Analysis about robustness of two Non-negative Matrix Factorization algorithms

Yang Ge (450005028), Haoyue Li (470136555), Suwan Zhu (480090900)

Abstract

With the rapid development of calculating power, storage space and network, the amount of data accumulated by human beings is growing quickly. In this circumstance, machine learning is particularly necessary. Solving classification problems is one of the important tasks in machine learning and data mining. This paper will focus on the application of logistic regression algorithm, one of the fields of machine learning, which solve the problem of image multi-classification and expounding its principle and algorithm process. We use a logistic regression classification algorithm to build a classifier that trains 30,000 data sets. The test model was used to predict 2000 test data sets and divided into 10 categories. The performance of the classification algorithm is evaluated and its accuracy is calculated.

1.Introduction

1.1 The aim of the study. Classification is an important task in data mining and machine learning. The purpose of classification is to learn a classification function or classification model (also often referred to as a classifier) that maps data items in a database to a particular category in given categories. Classification is a kind of guided learning, and the learning of the model is carried out under the guidance of which class each training sample belongs to. And randomly selected from the sample group. Each training sample has a corresponding specific class label, which is not used for unsupervised learning (clustering).

The goal of this study is to develop an automated image classifier. The classifier predicts a given picture and determines which category the picture belongs to. The original data set is trained using a logistic regression algorithm to build a training model. The unknown image data is input into the training model for predicting, and then the category of the unknown image is analyzed. By learning classification algorithms and developing automated categorizers, the process of processing big data becomes more efficient, saving time and improving accuracy.

1.2 The importance of the study. In this study, it is mainly to solve the multi-classification problem of pictures. The logistic regression algorithm is a powerful statistical method that can be used to model data containing one or more variables as a binomial type model, estimating the probability by using a logical function of cumulative logical distribution. The advantage of this algorithm is that the output value naturally falls between (0, 1) and is more probabilistic. Therefore, it applies to this study, which will be discussed in following section. Such algorithms are also often used in real life, such as credit evaluation, measuring the success of marketing and sales, predicting the profitability of a product, and predicting whether an earthquake will occur on a particular day.

By collecting a large amount of information data, the amount of data of training samples is continuously improved. Analyzing data and building the machine models of massive data. Use models to predict unknown data and classify them. This approach can process large amounts of application data in a reasonable amount of time. Currently, this multi-class classification task is usually done manually, and humans cannot avoid errors when dealing with big data. Therefore, developing automated classifier will save significant time and improve accuracy. This is also the significance of this study.

2. Methods

2.1 Pre-processing.

Min-Max normalization. In the field of machine learning, different evaluation indicators often have different dimensions and dimension units. Such situations will affect the results of data analysis. In order to eliminate the dimensional influence between indicators, data standardization is needed. This will solve the comparable problem between data indicators. After the original data is processed by data standardization, each indicator is in the same order of magnitude, which is suitable for comprehensive comparative evaluation.

The original data set is a pixel feature value after the RGB image is converted into a grayscale image, and its range is limited to (0, 255). Since this study is processing the image, the values of the entire data set are concentrated, so Min-Max normalization is used in the preprocessing of the data compatibly. This method can be better applied to quantify the pixel intensity. Min-Max normalization scales the eigenvalues within a specific interval of (0,1), making it easier to perform subsequent calculations and eliminate the adverse effects of singular sample data. After the data is normalized, the optimization process of the optimal solution will obviously become smoother, and it is easier to correctly converge to the optimal solution, which ultimately improves the accuracy of the classifier. The function of Min-Max normalization is:

$$x' = \frac{x - min(x)}{max(x) - min(x)} \tag{1}$$

PCA (Principal Component Analysis). In order to reduce the cost of data training, a PCA algorithm is applied to reduce the dimension of the original high-dimensional data, which consists of some variables correlated with each other, either heavily or lightly. Through PCA, the original data is transformed into a set of linearly independent representations. And the representation of original data will be in a pretty lower dimension which will cost less when do the data training.And when reducing dimension, the useless features of the original data will be deleted.This will deduce the impact of noise features on prediction. However, there is also a risk, the features of original data maybe loss with the dimension reduced. Blow are the main steps of PCA.

