Computer Science M146, Homework 4

Michael Wu
UID: 404751542

March 8th, 2018

Problem 1

We have the following data

i	x	y	Label
1	0	8	$-$
2	1	4	$-$
3	3	7	$+$
4	-2	1	$-$
5	-1	13	$-$
6	9	11	$-$
7	12	7	$+$
8	-7	-1	$-$
9	-3	12	$+$
10	5	9	$+$

and after running our AdaBoost algorithm we get the following table.

		Hypothesis 1 (1st iteration)				Hypothesis 2 (2nd iteration)			
i	Label	D_0	$f_1 \equiv [x > 2]$	$f_2 \equiv [y > 6]$	$h_1 \equiv [x > 2]$	D_1	$f_1 \equiv [x > 11]$	$f_2 \equiv [y > 11]$	$h_2 \equiv [y > 11]$
(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)
1	$-$	$\frac{1}{10}$	$-$	$+$	$-$	$\frac{1}{16}$	$-$	$-$	$-$
2	$-$	$\frac{1}{10}$	$-$	$-$	$-$	$\frac{1}{16}$	$-$	$-$	$-$
3	$+$	$\frac{1}{10}$	$+$	$+$	$+$	$\frac{1}{16}$	$-$	$-$	$-$
4	$-$	$\frac{1}{10}$	$-$	$-$	$-$	$\frac{1}{16}$	$-$	$-$	$-$
5	$-$	$\frac{1}{10}$	$-$	$+$	$-$	$\frac{1}{16}$	$-$	$+$	$+$
6	$-$	$\frac{1}{10}$	$+$	$+$	$+$	$\frac{1}{4}$	$-$	$-$	$-$
7	$+$	$\frac{1}{10}$	$+$	$+$	$+$	$\frac{1}{16}$	$+$	$-$	$-$
8	$-$	$\frac{1}{10}$	$-$	$-$	$-$	$\frac{1}{16}$	$-$	$-$	$-$
9	$+$	$\frac{1}{10}$	$-$	$+$	$-$	$\frac{1}{4}$	$-$	$+$	$+$
10	$+$	$\frac{1}{10}$	$+$	$+$	$+$	$\frac{1}{16}$	$-$	$-$	$-$

For the first iteration we get the error and vote

$$\epsilon_1 = \frac{2}{10}$$

$$\alpha_1 = \frac{1}{2}\log_2\left(\frac{1-\frac{2}{10}}{\frac{2}{10}}\right) = 1$$

which allows us to generate the next weights

$$D_1(i) = \frac{D_0(i)}{Z_1}2^{-\alpha_1 y_i h_1(x_i)} = 2^{-3-y_i h_1(x_i)} = \begin{cases} \frac{1}{4} & \text{if correct} \\ \frac{1}{16} & \text{if incorrect} \end{cases}$$

For our second iteration we get the error and vote

$$\epsilon_2 = \frac{1}{4}$$

$$\alpha_2 = \frac{1}{2}\log_2\left(\frac{1-\frac{1}{4}}{\frac{1}{4}}\right) = \frac{\log_2(3)}{2}$$

and combining these two gives us our final hypothesis

$$H_{\text{final}}(x,y) = \operatorname{sgn}\left(h_1(x,y) + \frac{\log_2(3)}{2}h_2(x,y)\right) = \begin{cases} 1 & \text{if } x > 2 \\ -1 & \text{if } x \leq 2 \end{cases}$$

which is the same as hypothesis h_1 because $\alpha_1 > \alpha_2$, so h_1 effectively overpowers h_2.

Problem 2

a)

1. One vs. All learns k classifiers, All vs. All learns $\binom{k}{2}$ classifiers.

2. One vs. All uses m examples for each classifier, All vs. All uses $2\frac{m}{k}$ examples for each classifier.

3. One vs. All will decide using the label for the classifier which most strongly classifies an example as positive. For a linear classifier like perceptron this means the classifier with the maximum value of $\mathbf{w}^T\mathbf{x}$, where $\mathbf{w}^T$ is the learned weight vector and $\mathbf{x}$ is the example.

   All vs. All will decide using the label that gets the most votes out of all the classifiers. This is a majority vote style of evaluation. A tournament style of evaluation can also be used, where each label is tested only once and if it loses against the current label, it is not considered again. The winning label moves on to the next round of the tournament, and the last label remaining is assigned to the example.

4. Assuming that the learning complexity of L is constant, the complexity of One vs. All is $\Theta(k)$, because it needs to train k classifiers. All vs. All is $\Theta(k^2)$, because $\binom{k}{2}$ scales with k^2. However, due to the smaller number of examples in each All vs. All classifier this may not be true in reality, as the learning complexity may vary with the number of examples for each classifier. Assuming L is the perceptron algorithm, it has the complexity at least $\Omega(n)$ when learning on a set of n examples, as it needs to check every example once to determine convergence. Then One vs. All has $\Omega(km)$ complexity since it uses m examples and All vs. All has $\Omega(km)$ complexity as well since it uses $2\frac{m}{k}$ examples.

b) I would prefer One vs. All, as it is simpler than All vs. All. It is the most natural reduction of multi-class classification to binary classification. Although it cannot classify certain data sets where the data is not easily separable, it works well in practice. Additionally it requires learning fewer classifiers, so it is more intuitive to understand. Its best case performance is about the same as All vs. All, so we do not incur a large penalty for this choice.

c) Kernel perceptron needs to compute the kernel between every pair of points which is $\Theta(n^2)$ complexity on n examples. Then One vs. All has $\Omega(km^2)$ complexity and All vs. All has $\Omega(m^2)$ complexity. So All vs. All benefits from the reduced amount of examples per classifier, leading it to have a faster training time. Thus I would choose All vs. All if I was using kernel perceptron.

d) One vs. All is $O(kdm^2)$, and All vs. All is $O(dm^2)$. This is because the number of examples per classifier scales inversely with k in All vs. All. Thus the All vs. All is more efficient.

e) One vs. All is $O(kd^2m)$, and All vs. All is $O(kd^2m)$. This is because the number of examples per classifier scales inversely with k in All vs. All. Thus they are equally efficient.

f) For counting, we need to calculate $w_i^T x$ for every $\binom{m}{2}$ classifier. This takes $O(dm^2)$ time. For knockout, we need to calculate $w_i^T x$ for m classifiers. This takes $O(dm)$ time. Thus knockout is the faster evaluation method.