Adaboost

April 8, 2020

1 Adaboost

[1]:
```
import numpy as np
from matplotlib import pyplot as plt
from sklearn.datasets import make_classification
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import train_test_split
```

1.1 Define Adaboost class and methods

[2]:
```
class Adaboost:
    def __init__(self):
        return

    def adaboost(self):
        for t in range(self.T):
            wl = DecisionTreeClassifier(max_depth=1)
            wl.fit(self.x, self.y, sample_weight=self.D[t])
            self.h.append(wl)

            y_predict = wl.predict(self.x)
            error_ids = (y_predict != self.y)
            weighted_error = np.sum(self.D[t][error_ids])

            self.alpha[t] = 1 / 2 * np.log((1 - weighted_error) /␣
 ↪weighted_error)

            self.D[t+1] = self.D[t] * np.exp(-self.alpha[t] * self.y * self.
 ↪h[t].predict(self.x))
            Zt = np.sum(self.D[t+1])
            self.D[t+1] /= Zt

        return

    def fit(self, x, y, T):
        self.x = x
```

```
        self.y = y
        self.T = T
        self.N = self.y.size

        self.D = np.zeros((self.T+1, self.N))
        self.D[0] = 1 / self.N
        self.alpha = np.zeros(self.T)

        self.h = []

        self.adaboost()

        return

    def predict(self, x):
        H = np.zeros(x.shape[0])
        for t in range(self.T):
            H += self.alpha[t] * self.h[t].predict(x)

        return np.sign(H)

    def score(self, x, y):
        ypredict = self.predict(x)
        return np.sum(ypredict == y) / y.size
```

1.2 Load datasets, train and evaluate

[3]:
```
datasets = ["bank", "breast-cancer", "congressional-voting", "hepatitis",␣
↪"ionosphere", "magic",
           "ozone", "parkinsons", "ringnorm", "spambase"]
```

[4]:
```
for i in range(len(datasets)):
    path = "../UA-ECE523-EngrAppMLData/data/" + datasets[i] + ".csv"
    data = np.loadtxt(path, delimiter=",")
    X, y = data[:, :-1], data[:, -1]
    y[y == 0] = -1

    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.20)

    clf = Adaboost()
    clf.fit(X, y, 10)
    print("Training accuracy: {:.2f}. Test accuracy: {:.2f}".format(clf.
↪score(X_train, y_train), clf.score(X_test, y_test)))
```

```
Training accuracy: 0.90. Test accuracy: 0.87
Training accuracy: 0.71. Test accuracy: 0.83
```

```
Training accuracy: 0.61. Test accuracy: 0.67
Training accuracy: 0.85. Test accuracy: 0.90
Training accuracy: 0.93. Test accuracy: 0.97
Training accuracy: 0.80. Test accuracy: 0.81
Training accuracy: 0.97. Test accuracy: 0.97
Training accuracy: 0.92. Test accuracy: 0.97
Training accuracy: 0.81. Test accuracy: 0.82
Training accuracy: 0.91. Test accuracy: 0.91
```

[ ]: