CSE 537 Assignment 5 Report: ML Classifiers

Remy Oukaour
SBU ID: 107122849
remy.oukaour@gmail.com

Jian Yang
SBU ID: 110168771
jian.yang.1@stonybrook.edu

Wednesday, December 9, 2015

1 Introduction

This report describes our submission for assignment 5 in the CSE 537 course on artificial intelligence. Assignment 5 requires us to implement two kinds of machine learning classifiers: a decision tree classifier for predicting credit worthiness of applicants, and a naïve Bayes classifier for classifying handwritten digits. In this report, we discuss the implementation details and performance of our solutions.

2 Decision tree classifier

Jian Yang developed the decision tree classifier for predicting credit worthiness of applicants.

3 Naïve Bayes classifier

Remy Oukaour developed the naïve Bayes classifier for classifying handwritten digits.

3.1 Instructions

To run the classifier, enter $pythonnaive{-}bayes.py$. It will read from $trainingimages.txt$, $traininglabels.txt$, $testimages.txt$, and $testlabels.txt$, and output to $predictedlabels.txt$ and $confusion - matrix.txt$.

3.2 Implementation

The classifier is a straightforward implementation of naïve Bayes, using the formula "log posterior $\propto$ log prior + log likelihood":

$$\log P(label|features) \propto \log P(label) + \sum_{feature} P(feature|label)$$

The prior probability $P(label)$ is estimated to be the fraction of training instances with a given label (0 to 9). The likelihood of a feature having a certain value for a test instance is estimated to be the fraction of training instances with that value for that feature. (We use Laplace smoothing to handle novel feature values in the test data, with a smoothing value of 0.001.) We then pick the label of each test instance using a maximum likelihood estimator:

$$classification(features) = \operatorname*{argmax}_{labels} \log P(label|features)$$

3.3 Feature selection

3.4 Parameter tuning

3.5 Performance results

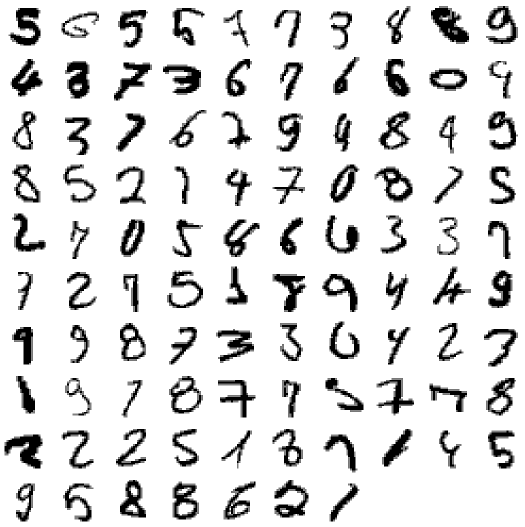

Figure 1: The 97 digits which the algorithm misclassified.