Hand-written Digits Classification and Letter Recognition

Group 10
Sixiang Ma, Yuan Xiao, Xiaokuan Zhang

12/09/2016

Abstract

say something

1 Introduction

1.1 Problem Description

Our group wants to solve the problem of classifying hand-written digits and black-and-white rectangular pixel-displayed letters. We have two datasets from UCI machine learning repository [10]. One is hand-written digits dataset [1]. It has 5620 instances. For each instance, there are 1024 attributes (32x32 matrix), whose values are either 0 or 1. After grouping every 4x4 blocks, the dimension is reduced to 64 (8x8 matrix), and each attribute ranges from 0 to 16. Another is letter recognition dataset [2]. The character images were based on 20 different fonts. It has 20000 instances. For each instance, there are 16 attributes, whose values range from 0 to 15. More details can be found by visiting the links.

1.2 Classifying Algorithms

In this report, we test three machine learning algorithms, i.e., Support Vector Machine (SVM), Neural Networks, and Naive Bayes. We will present the detail of algorithms in Section 3.

1.3 Result Summary

For the first dataset, after tuning the parameters, the three classifiers can achieve , , , accuracy.

2 Background

The hand-written digit dataset was first used in [9]. In the paper they combined different classifiers to obtain a better performance. The letter recognition dataset was first used in [6]. They generated classification rules to distinguish different letters.

3 Methodology

3.1 Support Vector Machine

Support vector machines (SVMs) [4] are supervised learning models that analyze data for classicifation or regression. Given a set of training examples, which are marked with their belonging categories, a SVM algorithm builds a model to recognize and assign testing examples to the predicted categories.

(Say something about kernel functions).

3.2 Neural Networks

Neural networks, or artificial neural networks [7], simulate the functions of nerve cells of human brain and serve as an important computational approach in machine learning. They typically form a structure of multiple layers of basic perceptrons and support both supervised and unsupervised learning.

Neural networks have a long history, dating back to the 1940s [11]. However, the idea of artificial neural networks was not popular at early days due to its limitation in solving logical calculations [12]. Modern neural networks revived in the past decade, along with the rise of deep learning [3, 13].

3.3 Naive Bayes

Naive Bayes classifier makes use of the Bayes Theorem. It is basically a conditional probability model. It is one of the simplest machine learning algorithms. Compared to Bayesian Networks, Naive Bayes is technically a special case by assuming that all features are conditionally independent from each other given the class label. One of the earliest papers that described this algorithm was from 1970s [5].

4 Evaluation

In this project, we mainly used `Weka` [8] to test different machine learning algorithms. For each algorithm, we use different parameter settings, which will be covered in this section.

4.1 Hand-written Digits Classification

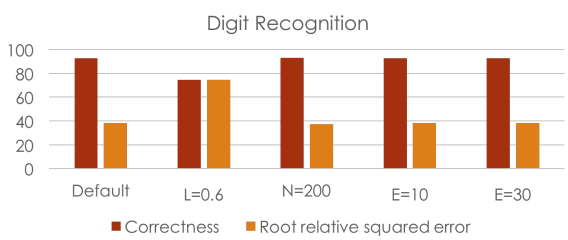
4.1.1 Support Vector Machine

4.1.2 Neural Networks

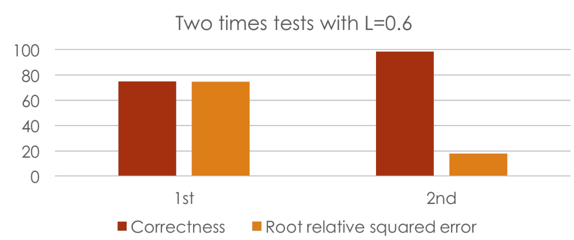
In order to understand how the many parameters influence the performance of multilayer perceptron in classification, we decide to change only one parameter at a time. The default setting is "-L 0.3 -M 0.2 -N 500 -V 0 -S 0 -E 20 -H a". And the parameters that we are interested in are learning rate, training time and validation threshold.

Figure 1a shows the average performance for different parameter combinations. For default settings, we have the average accuracy of 92.73%. And the root relative squared error is as low as 38.53%. Changing validation threshold to either more or less does not affect the performance at all. And if we let training time be less, the overall accuracy lowers a little, which matches our expectation.

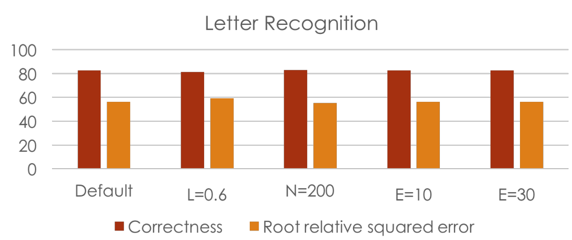
It can be easily noticed that changing learning rate to 0.6 affects the performance heavily. Thus we tested with the same parameter combination again. In Figure 1b, when we ran the test again, the accuracy grew to 98.36%. Such inconsistency in the test result showed that the performance of multipayer perceptron is influenced by stochastic. If we increase the stochastic by modifying the parameters, the fluctuation grows.



(a) Digit Classification



(b) L=0.6 in Digit Classification



(c) Letter Recognition

Figure 1: Performance of Artificial Neural Networks

4.1.3 Naive Bayes

The Naive Bayes works well on the Hand-written Digit dataset. In Figure 2, K means whether to use kernel estimator, and D means whether to use supervised discretization. In Figure 2a, when K=F, D=F, which is the worst case scenario, the accuracy and F-measure are still higher than 90%. In the best case (K=T, D=F), they both exceeds 92%.

