CS 7641: Supervised Learning Assignment

Ymakram3

# Introduction

In this assignment we are analyzing a set of supervised machine learning algorithms on two different datasets to gain an intuition on how different algorithms perform. We are using Python, sickit-learn, and matplotlib libraries to perform the experiments. The experiments are performed in a set of Jupyter notebooks, a notebook per algorithm. The algorithms are Decision Trees, Neural Networks, Boosting, Support Vector Machines, and k-Nearest Neighbors.

We are using two datasets to evaluate the different algorithms. The first dataset is a tic-tac-toe end game dataset, with a board state as features, and win or loose as prediction. The second dataset is MNIST digits dataset, with a set of handwritten digits pixels as features, and numbers as classes.

# Datasets

## Tic-tac-toe End Game Dataset

<https://www.openml.org/d/50>

The dataset has 958 instances representing all valid end game configuration of tic tac toe games assuming x played first. Each instance has 9 features representing one tic-tac-toe square. Each feature can be one of {x, o, b} values. The value x means player x has taken the square, value o means player o has taken the square, and value b means a blank square. The target label represents win or loose configuration with two possible classes {positive, negative}. The value positive represents a win for x, and value negative represent a loss for x. The class distribution is 65.3% positive and 34.7% negative. This is a slight imbalance that need to be addressed either with sampling, or some precision-based scoring.

This dataset was selected for the following reasons:

* Simplicity: The relatively small number of features and instances allows faster iterations on algorithms and experiment with API, and algorithm configuration, while not being trivial for meaningful results.
* Categorical: The features have string labels, which requires preprocessing for different algorithms to work.
* Complete: The dataset represents the full problem space of the game.
* Relatively small number of instances: It is interesting to compare how different algorithms perform on relatively small number of instances.

## MNIST Database of Handwritten Digits

<http://yann.lecun.com/exdb/mnist/>

The MNIST database has a training set of 60,000 examples of handwritten digits, and a test set of 10,000 examples. The database has been preprocessed and normalized to 28x28 images with various grey levels generated by anti-aliasing. The number of features is 784 corresponding to pixel values in 28x28 images. The target classes are {0, 1, 2, 3, 4, 5, 6, 7, 8, 9} representing single digit numbers.

|  |  |
| --- | --- |
| **Class** | **Number of Instances** |
| 0 | 6903 |
| 1 | 7877 |
| 2 | 6990 |
| 3 | 7141 |
| 4 | 6824 |
| 5 | 6313 |
| 6 | 6876 |
| 7 | 7293 |
| 8 | 6825 |
| 9 | 6958 |
| **Total** | 70000 |

The dataset was selected for the flowing reasons:

* High Dimensionality: The dataset has 784 features, which is interesting to evaluate how algorithms are performing on high dimensional set.
* Relatively large number of instances: Interesting to compare the performance of different algorithms compared to the small number of instances in the first dataset.
* Relatively large of number of classes: Interesting to compare the performance of classifying 10 different classes compared to the two classes in the first dataset.