m^2l^2: My Machine Learning Library

Hauke Neitzel

July 8, 2015

1 Introduction

1.1 About this library

"m^2l^2" is a small machine learning library I have written / am writing during my summer 2015 internship at the Cavendish Laboratory Lab for Scientific Computing (LSC). The purpose of the library is not to be much use for actual machine learning, instead it is mostly an educational tool for me (and maybe others). There exist mature machine learning libraries for many languages (e.g. 'scikit-learn' for python) and I have neither the expertise nor the time to implement the algorithms in a complete (all the different options and statistical considerations) and efficient (well-optimised) manner.

2 Structure

The library is divided into several modules corresponding to different general areas in machine learning. This section will give an overview of how the algorithms in the different modules are used, while section 3 contains documentation for each individual algorithm.

2.1 m2l2. classification

The algorithms in this module deal with the task of classification, i.e. deciding what class $C_k, k \in \{0 \dots K-1\}$ a set of features (feature vector $\mathbf{x} = \{x_0, x_1, \dots, x_n\}$) belongs to based on a set of training data $\{(\mathbf{x}_i, c_i)\}, c_i \in \{C_k\}$.

The algorithms are implemented as classes to retain the parameters and training data given. After a classifier is created (in some cases taking parameters for the algorithm), it can be trained on some training data. Once it is trained it can be used to classify further feature vectors.

```
class Classifier:
  def __init__(self, parameters):
    # 'parameters' can be different things used to
    # specify the exact operation of the classifier.
    # This includes the kernel for a SVM and the
    # number of classes for multi-class classifiers.

  def train(self, X, y):
    # X is an array of feature vectors and y is an
    # array of the associated classes. The classes
    # are expected to be numbered 0 to K-1

  def classify(self, x):
    # x is a feature vector to be classified.
    # For a binary classifier, a negative result means
    # class 0, a positive result class 1, with the
    # absolute value giving some measure of certainty.
    # For multi-class classifiers, the result is just
    # the number of the class.
```

3 Documentation