1. Transform the original data into matrix X with n rows and m columns;
2. Zero-averaging each row of X (each element in this row subtracts the mean of this row);
3. Calculate the covariance matrix C;

$$C = \frac{XX.T}{m} \tag{2}$$

4. Calculate the eigenvalues of the covariance matrix and the corresponding eigenvectors;
5. Arrange the feature vectors into a matrix according to the corresponding feature value from top to bottom, taking the first k(new dimension) rows to form a new matrix P;
6. Get the new Matrix Y with k dimension;

$$Y = PX \tag{3}$$

2.2 Logistic Regression. In this experiment, we use Logistic Regression as our classifier. Logistic Regression is a discriminative model for binary classification.

Considering weights for a Linear Regression,

$$W = \begin{pmatrix} w_0 \\ w_1 \\ ... \\ w_n \end{pmatrix} \tag{4}$$

Then the Linear Regression could be represented as

$$z_i = w_0 + w_1x_1 + w_2x_2 + ... + w_nx_n = X_iW \quad (5)$$

Logistic Regression use Linear Regression with logistic transformation applied, which is sigmoid function, to calculate a probability $p(x)$.
Sigmoid Function:

$$f(x) = \frac{1}{1+e^{-x}} \quad (6)$$

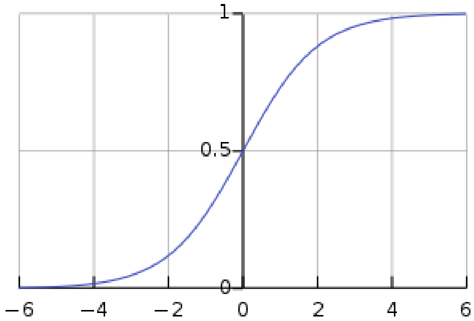


Figure 1. graph for sigmoid function

Sigmoid function maps any real value into value between 0 and 1. In Logistic Regression, we use it to find probabilities.
For $Y\epsilon\{1,\ 0\}$,

$$P(Y_i = 1|X_i, W) = \frac{e^{z_i}}{1+e^{z_i}} = \frac{1}{1+e^{-z_i}} \quad (7)$$

$$P(Y_i = 0|X_i, W) = 1 - P(Y_i = 1|X_i, W) \quad (8)$$

Where $P(Y_i = 1|X_i, W)$ is the probability that data belong to label $Y_i = 1$, $P(Y_i = 0|X_i, W)$ is the probability that data belong to label $Y_i = 0$. If $P(Y_i = 1|X_i, W) > P(Y_i = 0|X_i, W)$, the data would be assign label $Y_i = 1$.
Then our objective is to find appropriate weights W for prediction.
Let $p_i = P(Y_i = 1|X_i, W)$, considering Bernoulli distribution, the formula for probability could be write as

$$P = \begin{cases} p_i,\ y_i = 1 \\ 1 - p_i,\ y_i = 0 \end{cases} = p_i^{y_i}(1-p_i)^{1-y_i} \quad (9)$$

The likelihood can be written as:

$$L(W) = \prod_{i=1}^{N} p_i^{y_i}(1-p_i)^{1-y_i} \quad (10)$$

Our objective become to find a W that maximize the likelihood. Taking log, the log-likelihood can be written as

$$l(W) = \sum_{i=1}^{N}(y_ilog(p_i) + (1-y_i)log(1-p_i)) \quad (11)$$

then taking minus, simplify the function, our loss function become to

$$loss(W) = -l(W) = \sum_{i=1}^{N}(log(1+e^{z}) - y^iz^i) \quad (12)$$

Our objective is to find W that minimize this loss function

$$W = \underset{W}{argmin}\ loss(W) \quad (13)$$

One approach is using Gradient Descent Method.

