Assignment 1: Image Classification Based on Logistic Regression

Group members: Dehong Liang(470188761), Tianzuo Zhang(470085460), Shanshan Luo(470349580)

Abstract— Nowadays, machine learning has become the important concept of the digitalization development which is applying the learning of specific task by machine. With the classification method has received considerable attention, we are looking for an appropriate algorithm to classify items into categories. In fact, by analyzing the performance of different kinds of algorithm, we finally choose the logistic regression algorithm as the main method to deal with this problem. By estimating the parameters of the logistic model and label the binary dependent variable by value 0 or 1, the good performance and less running can be gained combined with using maximum likelihood estimation and gradient descent algorithm.

Keywords: machine learning, image classifier, multiple classifiers, logistic regression, maximum likelihood estimation, gradient descent.

I. INTRODUCTION

With the population of artificial intelligence, machine learning which can automate analyze the data by building the analytical model has gained many attentions. Applying different algorithm to classify datasets in machine learning can gain different performance, which leads us to find the appropriate algorithm to deal with different targets classification. In our experiment, logistic regression is the best algorithm to classify targets into categories, which is also the main concept of our classification process. [1]

Application of logistic regression in machine learning area is widely spread, including image segmentation and categorization, geographic image processing, handwriting recognition and so on. As for our experiment, we apply logistic regression to divide the training data into 10 categories as the classification which contains the images of the size 28×28 which get the good performance, less cost and running.

From the concept view of point, with applying the logistic regression, sigmoid function is used as the important function which can label factors by 0 or 1. Then maximum likelihood estimation is used to construct a regression model, which is based on the optimization method to determine the best regression coefficient. By using the gradient descent method and choosing the appropriate learning rate, we can get the prediction of using test data after training the data and get the total number of errors, correct rate and cost time. [2]

From the python language view of point, the gradient decent function and cost function is defined at the beginning. With using one-vs-all method which can construct certain number of logistic classifiers and the vector set which is k*(n+1), training data is finally well compared by the "minimize" function to get the best theta number. After getting the best theta, we can make a good prediction by choosing the maximum rate of training samples with the error rate or correction rate.[3]

In this study, we compare the efficiency of the three kinds algorithm (k nearest neighbour, logistic regression and Bayes algorithm). the experimental results demonstrate that the logistic regression method is superior to those two algorithms.[4] To be concrete, the time cost of k nearest neighbour algorithm is triple than logistic regression, due to the character tic of k nearest neighbour algorithm which will compare the test datasets to all categories of training datasets and cause the time-consuming problem. The Bayes algorithm is not suitable in this study compared with using logistic regression algorithm because the feature vector in this study is difficult to extract by using this algorithm.

Because this experiment is to classify the category of sample data by using certain classifier to train datasets and get the result, we aim to be more familiar with several basic kinds of classification algorithm during this study and find the benefits and drawbacks towards applying different algorithms in the machine learning process. [5]

This study is very important for us to have a good understanding of different kinds of algorithm and how to apply it to make classification in practical application process. By comparing different algorithms, we can find the appropriate algorithm which is used with the result of high accuracy rate and less running time. What's more, our experiment can be used by others, which can provide a detailed description of basis classification for people and help them to have a better understanding of logistic regression algorithm and gradient descent and so on. Our coding can be the supplementary for people's learning of machine learning and practicing in python area.[6]

The whole experiment process is smart enough to reduce the amount of computation and maintain the accuracy rate of classification. While pre-processing

step is not used during our studying process because of the drawbacks of applying principle component analysis method (PCA), which will be discussed in the rest chapter.

II. METHODS

A. logistic regression

1) logistic regression

Logistic regression[7] is a relatively common machine learning method used in the data analysis to estimate the likelihood of something. For example, it can be used to predict the possibility of a user purchasing a certain product, the possibility that a patient has a certain disease, or the possibility that an advertisement is clicked by the user. In this study, it is mainly used to analyse the corresponding labels of pictures that belong to various classifications.

