LazyBoosting
How does Adaboost perform with sub-optimal base classifiers?

Chakshu Sardana and Vighnesh Birodkar

NYU Courant Institute

December 20, 2016

Introduction - Weak learners

As defined in [Mohri et al., 2012] concept class C is said to be weakly PAC-learnable if there exists an algorithm A, $\gamma > 0$, and a polynomial function $poly(\cdot, \cdot, \cdot, \cdot)$ such that for any $\epsilon > 0$ and $\delta > 0$, for all distributions D on X and for any target concept $c \in C$, the following holds for any sample size $m \geq poly(1/\epsilon, 1/\delta, n, size(c))$.

$$\Pr_{S \sim D^m}\left[R(h_s) \leq \frac{1}{2} - \gamma\right] \geq 1 - \delta \tag{1}$$

where $R(h_s)$ denotes the generalization error of h_s. The hypothesis returned by the weak learner is referred to as the base classifier.

Introduction - Adaboost

Input: $(x_1, y_1), (x_2, y_2), \dots (x_n, y_n)$ where $x_i \in \mathbb{R}$ and $y_i \in \{-1, +1\}$.

Initialize $D_1(i) = 1/m$ for all $1 \leq i \leq m$

for $t = 1$ *to* T **do**

  Train base classifier with small error $\epsilon_t = \Pr_{i \sim D_t}[h_t(x_i) \neq y_i]$

  $\alpha_t \leftarrow \frac{1}{2} \log\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$

  $Z_t \leftarrow 2[\epsilon_t(1-\epsilon_t)]^{1/2}$

  for $i = 1$ *to* m **do**

    $D_{t+1}(i) \leftarrow \frac{D_t(i) \exp(-\alpha_t y_i h_t(x_i))}{Z_t}$

  end

end

$g = \sum_{t=1}^{T} \alpha_t h_t$

return $h = \mathrm{sgn}(g)$

Analysis

- Do we need base classifier with smallest possible error in Step 3 ?

Analysis of Adaboost

Let g be the function computed by adaboost in step 10 of Adaboost. Let H be the family of base classifiers and $\hat{R}_\rho(\cdot)$ be the empirical margin loss with margin ρ for a function. Then [Mohri et al., 2012] show that the following holds

$$R(g) \leq \hat{R}_\rho(g/\|\alpha\|_1) + \frac{2}{\rho}\mathfrak{R}_m(H) + \sqrt{\frac{\log\frac{1}{\delta}}{2m}} \tag{2}$$

with probability at least $1-\delta$, where $\mathfrak{R}_m$ is the expectation of empirical Radamacher complexity over all samples of size m. [Mohri et al., 2012] also show that

$$\hat{R}(g/\|\alpha\|_1) \leq \left[(1+2\gamma)^{1+\rho}(1-2\gamma)^{1-\rho}\right]^{T/2} = f(\gamma)^{T/2} \tag{3}$$

Possible benefit of sub-optimal base classifiers

Taking the derivative with respect to γ of Equation 3 we get

$$
\begin{aligned}
\frac{df}{d\gamma} &= (1+2\gamma)^{\rho}(1+\rho)(2)(1-2\gamma)^{1-\rho} + (1-2\gamma)^{-\rho}(1-\rho)(-2)(1+2\gamma) \\
&= \Big[(1+2\gamma)^{\rho}(1-2\gamma)^{-\rho}\Big]\Big[2(1+\rho)(1-2\gamma) + (-2)(1-\rho)(1+2\gamma)\Big] \\
&= \Big[(1+2\gamma)^{\rho}(1-2\gamma)^{-\rho}\Big]2\Big[1-2\gamma+\rho-2\rho\gamma-1-2\gamma+\rho+2\rho\gamma\Big] \\
&= 4\Big[(1+2\gamma)^{\rho}(1-2\gamma)^{-\rho}\Big](\rho-2\gamma) \qquad (4)
\end{aligned}
$$

Since $\gamma < \frac{1}{2}$ Equation 4 is positive for $\gamma < \rho/2$. This indicates that f is increasing for $\gamma < \rho/2$ and decreasing for $\gamma > \rho/2$.

Possible benefit of sub-optimal base classifiers

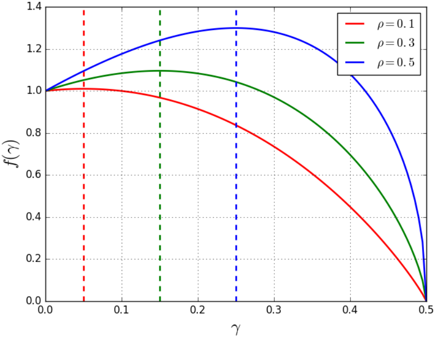


Figure 1: Plot of function f which is an upper bound on $\hat{R}(g/\|\alpha\|_1)$. The dotted line represents the maximum of each curve. To the left of the maximum, returning a sub-optimal base classifier will decrease the bound

Attempt to return sub-optimal base classifiers

- A deterministic polynomial time algorithm for finding a weak linear classifier.
- Sub-optimal boosting stumps.