2.3 Gradient Descent. Gradient descent is a algorithm that used to find the value that minimize a function. For a convex function, Gradient Descent could find global minimum, otherwise, it may find a local minimum.

Lemma 1 *Taylor's Theorem: Let $k \geq 1$ be an integer and let the function $f : R \rightarrow R$ be k times differentiate at the point $a\epsilon R$. Then there exists a function $W_k : R \rightarrow R$ such that*

$$f(x) = f(a)+f'(a)(x-a)+...+\frac{f^k(a)}{k!}(x-a)^k+W_k(x)(x-a)^k \quad (14)$$

and $\lim\limits_{x\rightarrow a} W_k(x) = 0.$

Let $W_{k+1} = W_k + \eta d_k$, by Taylor's theorem, we have

$$loss(W_{k+1}) = loss(W_k) + \eta\bigtriangledown loss(W_k)^Td_k + o(\eta) \quad (15)$$

Then if we choose a small η, $loss(W_{k+1})$ is smaller than $loss(W_k)$, if the direction d_k is chosen so that

$\nabla loss(W_k)^T d_k < 0$ when $\nabla loss(W_k) \neq 0$.
Gradient descent method set $d_k = -\nabla loss(W_k)$, so that $loss(W_{k+1})$ is smaller than $loss(W_k)$. Therefore, the gradient descent method could be represented as

Algorithm 1 Gradient Descent

```
initialize $\eta$ and $W$  while $W$ does not converge do
    $W_{d+1} \leftarrow \eta \frac{\partial}{\partial W_d} loss(W)$
end
```

In this experiment, the derivative for weight is

$$\frac{\partial}{\partial w_d} loss(W) = \sum_{i=1}^{N} (\frac{e^z}{1+e^z} - y_i) x_{id} \tag{16}$$

2.4 multiple class classification.

One Vs All (OVA). In general, most of the classification problems in machine learning solve the two-class problem, that is, there are only two outputs, 0 or 1. In reality, multi-classification learning tasks are often encountered. Some binary learning methods can be directly extended to multiple classifications, but in more cases, it is necessary to use a two-class learning device to solve the multi-classification problem based on some basic strategies. In this study, 10 different categories of images were processed, which is a multi-classification problem. Instead of y=(0,1), this study expand the definition so that y=(0,1,...,9).

Generally, considering N categories $C1, C2, ..., C_N$, the basic idea of multi-classification is "disassembly method". It is to split the multi-category task into multiple two-category tasks for solving. Specifically, the problem is split first, and then a classifier is trained for each of the split two tasks. At the time of testing, the predictions of these classifiers are integrated to obtain the final multi-category results. The method used in this study is One Vs All or One Vs Rest.

The so-called one-vs-all method is to apply the two-category method to multi-class classification. Each time a sample of a class is used as a positive class, all other classes of samples are used as negative classes to train N classifiers. If only one classifier produces a positive class during the test, the final result is the classifier. If multiple positive cases are generated, the confidence of the classifier is judged, and the sub-category mark with high confidence is selected as the final classification result. In this study, one category is treated as 1 and the remaining 9 categories are considered as 0. This is back to the binary classification problem, and it is easier to solve the problem.
In this experiment, we train 10 models as there are 10 different labels. Given a single test data, we can get 10 probabilities, each of them represent the probability that the data belong to a particular class. We assign the label with the highest probability to the data.

3. Experiments and Discussions

In this section, a compared experiment is implemented to explore the impact of data dimension on the prediction. And two aspects are measured, which are accuracy of prediction model and cost(time) on training data. In reality, a different k dimension will be chosen for different purpose so as the experiment.

3.1. Datasets

Original Data. The original dataset consists of 30,000 clothes images belonging to 10 different category and a test set of 2,000 examples. All images are store in hdf5 format with the size of 28x28.