4.2 Letter Recognition

4.2.1 Support Vector Machine

4.2.2 Neural Networks

The overall performance of multilayer perceptron is quite stable. The accuracy is around 82% and the root relative squared error is around 56%. However, there are some classes with relatively low accuracy making the average performance not so good as digit recognition. Figure 1 shows the least correct classes: ‘G’, ‘H’ and ‘S’.

Neural networks do a better job than the other two algorithms in recognizing ‘O’ and ‘Q’ and it can correctly tell ‘X’ apart from ‘S’. Changing learning rate higher helps the performance with ‘H’.

Parameters	Class (Letters)	Accuracy
Default	G	69%
	H	64.4%
	S	65%
L = 0.6	G	68.2%
	H	69.1%
	S	64%
N = 200	G	69.3%
	H	65.1%
	S	64.4%

Table 1: Worst Cases: Multilayer Perceptron

4.2.3 Naive Bayes

Naive Bayes does not work well on the Letter Recognition dataset. From Figure 2b, we can see that the highest accuracy is lower than 75%. Generally speaking, Naive Bayes is not suitable for classifying letters.

In this dataset, we are also interested in which letters Naive Bayes performs worst. Table 2 shows the top 3 worst cases when using Naive Bayes with different parameters. In all three parameter settings, ‘H’ is always one of the top 3, which means that ‘H’ is quite hard to classify for Naive Bayes. Also, the same applies to ‘S’ and ‘X’, as they appear in two of the three cases.

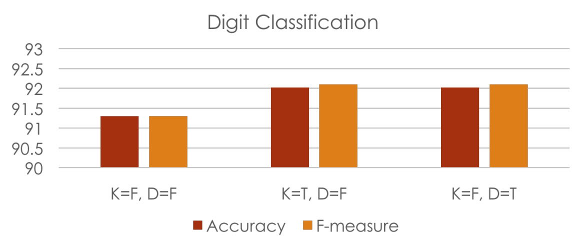
5 Discussion

5.1 Evaluation Matrix

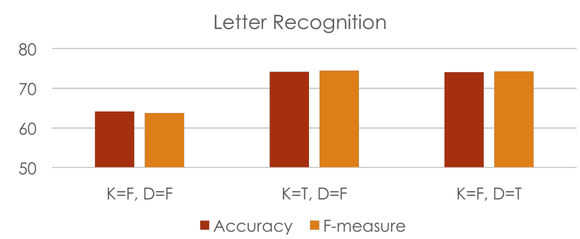
In our presentation, we used `root relative squared error` to evaluate the performance of our algorithms and falsely claimed that SVM was not suitable for our datasets. However, it makes little sense to evaluate this feature on non-binary datasets. So in the report, we changed it to F-measure.

Parameters	Class (Letters)	Accuracy
K = F, D = F	S	29.4%
	H	30.5%
	Y	33.1%
K = T, D = F	H	57.4%
	S	64.2%
	X	64.3%
K = F, D = T	H	57.5%
	E	60.7%
	X	64.4%

Table 2: Worst Cases: Naive Bayes



(a) Digit Classification



(b) Letter Recognition

Figure 2: Performance of Naive Bayes

5.2 Different Kernels of SVM

In our presentation, we only talked about the polynomial kernel. In the report, we added xxx, xxx kernels to compare the performance. By comparison, we can see that ......

5.3 Parameters of Multilayer Perceptron

Multilayer Perceptron has a number of parameters. In the project, we would like to look into the following specific parameters: learning rate, training time and validation threshold. Other parameters include momentum, seed, nominalToBinaryFilter, hiddenLayers etc.

Learning rate (-L) stands for the amount the weights are updated. Training time (-N) is the number of epochs to train through. If the validation set is non-zero then it can terminate the network early. Validation threshold (-E) is used to terminate validation testing. The value here dictates how many times in a row the validation set error can get worse before training is terminated.

In our tests, the validation threshold does not affect the performance.

6 Conclusion

References

[1] Optical recognition of handwritten digits data set. `http://archive.ics.uci.edu/ml/datasets/Optical+Recognition+of+Handwritten+Digits`.

[2] Optical recognition of handwritten digits data set. `https://archive.ics.uci.edu/ml/datasets/Letter+Recognition`.

[3] Yoshua Bengio. Learning deep architectures for ai. *Foundations and trends® in Machine Learning*, 2(1):1–127, 2009.

[4] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[5] Richard O Duda, Peter E Hart, et al. *Pattern classification and scene analysis*, volume 3. Wiley New York, 1973.

[6] Peter W Frey and David J Slate. Letter recognition using holland-style adaptive classifiers. *Machine Learning*, 6:161, 1991.

[7] Martin T Hagan, Howard B Demuth, Mark H Beale, and Orlando De Jesús. *Neural network design*, volume 20. PWS publishing company Boston, 1996.

[8] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H Witten. The weka data mining software: an update. *ACM SIGKDD explorations newsletter*, 11(1):10–18, 2009.

[9] C Kaynak. Methods of combining multiple classifiers and their applications to handwritten digit recognition. Master's thesis, Institute of Graduate Studies in Science and Engineering, Bogazici University, 1995.

[10] M. Lichman. UCI machine learning repository, 2013.

[11] Warren S McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133, 1943.

[12] Marvin Minsky and Seymour Papert. Perceptrons: An introduction to computational geometry (expanded edn), 1988.

[13] Jürgen Schmidhuber. Deep learning in neural networks: An overview. *Neural Networks*, 61:85–117, 2015.