

AN EMPIRICAL COMPARISON OF SUPERVISED LEARNING IN HANDWRITTEN DIGITS CLASSIFICATION

Yage G. Xin

A17304338

COGS 118A Final Report

yaxin@ucsd.edu

ABSTRACT

In this paper, we used some of the supervised machine learning algorithms reported in “An Empirical Comparison of Supervised Learning Algorithms”. We experimented with different classifiers on several datasets, all designed for the handwritten digits classification problem.

1 INTRODUCTION

...

2 METHODOLOGY

2.1 LEARNING ALGORITHMS AND TRAINING PROCEDURE

We compare the performance within and across the bagging classifiers using the following procedure. We use three classifiers, experimented on three different splits of training and testing set size: 20/80, 50/50, 80/20, with respect to four different datasets. In each experiment, we fine tune these classifiers using 3-fold cross validation on the training set to find the best hyper-parameters. Then we use the whole training set to train the classifier and report the metrics on the test set.

2.1.1 LOGISTICS REGRESSION

We search different regularization term C from the range $\{0.001, 0.01, 0.1, 1, 10\}$.

2.1.2 DECISION TREE

For comparison sake, we only fine tune the maximum depth parameter in the decision tree once. The parameter is set to 5 by cross validation on the first dataset, using a 50/50 training/testing split from the range of $(1, 11)$. By fixing the maximum depth parameter, we ensure that these decision trees used in different experiments are the same algorithm.

2.1.3 LINEAR SVM

We search different regularization term C from the range $\{0.001, 0.01, 0.1, 1, 10\}$.

2.2 DATASET

To evaluate how the classifiers respond to changes in task difficulty, we chose four handwritten-digit datasets that differ systematically in structure and complexity. These datasets are chosen in order to provide a consistent task across different sources.

Using multiple datasets allows us to test how well models generalize across variations in handwriting styles, digit resolution, and dataset size. It also provides a more robust assessment than relying on a single dataset, as models might overfit to specific dataset characteristics. By including all four

Table 1: CV mean performance for decision tree

Dataset	Split	C	CV
MNIST	0.2/0.8	-	0.9627 ± 0.0038
MNIST	0.5/0.5	-	0.9620 ± 0.0022
MNIST	0.8/0.2	-	0.9624 ± 0.0011
Pen	0.2/0.8	-	0.9763 ± 0.0043
Pen	0.5/0.5	-	0.9780 ± 0.0022
Pen	0.8/0.2	-	0.9779 ± 0.0012
Semeion	0.2/0.8	-	0.9067 ± 0.0239
Semeion	0.5/0.5	-	0.9250 ± 0.0192
Semeion	0.8/0.2	-	0.9296 ± 0.0099
Optical	0.2/0.8	-	0.9462 ± 0.0133
Optical	0.5/0.5	-	0.9629 ± 0.0111
Optical	0.8/0.2	-	0.9685 ± 0.0082

Table 2: CV mean performance for logistic regression

Dataset	Split	C	CV
MNIST	0.2/0.8	0.01	0.9776 ± 0.0015
MNIST	0.5/0.5	0.01	0.9813 ± 0.0006
MNIST	0.8/0.2	0.01	0.9825 ± 0.0006
Pen	0.2/0.8	1.00	0.9527 ± 0.0027
Pen	0.5/0.5	10.00	0.9575 ± 0.0012
Pen	0.8/0.2	10.00	0.9603 ± 0.0013
Semeion	0.2/0.8	10.00	0.9468 ± 0.0095
Semeion	0.5/0.5	0.10	0.9593 ± 0.0037
Semeion	0.8/0.2	0.10	0.9609 ± 0.0046
Optical	0.2/0.8	10.00	0.9629 ± 0.0056
Optical	0.5/0.5	10.00	0.9690 ± 0.0035
Optical	0.8/0.2	0.10	0.9721 ± 0.0026

datasets, we can compare model performance across datasets of varying difficulty, evaluate hyperparameter effects consistently, and draw conclusions that are not dataset-specific but generalizable to the handwritten digit recognition problem.