Logistic regression is a type of linear regression. As for regression, regression is actually estimating the unknown parameters of known formulas. For example, the known formula is

$y = a * x + b$, and the unknown parameters are a and b. We now have a lot of data, $(x, y) \in R$,as training samples. And regression is the use of these data to automatically estimate the values of a and b. The estimated method can be simply understood as that, given a training sample point and a known formula, for one or more unknown parameters, the machine will automatically enumerate all possible values of the parameter until the one that best fits the distribution of the sample points is found.

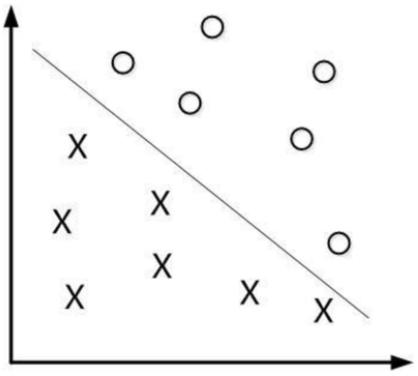


Figure 2.1.1-1 Logistic regression classifier schematic diagram

2) The derivation of logistic

Unlike linear regression, the sensitivity is consistent throughout the real number domain. Logistic regression is mainly used for classification, while the classification range needs to be in [0, 1]. Logistic regression is a regression model[8] that reduces the prediction range and limits the prediction value to [0,1]. The regression curve is shown in Figure 2.2.2-1.

The remainder of the paper is organized as follows: section 2 is the main method we used, section 3 is the experiment and result, section 4 is the discussion part and section 5 is the conclusion part.

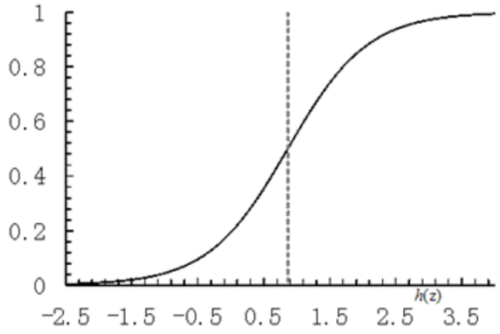


Figure 2.1.2-1 Logistic Regression curve

At the beginning of bipartition, the used function is the Heaviside step function, which can well represent the situation of the two classes. But this function jumps[9] from 0 to 1 at the jump point, and this instantaneous jump process[10] is sometimes difficult to handle. To avoid this, the researchers used another function with similar properties, the Sigmoid function, and it was mathematically easier to handle[11].

Figure 2.2.2-2 shows two graphs of the Sigmoid function at different coordinate scales. When x is 0, the value of the Sigmoid function is 0.5. As x increases, the corresponding Sigmoid value[12] will approach 1; and as x decreases, the Sigmoid value will approach 0. If the abscissa is large enough, the Sigmoid function looks a lot like a step function.

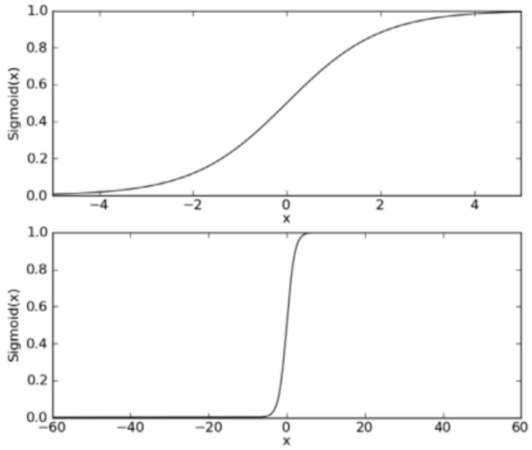


Figure 2.1.2-2 curves of sigmoid function at two coordinate scales

The specific calculation formula of the Sigmoid function is as follows:

$$g(z) = \frac{1}{1 + e^{-z}}$$

The input to the Sigmoid function is denoted by z and is derived from the following formula

$$z = \theta_0 x_0 + \theta_1 x_1 + \cdots + \theta_n x_n$$

And n is the number of features of the sample.

If vector is used, the above formula can be written as:

$$z = \theta^T X$$

It means that the two corresponding elements of the numerical vector are multiplied and then all added to obtain the z value. The vector X is the input data of the classifier[13], and the vector θ is the best parameter or coefficient that we want to find, making the classifier as accurate as possible.

