```

SVM+LDA

```
Classification Report
             precision    recall  f1-score

    class 0       0.63      0.72      0.67
    class 1       0.52      0.45      0.48
    class 2       0.38      0.53      0.45
    class 3       0.38      0.03      0.06
    class 4       0.24      0.12      0.16
    class 5       0.31      0.68      0.43
    class 6       0.25      0.01      0.02
    class 7       0.38      0.43      0.40
    class 8       0.47      0.60      0.53
    class 9       0.47      0.66      0.55

avg / total       0.40      0.42      0.37
```

Figure 2: SVM Matrices and Classification Report

5 References

(1) - K. L. Lee and M. M. Mokji, "Automatic target detection in GPR images using Histogram of Oriented Gradients (HOG)," 2014 2nd International Conference on Electronic Design (ICED), Penang, 2014, pp. 181-186.

(2) - Nasser M. Nasrabadi, Nasser M. Nasrabadi, "Pattern Recognition and Machine Learning," Journal of Electronic Imaging 16(4), 049901 (1 October 2007).

(3) - Martinez, A. M., & Kak, A. C. (2001). PCA versus LDA. IEEE Transactions on Pattern Analysis and Machine Intelligence, 23(2), 228–233. doi:10.1109/34.908974

(4) - Noble, W. S. (2006). What is a support vector machine? Nature Biotechnology, 24(12), 1565–1567. doi:10.1038/nbt1206-1565