We converted the multi-class digit labels into a binary classification problem. Specifically, the labels are converted from 0-9 to “7” and “not 7”. This allows us to examine model performance on a simplified task. This binary conversion also facilitates comparison of model behavior across datasets, highlighting differences in learning patterns and sensitivity to hyperparameters.

2.2.1 OPTICAL RECOGNITION OF HANDWRITTEN DIGITS

... For the rest of the report, this dataset is referred to as “Optical”.

2.2.2 MNIST

3 EXPERIMENTAL RESULTS

3.1 HYPERPARAMETERS

3.2 TEST ACCURACY

... Due to missing prediction data during the runs, F1 scores could not be computed for some models.

Table 3: CV mean performance for SVM

Dataset	Split	C	CV
MNIST	0.2/0.8	0.01	0.9756 ± 0.0018
MNIST	0.5/0.5	0.01	0.9819 ± 0.0006
MNIST	0.8/0.2	0.01	0.9832 ± 0.0006
Pen	0.2/0.8	1.00	0.9609 ± 0.0026
Pen	0.5/0.5	1.00	0.9625 ± 0.0013
Pen	0.8/0.2	1.00	0.9657 ± 0.0017
Semeion	0.2/0.8	0.10	0.9505 ± 0.0100
Semeion	0.5/0.5	0.01	0.9662 ± 0.0040
Semeion	0.8/0.2	0.01	0.9683 ± 0.0037
Optical	0.2/0.8	0.10	0.9738 ± 0.0066
Optical	0.5/0.5	1.00	0.9715 ± 0.0052
Optical	0.8/0.2	0.01	0.9842 ± 0.0036

Table 4: Test accuracy for split 0.2/0.8

Model	MNIST	Optical	Pen	Semeion
DecisionTree	0.964	0.958	0.976	0.911
LogisticRegression	0.982	0.991	0.979	0.968
SVM	0.983	0.988	0.981	0.970

3.3 OVERFITTING ANALYSIS

3.4 DATASET DIFFICULTY

4 DISCUSSION AND CONCLUSION

A APPENDIX

A.1 TRAINING AND VALIDATION ACCURACY

A.2 SOURCE CODE

The code for our experiments is available at https://github.com/yagregx/COGS118A_final.

Table 5: Test accuracy for split 0.5/0.5

Model	MNIST	Optical	Pen	Semeion
DecisionTree	0.963	0.969	0.981	0.921
LogisticRegression	0.984	0.992	0.982	0.980
SVM	0.984	0.993	0.983	0.979

Table 6: Test accuracy for split 0.8/0.2

Model	MNIST	Optical	Pen	Semeion
DecisionTree	0.963	0.977	0.979	0.947
LogisticRegression	0.984	0.991	0.982	0.983
SVM	0.984	0.993	0.983	0.984

Table 7: Training and validation accuracy for split 0.2/0.8

Model	MNIST_train	MNIST_val	Optical_train	Optical_val	Pen_train	Pen_val	Semeion_train	Semeion_val
DecisionTree	0.969	0.963	1.000	0.946	0.991	0.976	0.988	0.907
LogisticRegression	0.987	0.982	1.000	0.993	0.983	0.980	1.000	0.966
SVM	0.988	0.982	0.996	0.992	0.984	0.981	0.996	0.968

Table 8: Training and validation accuracy for split 0.5/0.5

Model	MNIST_train	MNIST_val	Optical_train	Optical_val	Pen_train	Pen_val	Semeion_train	Semeion_val
DecisionTree	0.967	0.962	0.997	0.963	0.987	0.978	0.970	0.925
LogisticRegression	0.986	0.984	0.999	0.991	0.983	0.980	1.000	0.979
SVM	0.988	0.984	0.997	0.991	0.983	0.982	0.992	0.978

Table 9: Training and validation accuracy for split 0.8/0.2

Model	MNIST_train	MNIST_val	Optical_train	Optical_val	Pen_train	Pen_val	Semeion_train	Semeion_val
DecisionTree	0.966	0.962	0.994	0.968	0.985	0.978	0.969	0.930
LogisticRegression	0.986	0.984	0.997	0.992	0.984	0.982	0.999	0.977
SVM	0.987	0.984	0.996	0.994	0.984	0.983	0.991	0.975