When classifying, we can construct the following formula to fit the classification:

$$\begin{cases} P(y=1|X,\theta) = \dfrac{1}{1+e^{-\theta^T X}} \\ P(y=0|X,\theta) = \dfrac{1}{1+e^{-\theta^T X}} \\ \quad = 1 - P(y=1|X,\theta) \\ \quad = P(y=1|X,-\theta) \end{cases}$$

Make:

$$h_\theta(x) = g(\theta^T X) = \frac{1}{1+e^{-\theta^T X}};$$

$$g(z) = \frac{1}{1+e^{-\theta^T X}}$$

Next is to construct a regression model. This study uses maximum likelihood estimation[14] to construct a regression model.

First, for a single sample, the posterior probability is:

$$P(y|X,\theta) = \big(h_\theta(x)\big)^y (1 - \big(h_\theta(x)\big)^{1-y},$$

$$y \in (0,1)$$

Then, the maximum likelihood function is:

$$L(\theta|x,y) = \prod_{i=1}^{m} p\big(y^{(i)}|x^{(i)};\theta\big)$$

$$= \prod_{i=1}^{m} (h_\theta(x))^{y^{(i)}} (1 - h_\theta(x))^{1-y^{(i)}}$$

In order to reduce the computational complexity[15] of the calculation, it is processed logarithmically.

$$l(\theta) = \log\big(L(\theta|x,y)\big)$$

$$= \sum_{i=1}^{m} y^{(i)} \log\Big(h\big(x^{(i)}\big)\Big) + (1 - y^{(i)}) \log\big(1 - h(x^{(i)})\big)$$

3) Dtermination of the best regression coefficient based on the optimiation method

In 2.2.2, we obtained the regression model $h_\theta(x)$ used for classification. It can be seen from the above that if we want to obtain the best parameters of the logistic regression model $h_\theta(x)$, and it is equivalent to gain most value of loss function:

$$\theta^{\mathrm{T}} = \arg\min\big(l(\theta)\big)$$

In order to find the best coefficient to satisfy the data set, we use the gradient descent method.

3.1) gradient descent

The gradient descent method is based on the idea that to find the most value of a function[16], the best way is to search along the gradient direction of the function, because it is the most efficient way to find the best value. If the gradient is denoted by ∇[17], the gradient of the function f(x, y) is represented by:

$$\nabla f(x,y) = \begin{pmatrix} \dfrac{\partial f(x,y)}{\partial x} \\ \dfrac{\partial f(x,y)}{\partial y} \end{pmatrix}$$

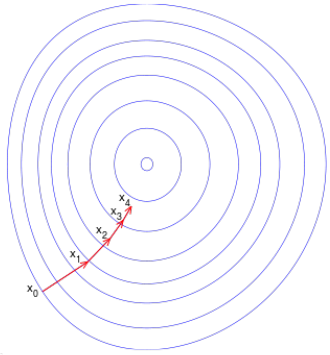


Figure 2.1.3.1-1 sample of the gradient descent

According to the logistic regression model, the partial derivative of θ is obtained by the loss function:

$$\frac{\partial}{\partial\theta_j}(l(\theta)) = \frac{\partial}{\partial\theta_j}(\sum_{i=1}^{m} y^{(i)}\log\left(h\left(x^{(i)}\right)\right) + (1-y^{(i)})\log\,(1 - h\left(x^{(i)}\right)$$

$$= \left(\frac{y^{(i)}}{h(x^{(i)})} - \left(1-y^{(i)}\right)\frac{1}{1-h(x^{(i)})}\right)\frac{\partial}{\partial\theta_j}\left(h\left(x^{(i)}\right)\right)$$

$$= \left(\frac{y^{(i)}}{g(\theta^T x^{(i)})} - \left(1-y^{(i)}\right)\frac{1}{1-g(\theta^T x^{(i)})}\right)$$

$$\frac{\partial}{\partial\theta_j}\left(g\left(\theta^T x^{(i)}\right)\right)$$

$$= \left(\frac{y^{(i)}}{g(\theta^T x^{(i)})} - \left(1-y^{(i)}\right)\frac{1}{1-g(\theta^T x^{(i)})}\right) g\left(\theta^T x^{(i)}\right)$$