3.2 Measurement methods

For evaluation of performance of the prediction model, the Average Accuracy (ACC) method is implemented also with P(precision), R(recall) and F-score. They are usually used to judge performance of machine learning algorithms and prediction accuracy of different category of data.

P(precision):the rate of diagonal of the matrix over sum of total numbers for predicted label. R(recall): the rate of diagonal of the matrix over

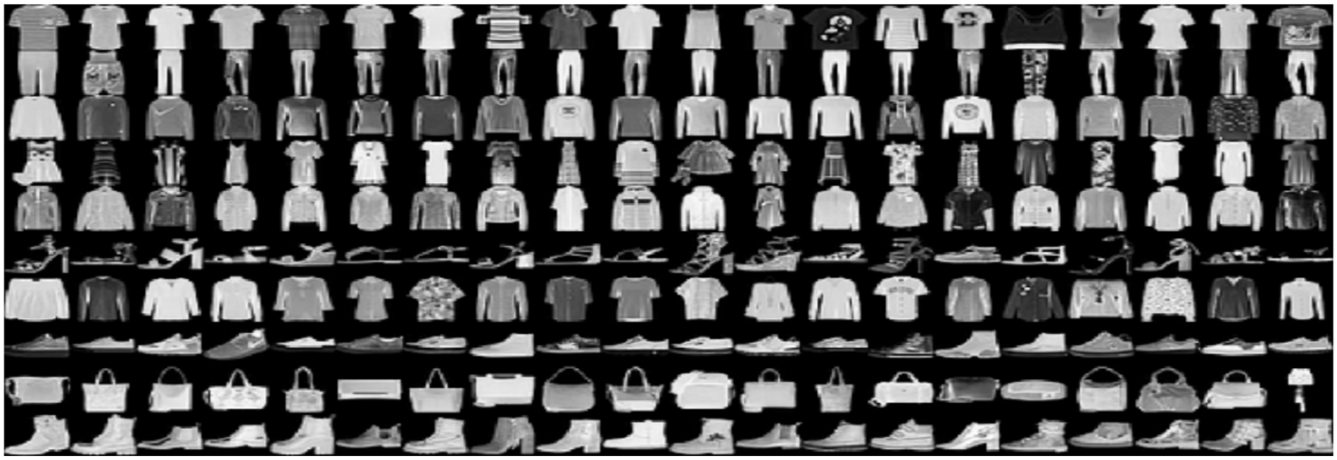


Figure 2. sample from dataset

sum of total numbers for actual label. F-score: is the harmonic average of precision and recall. F-score = 2RP/(R+P)

Average Accuracy (ACC) is the measurement method which is used to evaluate accuracy by comparing correct label with predicted label. The value of ACC represents the percentage of correct predicted labels, which means that the higher ACC value indicates better performance of measured algorithm. The formula of ACC is defined as:

$$Acc = \frac{number\ of\ correct\ classifications}{total\ number\ of\ test\ examples\ used} \quad (17)$$

$$Acc(Y, Y_{pred}) = \frac{1}{n}\sum_{i=1}^{n} 1\{Y_{pred}(i) == Y(i)\} \quad (18)$$

where Y_{pred} is predicted value by executing PCA and Logistic Regression algorithm. Y is the original labels which are correct. And n is the number of test data.

Measurement. There are four different situations occur when do prediction, which are TP(true positive), FP(false positive), TN(true negative) and FN(false negative).

1. TP(true positive): The positive class is judged to be a positive class;
2. FP(false positive): The negative class is judged as a positive class;
3. TN(true negative): The negative class is judged to be negative;
4. FN(false negative): The positive class is judged as negative class;

Confusion Matrix is a specific matrix used to visualize the performance of the algorithm. Normally, each column represents a predicted value, and each row represents the actual category.The correct number of predictions is shown on the diagonal of the matrix.

P(precision) is the rate of diagonal of the confusion matrix over sum of total numbers for predicted label. For each category, It calculates the ratio of all "correctly retrieved results (TP)" to all "actually retrieved (TP+FP)". The formula of P(precision) is defined as:

$$P = \frac{TP}{TP + FP} \quad (19)$$

R(recall) is the rate of diagonal of the confusion matrix over sum of total numbers for actual label. For each category , It calculates the ratio of all "correctly retrieved results (TP)" to all "results that should be retrieved (TP+FN)".The formula of R(recall) is defined as:

$$R = \frac{TP}{TP + FN} \quad (20)$$

F-score is the harmonic average of precision and recall.It is a balance of precision and recall.

Therefore, It is a more reasonable measurement. The formula of F-score is defined as:

$$R = \frac{2PR}{P + R} \tag{21}$$

where P is precision and R is recall.

3.3 Experiments

In this experiment, we first load the training data, do normalize, then apply PCA to reduce the dimension of the data, then apply one vs all method, that is for each label, we use scipy.optimize.minimize to find the appropriate Weight in Logistic Regression. For predicting, as sigmoid function is monotonically increasing, given a test data, we just need to find maximum z_i in Logistic Regression, which is $z_i = W_j X_i$ for $j = 0, 1, 2, ...9$, and then assign the label related to the W_j to the test data.

The pseudo-code for this program will be shown below as follows:

Algorithm 2 Classification Algorithm

```
function TRAIN(training data, training label, test data, test label)
    processed data ← Min-Max normalization(training data)
    processed data ← reduce dimension(processed data)
    model ← train with logistic regression(processed data, training label)
    predicted label ← predict(model, test data)
    test accuracy ← validate(predicted label, test label)
return test _accuracy
```

Dimension Influence. We measure Time and ACC value according to different k dimension after PCA to explore appropriate dimensions for the dataset when implementing the Logistic Regression algorithm. The result is shown in below:

Dimension(k)	Accuracy	Time(s)
784	83.25	807.26
600	83.45	634.09
500	83.85	191.26
400	84.1	115.32
300	84.35	71.55
275	84.5	68.89
265	84.89	66.59
260	85.15	63.67
255	84.8	55.89
250	84.65	56.28
240	84.25	51.01
220	84.25	49.43
200	83.95	39.63
150	84.25	26.06
100	84.25	16.18
90	83.75	12.51
80	84.05	10.46
75	83.35	10.20
50	82.45	6.00
40	81.8	4.91
30	80.4	3.91
20	78.75	2.25
10	73.55	1.22
5	67	0.78

Table 1. Time and Accuracy value according to different dimension dimension.

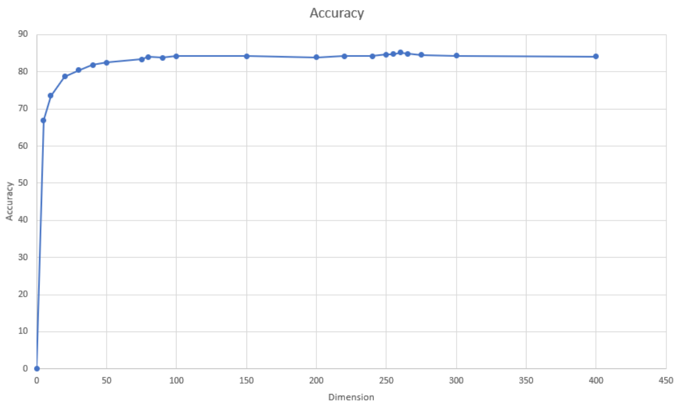


Figure 3. Accuracy-Dimension

Findings 1. As shown in Table 1, Figure 3 and Figure 4, four results can be obtained:

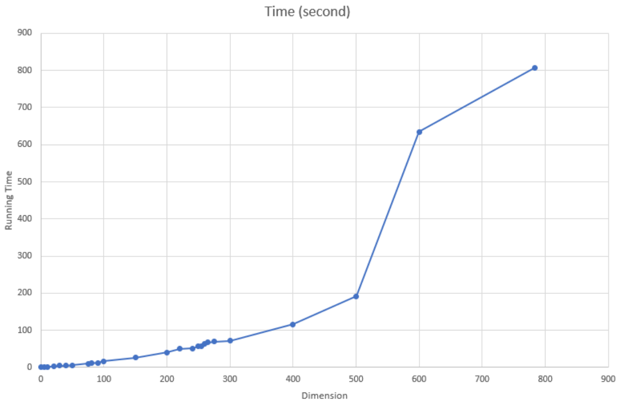


Figure 4. Time-Dimension

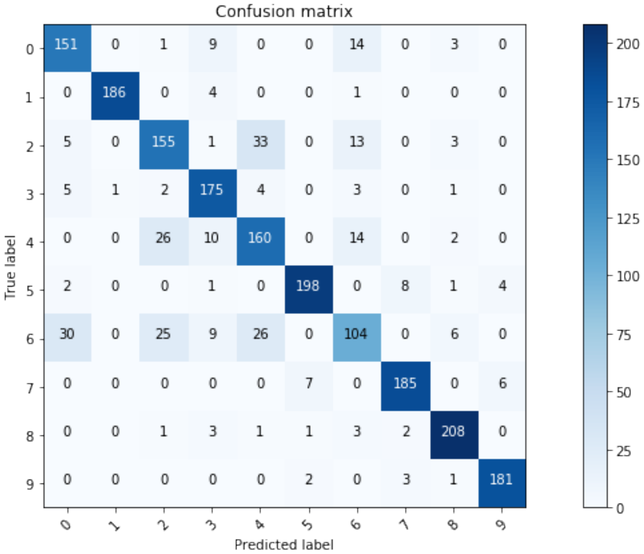
1. The highest accuracy 85.15 % be achieved when k is 260;

2. As the dimension decreases, the time of running algorithm is decreasing, which means cost less;

3. As the dimension decreases, the accuracy rate has been increasing until the 260 dimension, then it turns to decreases until 80 dimension,after 80 dimension, the accuracy still keeps decreasing;

4. 80 dimensions is economical.It can also achieve a good accuracy(84.05%) while cost(around 10 second) much less than 260 dimension(63.67 second).

Measurement. Confusion Matrix and F-score are used to measure the prediction performance on each category.In order to achieve best performance, the dimension k is chosen to be 260. Below is the result:



	0	1	2	3	4	5	6	7	8	9
0	151	0	1	9	0	0	14	0	3	0
1	0	186	0	4	0	0	1	0	0	0
2	5	0	155	1	33	0	13	0	3	0
3	5	1	2	175	4	0	3	0	1	0
4	0	0	26	10	160	0	14	0	2	0
5	2	0	0	1	0	198	0	8	1	4
6	30	0	25	9	26	0	104	0	6	0
7	0	0	0	0	0	7	0	185	0	6
8	0	0	1	3	1	1	3	2	208	0
9	0	0	0	0	0	2	0	3	1	181

Figure 5. Confusion Matrix

Label(L)	Precision	Recall	F-score
L0	0.782	0.848	0.814
L1	0.995	0.974	0.984
L2	0.738	0.738	0.738
L3	0.825	0.916	0.868
L4	0.714	0.755	0.734
L5	0.952	0.925	0.938
L6	0.684	0.52	0.591
L7	0.934	0.934	0.934
L8	0.924	0.95	0.937
L9	0.948	0.968	0.958

Table 2. Precision,Recall and F-score of each label in 260 dimension.

Findings 2. As shown in Table 2 and Figure 5, three results can be obtained:

1. The colour of confusion matrix diagonal is dark blue which means The prediction is pretty accurate;

2. F-score of Label-6 is much lower than others';

3. F-score of Label-2 and Label-4 are relatively lower;

3.4 Experiment Analysis

We conduct a deep analysis of experimental results in two aspects after the experiment.

