SVM for digit number recognition

kai ZHANG, mengyu PAN

October 2020

1 Introduction

The MNIST database of handwritten digits has a training set of 60,000 examples, and a test set of 10,000 examples. It is a subset of a larger set available from NIST. The digits have been size-normalized and centered in a fixed-size image[1].

In this experiment, we use the MNIST dataset to train a SVM classifier and predict the corresponding classes for the test dataset.

2 Method

At first, we evaluate our dataset. In the dataset, there are a number of features with huge data. In this case, we need to reduce dimensionality and we choose principal component analysis (PCA). Its principal is to find the main part of the dataset to represent the original one. It uses the linear transformation to reduce the dimension of the original data.

Then we use the support vector machine (SVM) to classify our data. This classifier aims to find the biggest margins between each class. For example, there are two types of data in two dimensions. SVM not only draws a line to separate them but also calculates the margin of the line by setting the support vector and choose the optimal one with the biggest margin. It can also be used to classify non-linear data as it has different kernels as linear, RBF, polynomial, and so on.

However, SVM has many different parameters. With different parameters, the result differs. In order to find the best result with proper parameters, we use GridSearch. We give a range of each parameter and the GridSearch will compare the result of SVM using the different parameters in the range and it will choose the best result with proper parameters. It takes much time to calculate the result of each combination but it is simple to understand.

3 Result

We evaluated the SVM method in the MNIST dataset, in which 60000 samples were used for training and 60000 samples for testing. For each sample, it has 784

features originally. To reduce the number of features, we used PCA to reduce the number of features from 784 to 100. To find the best parameters for SVM classification, we implemented the search method, called GridSearchCV, to test each pair of parameters and search the one with the best results.

As shown in Table 1, we tested three types of kernels, three different C values, and two types of gamma. Finally, after the GridSearch, we picked the best parameter pair and made a prediction, as shown in Table 2. The corresponding confusion matrix is illustrated in Table 3, from which we can see that for class 7, the SVM algorithm made the most mistakes.

params	kernel	C	gamma
value	linear	1	auto
	poly	5	scale
	rbf	10	

Table 1: Parameters of SVM algorithms

kernel	C	gamma	accuracy
rbf	10	auto	0.998

Table 2: Results of best prediction

class		True class									
		0	1	2	3	4	5	6	7	8	9
Predict Class	0	5920	0	0	0	1	1	0	0	1	0
	1	0	6737	3	0	0	0	0	2	0	0
	2	0	0	5946	3	1	1	1	3	1	2
	3	0	0	4	6114	0	6	0	2	4	1
	4	0	0	1	0	5837	0	0	2	0	2
	5	0	0	0	4	1	5411	1	0	4	0
	6	0	0	0	0	1	2	5915	0	0	0
	7	0	1	3	0	3	1	0	6254	2	1
	8	0	0	0	4	1	1	0	1	5844	0
	9	2	0	1	2	4	1	0	11	3	5925
sum error		2	1	12	13	11	13	2	21	15	6

Table 3: Confusion matrix of prediction results

4 Code

The code of this experiment is in `https://github.com/SummerOf15/machine-learning-in-robots/tree/master/SVM`

References

[1] Yann LeCun et al. “Gradient-based learning applied to document recognition”. In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324.