$$\left(1-g\left(\theta^T x^{(i)}\right)\right)\frac{\partial\theta^T x^{(i)}}{\partial\theta_j}$$

$$= \left(y^{(i)}\left(1-g\left(\theta^T x^{(i)}\right) - \left(1-y^{(i)}\right)g\left(\theta^T x^{(i)}\right)\right)\right)x_j$$

$$= (y^{(i)} - h_\theta(x^{(i)}))x_j$$

The above description helped us find the direction in which the function value changes the fastest, but it does not involve the amount of movement.[18] In machine learning, it is called the learning rate. For minimizing the optimization problem, it is only necessary to advance the parameter by one step[19] in the opposite direction of the gradient to achieve the reduction of the objective function. When we select a learning rate for , its update formula becomes:

$$\theta \leftarrow \theta - \eta \cdot \nabla_\theta l(\theta), \eta: learning\ rate$$

The choice of learning rate is a key question, because when the learning rate is too large, it will always oscillate when looking for the best value, and the convergence speed will be very slow. When the learning rate is too small, its movement is too short, and the speed of convergence will be also slow.

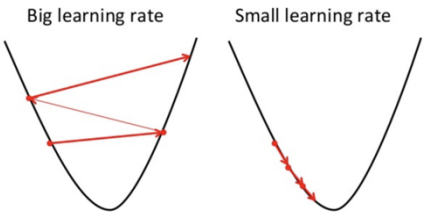


Figure 2.1.3.1-2 different speed of convergence with different learning rate

Corresponding to the partial derivative of the loss function of the above logistic regression, its update formula is:

$$\theta_j \leftarrow \theta_j + \eta(y^{(i)} - h_\theta(x^{(i)}))x_j^{(i)}$$

B. Some transformations for logistic regression

In this experiment, some transformations for logistic regression are used. The traditional logistic regression only applies to the two classifications. For this experiment, the number of classifications we need to complete is ten categories, that is, the sample set has ten label values. In this regard, ordinary logistic regression is not applicable. In order to achieve multiple classifications, we have made some modifications to the traditional algorithms. One-vs-all method is popular method applied to effectively transform binary classifier to multi-class classifier. When we implement the classifier, we set each class to the current class (1), and then the remaining nine classes are considered to be a class (0). So that after ten times are achieved, ten categories can be obtained. Calculate the class probability of each class on each training sample and select the highest probability as the predicted number label for the samples. In this way, samples are classified.

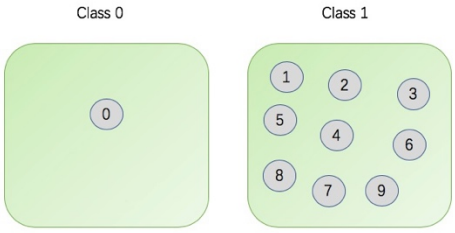


Figure 2.2-1 sample of the classifiers

III. EXPERIMENTS AND RESULTS

A. accuracy

Before analyzing the accuracy of this experiment, we first look at the information of the three tables, which are the number of samples in the ten sets of the training set and the real tags of the test set samples. And the number of errors in the test set sample label after the model is judged.

Class	0	1	2	3	4
Number	3011	2956	3020	3002	3029
Class	5	6	7	8	9
Number	3028	2967	2895	3002	3090

Table 3.1-1 label numbers of training data set; training data total number:30000

Class	0	1	2	3	4
Number	178	191	210	191	212
Class	5	6	7	8	9
Number	214	200	198	219	187

Table 3.1-2 actual-label numbers of test data set; test data total number:2000

Class	0	1	2	3	4
Number	58	15	60	62	62
Class	5	6	7	8	9
Number	194	156	42	20	4

Table 3.1-3 false-labels number of test data set after classifier; test data total number:2000, error number: 675

