Online Machine Learning Algorithms

1 Binary-classifiers

1.1 Algorithm

Generic implementation of online binary-classifier algorithm is shown below. Input and output arguments of the following implementation can be augmented, if necessary.

Algorithm 1: Online Binary-Classifier Algorithm

Input : $\mathcal{D}$: Training example data set, $\mathcal{N}$: Maximum number of training iterations
Output: w : final weight vector

initialize weight vector, $w = 0$
for $i \leftarrow 1$ **to** $\mathcal{N}$ **do**
  for $j \leftarrow 1$ **to** *each* $(x_t, y_t) \in \mathcal{D}$ **do**
    $\hat{y}_t = sign(w \cdot x_t)$
    if $\hat{y}_t \neq y_t$ **then**
      $w = w + \tau \cdot x_t \cdot y_t$
    end
  end
end
return final weight vector, w

In standard Perceptron algorithm, learning rate, τ, is assumed as constant (say, $\tau = 1$) and for Passive-Aggressive (PA) algorithm, learning rate, τ, needs to be computed using following equation for each mistake in prediction:

$$\frac{1 - y_t \cdot (w \cdot x_t)}{||x_t||^2}$$

For Averaged Perceptron, above algorithm can be modified by keeping track of a mistake counter variable, C, and cumulative weight vector, w_{sum}, and returning the final weight vector as w_{sum}/C.

1.2 binaryclasstest.m

Learn a binary classifier to classify even labels (0, 2, 4, 6, 8) and odd labels (1, 3, 5, 7, 9).

a. Compute the online learning curve for both Perceptron and PA algorithm by plotting the number of training iterations (1 to 50) on the x-axis and the number of mistakes on the y-axis.

b. Compute the accuracy of Perceptron, PA, and Averaged Perceptron algorithm on testing data for 20 training iterations by plotting the number of training iterations (1 to 20) on the x-axis and accuracy on the y-axis.

2 Multiple-classifiers

2.1 Algorithm

Generic implementation of online multi-classifier algorithm is shown below. Input and output arguments of the following implementation can be augmented, if necessary.

Algorithm 2: Online Multiple-Classifier Algorithm

Input : $\mathcal{D}$: Training example data set, $\mathcal{N}$: Maximum number of training iterations
Output: w : final weight vector

```
initialize weight vector, w = 0
for i ← 1 to N do
    for j ← 1 to each (x_t, y_t) ∈ D do
        ŷ_t = arg max_{y∈1,2,...k}(w · F(x_t, y_t))
        if ŷ_t ≠ y_t then
            w = w + τ · (F(x_t, y_t) − F(x_t, ŷ_t))
        end
    end
end
return final weight vector, w
```

Each training example is of the form (x_t, y_t), where $x_t \in \Re^d$ is the input and $y_t \in 1, 2, ..., k$ is the class (output) label. In this representation, you will have a single weight-vector $w \in \Re^{k \times d}$ and the augmented feature function $F(x_t, y) \in \Re^{k \times d}$ will have k blocks of size d and it will have zeroes everywhere except for the y^{th} block, which will have x_t in it.

Similar to binary-classifier implementation, learning rate, τ, will be considered constant in standard multi-classifier Perceptron and for standard multi-classifier Passive Aggressive algorithm, learning rate, τ, needs to be computed using following formula for each mistake in prediction:

$$\frac{1 - (w \cdot F(x_t, y_t) - w \cdot F(x_t, \hat{y}_t))}{||(F(x_t, y_t) - F(x_t, \hat{y}_t))||^2}$$

For multi-classifier Averaged Perceptron, above algorithm can be modified to return the final weight vector as w_{sum}/C, similar to binary-classifier Averaged Perceptron.

2.2 multiclasstest.m

Learn a multi-class classifier to map images to the corresponding fashion label. Repeat the same experiments using multi-class classifier algorithms as described in section 1.2 (**binaryclasstest.m**).