Firstly, the impact of data dimensions on experimental results. As the dimension of the original data is large, so we perform a pre-processing step to reduce the amount of computation through implementing PCA method and then we find that, accuracy of prediction and time of applying Logistic Regression algorithm to train prediction model are influenced by the dimension of data. The time of running algorithm is decreasing as the dimension decreases. And, at the beginning of reduce data dimension, as the dimension decreases, the accuracy rate has been increasing until the 260 dimension, which is the peak of accuracy. Then it turns to decreases except on 80 dimension; These clearly show that the dimension of data influences the accuracy and cost(time) of training a prediction model through implementing Logistic regression algorithm.

Secondly, algorithm performance and the main cause of the prediction error. Through observing the experimental results of Confusion Matrix and F-score, we find the most of the labels can be predicted in an satisfactory accuracy as the accuracy and f-score of most of them are higher than 90% except the label-6. Half of label-6 data are wrongly predicted. Therefore, basically in the experiment create an satisfactory prediction model on the data set. But the model can not work well with Label-6 category.

4.Personal Reflection

Shortcomes in our method. In this experiment, we are assuming that z in Logistic Regression has the following format: $z_i = w_0 + w_1x_1 + w_2x_2 + ... + w_nx_n = X_iW$, which may not be true.
For one vs. all (or rest) method, the issue about imbalanced classes may affect the final result. If the training dataset is balanced, i.e. each class makes up 10% of the data, by doing one vs. all method, we would have a 10-90 distribution in training process. It may become a problem.

Future experiments. We may change the format of z in Logistic Regression, e.g. $z = W_1X + W_2X^2$ or a kernel function, and compare the result, then choose z with the best result.
Undersampling or oversampling techniques, which are used to adjust the distribution of data, may be used to deal with the issue of imbalanced classes. One vs. one method would be a good alternative for this experiment.
We may implement other classifier, e.g. K-nearest neighbours or Gaussian Naive Bayes, compare the results and choose the classifier with the best result. Voting could give us a better result, i.e. if we implement three different classifiers c1, c2, c3, given a single test data, c1 and c2 assign label 1 to it, c3 assign label 2, then we would predict it as label 1.

5. Conclusion

In this paper, we implement Logistic Regression algorithm to create a prediction model after implementing PCA method to reduce the dimension of data. Experiments on data set show that, accuracy and cost (time) of prediction are influenced by the dimension of data. For implementing Logistic Regression algorithm this data set, 260 dimensions should be chosen to achieve the highest accuracy(85.15%) and the choice of 80 dimensions is more economical which means it can also achieve a good accuracy(84.05%) while cost much less(around 10 second). What's more, generally, for this data set the Logistic Regression algorithm is a good choice as the accuracy is pretty high. The short-come of the algorithm is that the Label-6 category can not be well predicted.

References

[1] Bishop, C. M. (2006). Pattern recognition and machine learning. New York: Springer

[2] Cheng, W., Hüllermeier, E. (2009). Combining instance-based learning and logistic regression for multilabel classification. Machine Learning, 76(2-3), 211-225. doi:10.1007/s10994-009-5127-5

Appendix

1.Hardware and Software.

- Processor: 3.1 GHz Intel Core i5
- Memory: 8 GB 2133 MHz LPDDR3
- Coding tool: jupyter notebook
- Interpreter: Python 3.6.4
- Libraries: h5py, numpy, scipy, matplotlib.pyplot, sklearn

2.Data Set.

- images_testing.h5
- images_training.h5
- images_training.h5
- labels_training.h5

3.Code Instruction. Code is written in Python 3.6.
Use following command to run the code in terminal

```
python3 assignment.py
```

or

```
jupyter notebook assignment.ipynb
```

To load the unlabeled test data result, we use following command:

```
with h5py.File('../Output/predicted_labels.h5','r') as H:
    test_result = np.copy(H['label'])
```

4.contribution of group members. The code were written in team work as each person in charge of particular parts of code, also as the work of data collection and analysis. After all coding work finished, we wrote the report together through using Overleaf. All in a word, each group member did the same contribution.