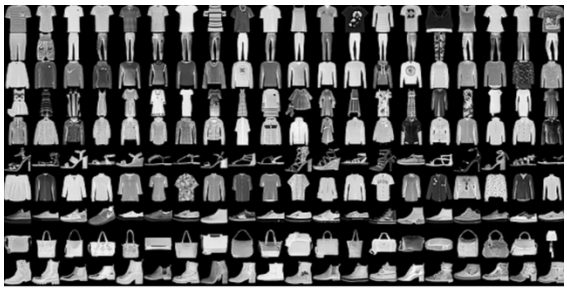


Figure 3.1-1 examples illustrating the data examples, each class takes one row

1) confusion matrix

In the field of machine learning, confusion matrices are commonly used analytical tables. It enables visualization of the performance of the algorithm. Its effect on reviewing the performance of the classifier is excellent because it makes it easy to see if the classifier confuses different classes.

		Actual class		
		Positive	Negative	Total
Predicted class	Positive	a(TP)	b(FP)	a + b
	Negative	c(FN)	d(TN)	c + d
	Total	a + c	b + d	N

Table 3.1.1-1 CONFUSION MATRIX

TP	true positive
TN	true negative
FP	false positive
FN	false negative

Table 3.1.1-2 meaning of TP, TN, FP and FN

Some Cost-Sensitive Measures as:

positive predictive value:

$$\text{Precision(p)} = \frac{a}{a+c}$$

true positive rate:

$$\text{Recall(r)} = \frac{a}{a+b}$$

F1 score is the harmonic mean of precision and sensitivity:

$$\text{F} - \text{measure(F)} = \frac{2a}{2a+b+c}$$

Accuracy (ACC)

$$\text{ACC} = \frac{a+d}{N}$$

2) result

According to the training data set, it takes 2min 30.6660 seconds to get the logistic regression model. And it just uses 4.0660 seconds to predict the labels of test samples. There are some table to show the precision, recall and F-measure.

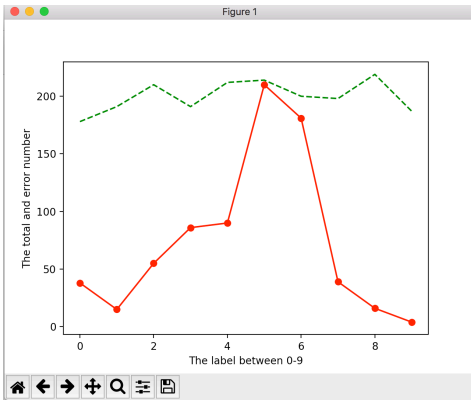
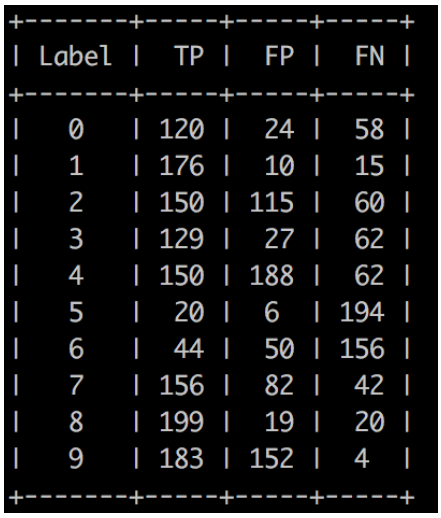


Figure 3.1.2-1 The total and error number of class 0-9

TP	FP	FN	TP rate
1327	673	673	0.6635

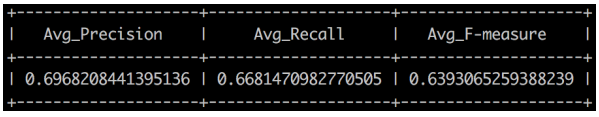
Figure 3.1.2-2 TP, FP, FN and TP rate of the test data set



Label	TP	FP	FN
0	120	24	58
1	176	10	15
2	150	115	60
3	129	27	62
4	150	188	62
5	20	6	194
6	44	50	156
7	156	82	42
8	199	19	20
9	183	152	4

Figure 3.1.2-2 TP, FP, FN of the different classes

Figure 3.1.2-3 Precision, Recall, F-measure of the different classes



Avg_Precision	Avg_Recall	Avg_F-measure
0.6968208441395136	0.6681470982770505	0.6393065259388239

Figure 3.1.2-4 Avg_Precision, Avg_Recall and Avg_F-measure of the whole data set

With these figures, we can get the accuracy of our logistic Regression model: 66.35%.