Weak-linear classifier

- The problem of finding a linear classifier with minimal classification error in NP-hard in terms of the dimensionality of the feature vector [Johnson and Preparata, 1978].
- However, for any finite distribution of data, there exists a linear classifier with $> 1/2$ accuracy and it can be found in polynomial time as shown by [Mannor and Meir, 2001]. For any input distribution D on m points, the algorithm guarantees to find a and b with $\hat{y}_i = \mathrm{sgn}(a \cdot x + b)$ such that $\sum_{i=1}^{m} D_i 1_{\hat{y}_i \neq y_i} \leq \frac{1}{2} - \frac{1}{4m}$.

Experiments - Weak linear classifier

Extremely slow convergence

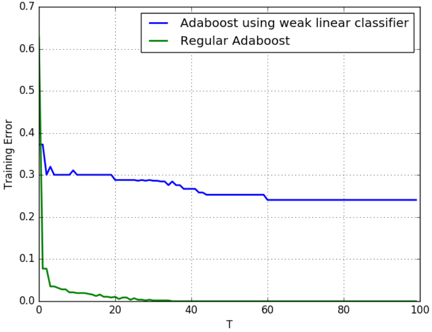


Figure 2: Convergence of Adaboost using weak linear classifier as compared to using boosting stumps on the `iris` dataset.

Experiments - Weak linear classifier

- The poor convergence of the method can be explained by the bound on the training error of adaboost as shown by [Mohri et al., 2012]

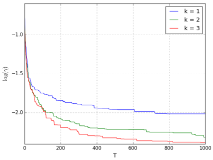
$$\hat{R}(h) \leq \exp(-2\gamma^2 T) \tag{5}$$

- Since for the weak linear classifier the minimum value γ is $1/4m$, for large m, $\gamma \to 0$ and $\gamma^2 \to 0$ even faster. Since $\lim_{\gamma \to 0} \big(\exp(-2\gamma^2 T)\big) \approx 1 - 2\gamma^2 T$ the upper bound is approximately linear in T.

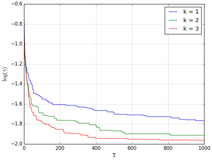
Sub-optimal boosting stumps

- Instead of of boosting stumps which select the best feature and best threshold to split each feature, we instead the 2nd, 3rd, or in general the k^{th} best feature and then perform an optimal split.
- This results in a γ lesser than or equal to what would have been possible with an optimal split at each step.

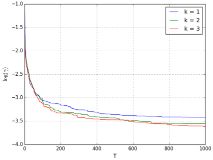
Experiments - Sub-optimal boosting stumps



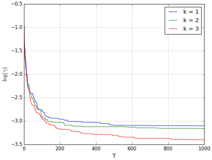
(a) `ocr` dataset.



(b) `digits` dataset.



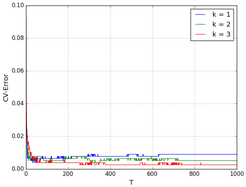
(c) `ionosphere` dataset.



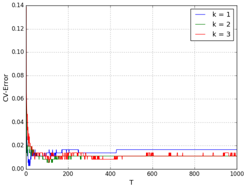
(d) `breastcancer` dataset.

Figure 3: Comparison of observed γ for the datasets used when varying k for varying time steps of Adaboost T.

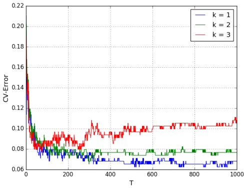
Experiments - Sub-optimal boosting stumps



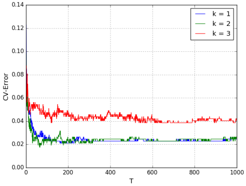
(a) `ocr` dataset.



(b) `digits` dataset.



(c) `ionosphere` dataset.



(d) `breastcancer` dataset.

Figure 4: Variation of cross validation error with number of time steps T

Experiments - Sub-optimal boosting stumps

Dataset	**Min. Error**		
	k=1	**k=2**	**k=3**
`breastcancer`	0.021084(T=162)	**0.017575(T=65)**	0.038628(T=520)
`ionosphere`	0.062540(T=516)	0.065317(T=300)	0.076825(T=75)
`digits`	0.002703(T=13)	0.005556(T=101)	0.005556(T=110)
`ocr`	0.003863(T=33)	0.003879(T=106)	**0.002581(T=255)**

Table 1: Minimum cross validation error for each suboptimal stump along with the time step T which gave the minimum error.

Conclusion

- Choosing the base classifier with least error might not always give the best final classifier Adaboost
- k Can be a hyper-parameter left to the user, possibly found by cross validation.
- γ might be chosen in a data-dependent way to possibly provide more benefits.

References I

Johnson, D. and Preparata, F. (1978).
The densest hemisphere problem.
Theoretical Computer Science, 6(1):93 – 107.

Mannor, S. and Meir, R. (2001).
Weak learners and improved rates of convergence in boosting.
In Leen, T. K., Dietterich, T. G., and Tresp, V., editors, *Advances in Neural Information Processing Systems 13*, pages 280–286. MIT Press.

Mohri, M., Rostamizadeh, A., and Talwalkar, A. (2012).
Foundations of Machine Learning.
The MIT Press.
Chapter 6, Boosting.