Optional Assignment - Cost-Boosting

Shaun Pritchard

Florida Atlantic University

CAP 6778

September -13-2021

M. Khoshgoftaar

The cost boosting algorithm uses time-varying cost function weight update to count the appropriate weight for misclassification between type I and Type II errors to account for variation in the meta learner or weak learner stumps in the decision tree. Which unlike adaboost takes the misclassifications into account during the weight update. Through iteration through each algorithm's weak learner cost matrix, information about each algorithm, its error, and its cost function are voted on and weighted to produce a more accurate result.

We start by assigning the boost class in Y = yi = (-1 || 1) as the instance which is assigned to the feature vectors through error calculations and initial weights 1/N(0.1) = $\boldsymbol{D_1(i)}$. There are 15 data points in this set which will mean $\boldsymbol{D_1(i)}$ *the sample weight* will equal 1/15 = 0.067. The predicted output = $\boldsymbol{h_1(i)}$ representing the original trained data on the first learner is ether $\boldsymbol{(yi = h_1(i)}$) = 1, or $\boldsymbol{(yi \neq h_1(i))}$ = -1. *(note if the sample weight exceeds 0.5 then the iterative process yields).*

Proceeding for each new tree **[$D_1(i)$ + $D_2(i)$ … $D_t(i)$]** we will determine the distribution for weight instances through the weak hypothesis based on the chosen parameter as a weight for the weak hypothesis.

$\alpha = \frac{1}{2}\ln\left(\frac{1-\frac{1}{15}}{1/15}\right)$ = 1.3195, The new sample weight is the original sample weight * $e^{\frac{\alpha}{}}$ = 0.067 $(1/15)\frac{1.3195}{}$ for each ***t*** iteration value {-1 || 1} for **[$D_1(i)$ + $D_2(i)$ … $D_t(i)$]**. α = the amount of say for the stumps inference vote.

For **[$D_2(i)$, $D_3(i)$ … D_t (i)]** sample weights will be normalized by dividing each weight by the new weighted instances to equal 1 and either increased or decreased based on the cost adjustment factor for each iteration. Furthermore, subtracting the difference between the misclassifications of the original sample weights to derive new sample weights for **[$D_2(i)$, $D_3(i)$ ... $D_t(i)$].** Observe, the 14 row the verified misclassification -1 weight of 2 as the cost. The iteration for the continuing weights increases or decreases according to cost, boosting weak learners by dividing error by the data samples then subtracting the error for each new sample. 0.067/15 -0.067 = 0.063 for $D_2(i)$, $D_3(i)$ = 0.063 /15 - 0.063 = 0.058 and for all misclassifications of weight instances 1 and greater we see the number increase for false positive values which means the vote will earn less.

Cost Boosting

i	*yi*	$D_1(i)$	$h_1(i)$	$D'_2(i)$	$D_2(i)$	$h_2(i)$	$D'_3(i)$	$D_3(i)$
1	1	0.067	**1**	**1**	0.063	**1**	0.938	0.058
2	1	0.067	**-1**	**1**	0.063	**1**	0.938	0.058
3	1	0.067	**1**	**1**	0.063	**1**	0.938	0.058
4	1	0.067	**-1**	**1**	0.063	**-1**	1.0	0.062
5	1	0.067	**-1**	**1**	0.063	**1**	0.938	0.058
6	1	0.067	**1**	**1**	0.063	**-1**	1.0	0.062
7	1	0.067	**1**	**1**	0.063	**1**	0.938	0.058
8	1	0.067	**-1**	**1**	0.063	**-1**	1.0	0.062
9	1	0.067	**1**	**1**	0.063	**1**	0.938	0.058
10	1	0.067	**1**	**1**	0.063	**1**	0.938	0.058
11	-1	0.067	**-1**	**1**	0.063	**-1**	0.938	0.058
12	-1	0.067	**-1**	**1**	0.063	**-1**	0.938	0.058
13	-1	0.067	**-1**	**1**	0.063	**1**	2.000	0.123
14	-1	0.067	**1**	**2**	0.063	**-1**	1.875	0.115

15	-1	0.067	**-1**	**1**	0.063	**-1**	0.938	0.058