B. externsive analysis

According to the training model of the logistic regression we made, the accuracy of the model is 66.35%. This number is more than 50 percent but the benchmark of these data set is 89%, so our model is not accurate enough.

In order to show the advantages and the disadvantages of our model more directly, we compare the accuracy, cost and running time of k nearest neighbor algorithm and SVM. We conduct the model of k nearest neighbor algorithm by ourselves and conduct the model of SVM by importing libraries.

	Logistic Regression	K nearest neighbor	SVM-one vs all
Accuracy	66.35%	83.40%	64.80%
Running time	2min 33’	8min 40’	65min 30’
Cost	+	+	+++

Table 3.2-1 three models of classifiers

By comparing these three indicators, we decide to use the logistic regression model. As for the reasons, at first we would not use the SVM-one vs all model which take a long time to run and cost too much RAM. And then, although the accuracy of K nearest neighbor model is high and the running time is acceptable, it will waste much running time with extended test data set. Using K nearest neighbor algorithm, we need to compute the distance between the sample of test dataset and training dataset, which is very time consuming if the number of test dataset is too big. That is the reason why we choose to use the logistic regression model. What’s more, it just use most of the running time to get the model instead of spending the time on testing the test dataset.

IV. DISCUSSION

Before experiment, our team discussed some method to implement this classification task, Firstly, we thought of KNN algorithm, it’s relatively simple method for classification that support the multiple classes, but we worried about its efficiency while running the classifier, even though it has a high accuracy.

Another method we came up with is naive Bayes classifier method. It based on applying Bayes’ theorem with strong (naive) independent assumptions between the features. But after a careful research and discussion, we found it’s not convenient to abstract feature from pictures for Bayes, it could produce a large set of futures which present weekly for certain class. Meanwhile, we found the naive Bayes method is more suitable for processing the language. So, we almost abandoned this method before experiment.

Finally, the Logistic regression was proposed. But we encountered an issue, the logistic regression is primitively used to model a binary dependent variable, we need to come up with a way to change it for several classes, there’s a draft method called “random decision forests” that we can use, but we need more research and experiences to verify this method.

During the experience, in the beginning, we implements the KNN method and test with the half training dataset, with half test dataset, we found it’s really time-consuming. Without compression the

dataset, we cost almost 5 minutes to finish the "half-number" classification but we noticed the accuracy of the test can be up to 83 percent, we all think it's pretty high.

Sequently, one member proposed ways to reduce the dimension of dataset, there are two methods to choose SVD (Singular Value Decomposition) and PCA (Principal Component Analysis), Formally, the SVD of an m×n real M is a factorization of the form:

$$M = U \sum V^T$$

Where U is an m×m real matrix , Σ is an m×n rectangular diagonal matrix with nonnegative real numbers on the diagonal, and V is an n×n real or complex unitary matrix. The diagonal entries Σ are known as the singular value of M. The m columns of U and the n columns of V are called the left-singular vectors and right-singular vectors of M respectively.

PCA seeks a linear combination of variables such that the maximum variance is extracted from the variables. It then finds a second linear combination which explains the maximum properties of the remaining variance, and so on. Any principal component accounts for as much of the variability in the data as possible, and the next principal component should be orthogonal with the last principal component.

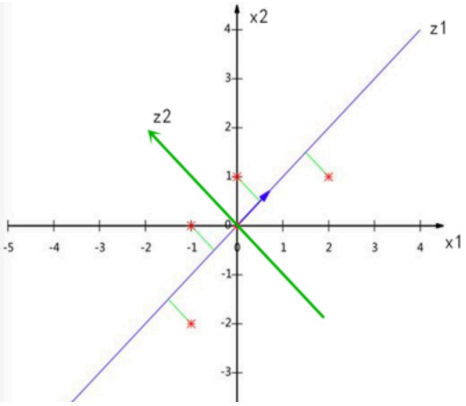


Figure 4-1 PCA decomposition schematic diagram

Actually, PCA is the simplest of the true eigenvector-based multivariate analyses. During the discussion we found that PCA is a special situation of SVD and its essence is similar, under this theory, we decided to choose PCA in our task.

Next, is about Logistic Regression implementation. First of all, we discussed about how to make this method to be suitable for multiple classes. Here's our solution: every time, we view one class as 1, and the rest of entities of other classes as 0, so we need to revalue the label list y, which should only contain 0 and 1, by doing this, we can get a theta for the certain class in y, that's the reason why we have total ten theta as last. For example, in the graph below, the circle points represent the entities of certain label, and the cross points represent the rest entities.

Another thing that needed to mention is the variable "learningRate" we have in our codes. We use this variable in both "cost" and "gradient" function in order to control the speed to descent. Changing it to different value could bring different outcome and time cost, in fact, after several experiments we found a good value between 1 to 1.5, which balance the performance and accuracy.

V. CONCLUSION

After using various method and adding compression in experiments we found that the Logistic Regression is relatively balanced algorithm for this task considering the performance and accuracy. Because the KNN is time-consuming and the Bayes algorithm is not suitable for this task. In the experiment, the final accuracy can reach 65percent at best, and cost a little over 2 minutes, that's why we choose Logistic Regression to implement the classifier.

The reason why we don't use compression in our task is that we think it's not that slowly using our method and it would reduce the accuracy drastically while using PCA compression before classification, for example, if we use the compression in KNN algorithm, and we reduce the dimension from 784 to 300 the accuracy could drop to nearly 50 percent, which we think is unacceptable. Likewise, if we compress the dimension to 300 in Logistic Regression, the accuracy could drop to below 45 percent.

REFERENCES

[1] Rodrigo Fernandes de Mello, M. A. P. "Machine Learning: A Practical Approach on the Statistical Learning Theory."

[2] Menard, S. (2010). "Logistic Regression: From Introductory to Advanced Concepts and Applications."

[3] Jake Lever, M. K., Naomi Altman (2016). "Logistic regression."

[4] Rice, D. M. (2013). "Calculus of Thought."

[5] Kuhle, S., Maguire, Bryan, Zhang Hongqun, Hamilton David, Allen Alexander C (2018). "Comparison of logistic regression with machine learning methods for the prediction of fetal growth abnormalities: a retrospective cohort study."

[6] Jebara, T. (2004). "Machine learning : discriminative and generative."

[7] Hosmer Jr D.W., Lemeshow S., Sturdivant R.X. Applied Logistic Regression John Wiley & Sons (2013)

[8] Andrew Y Ng, M. I. J. (2002). "On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes."

[9] Chikayama, E. (2014). "Decomposition of multivariate function using the Heaviside step function."

[10] Nikolay Kyurkchiev, S. M. (2016). "On the Hausdorff distance between the Heaviside step function and Verhulst logistic function." Journal of Mathematical Chemistry **54**(1).

[11] ZhixiangChen, F. (2009). "The approximation operators with sigmoidal functions." **58**(4).

[12] DaniloCostarelli, R. (2013). "Approximation results for neural network operators activated by sigmoidal functions." **44**.

[13] MICHA OFIR, J. K. (2003). "Variation in Onset of Summer Dormancy and Flowering Capacity Along an Aridity Gradient in Poa bulbosa L., a Geophytic Perennial Grass."

[14] Eliason, S. R. (1993). "Maximum Likelihood Estimation."

[15] Millar, R. B. (2011). "Maximum Likelihood Estimation and Inference: With Examples in R, SAS and ADMB."

[16] Olivier Chapelle, M. W. (2010). "Gradient descent optimization of smoothed information retrieval metrics." **13**(3).

[17] Shao-Bo Lin, D.-X. Z. (2018). "Distributed Kernel-Based Gradient Descent Algorithms." **47**(2).

[18] J. Fliege, A. I. F. V., L. N. Vicente (2018). "Complexity of gradient descent for multiobjective optimization."

[19] Thomas Villmann, S.0 H., Marika Kaden (2013). "Kernelized vector quantization in gradient-descent learning."