Some New Steps for AdaBoost Algorithms

Shuisheng Zhou
Computer Science,
UCSC
Outline

- Framework of AdaBoost algorithm
- Step of AdaBoost
- Step of AdaBoost $\rho^*$
- New step for AdaBoost
- New step for AdaBoost $\rho^*$
- Simple Approx. step for AdaBoost $\rho^*$
- Experimental results
Framework of AdaBoost algorithm

Input: $S = \{ x_n, y_n \}_{n=1}^{N}$.

Output: The hypothesis $h_t(x)$ and their coefficients $\alpha_t$, $t = 1, 2, \ldots, T$. Then the decision function is $I(x) = \text{sign}\left( \sum_{t=1}^{T} \alpha_t h_t(x) \right)$.

Initialization: Given the No. of iterations T. Set $d_n^0 = 1/N$ for all $n = 1, 2, \ldots, N$.

Do for $t = 1, 2, \ldots, T$:

1. **Step 1**: Train classifier on $\{S, d^t\}$ and obtain hypothesis $h_t: x \rightarrow [-1,1]$;

2. **Step 2**: Calculate the edge of $h_t(x)$: $\gamma_t = \sum_{n=1}^{N} d_n^t y_n h_t(x_n)$;

3. **Step 3**: If $|\gamma_t| = 1$, then the algorithm return with $\alpha_1 = \text{sign}(\gamma_t)$, $h_1 = h_t$, $T = 1$.

4. **Step 4**: Find a Step $\alpha_t$ to update $d^t$;

5. **Step 5**: Update $d_n^{t+1} = d_n^t e^{-\alpha_t y_n h_t(x_n)}/Z_t$ for $i = 1, \ldots, N$ with $Z_t = \sum_{n=1}^{N} d_n^t e^{-\alpha_t y_n h_t(x_n)}$. 

Step of AdaBoost

Step: \[ \alpha_t = \frac{1}{2} \ln \frac{1 + \gamma_t}{1 - \gamma_t} \]

Motivation: Minimize \[ Z_t(\alpha) = \sum_{n=1}^{N} d_{n}^{t-1} e^{-\alpha u'_n} \]

or keep \[ d^t(\alpha) \cdot u^t = 0 \]

Approximation by \[ e^{-\alpha x} \leq \frac{1}{2} (1 + x) e^{-\alpha} + \frac{1}{2} (1 - x) e^\alpha \]

\[ Z_t(\alpha) \leq \left( \frac{1}{2} \left( 1 + d^{t-1} \cdot u^t \right) e^{-\alpha} + \frac{1}{2} \left( 1 - d^{t-1} \cdot u^t \right) e^\alpha \right) \]

Instead to minimize RHS get step above.
Step of AdaBoost* 

**Step:** \( \alpha_t = \frac{1}{2} \ln \frac{1+\gamma_t}{1-\gamma_t} - \frac{1}{2} \ln \frac{1+\rho_t}{1-\rho_t} \)

**Motivation:** Minimize \( Z_t(\alpha) = \sum_{n=1}^{N} d_{n}^{t-1} e^{-\alpha(u_{n}^{t} - \rho)} \)

or keep \( d'(\alpha) \cdot u^{t} = \rho \)

**Approximation by** \( e^{-\alpha x} \leq \frac{1}{2} (1+x)e^{-\alpha} + \frac{1}{2} (1-x)e^{\alpha} \)

\( Z_t(\alpha) \leq \left( \frac{1}{2} \left(1+d'^{t-1} \cdot u^{t}\right)e^{-\alpha} + \frac{1}{2} \left(1-d'^{t-1} \cdot u^{t}\right)e^{\alpha} \right)e^{\alpha \rho} \)

**Instead to minimize RHS get step above.**
New step for AdaBoost

Step: \[ \alpha_t = \frac{1}{2} \ln \frac{\gamma_t^+}{-\gamma_t^-} \]

Motivation: Minimize \( Z_t(\alpha) \) or keep \( d^t(\alpha) \cdot u^t = 0 \)

Approximation by inequality

\[ e^{-\alpha x} \leq \begin{cases} 1 + x(1-e^\alpha), & x \in [-1, 0]. \\ 1 - x(1-e^{-\alpha}), & x \in (0, 1] \end{cases} \]

\[ Z_t(\alpha) \leq \left( 1 + \gamma_t^- (1-e^\alpha) - \gamma_t^+ (1-e^{-\alpha}) \right) \]

Instead to minimize RHS get step above.
The different inequality used

![Graph showing different inequalities.

Fig 1 The comparison of inequality (2) and (7) with $\alpha = 0.5$.](image)
New step for AdaBoost\( \rho \)

**Step:**
\[
\bar{\alpha}_t = \ln \frac{-b + \sqrt{b^2 - 4ac}}{2a}
\]
where \( a = -\gamma_t^- (\rho + 1), b = \rho (1 + \gamma_t^- - \gamma_t^+), c = -(1 - \rho) \gamma_t^+ \).

**Motivation:** Minimize
\[
Z_t(\alpha) = \sum_{n=1}^{N} d_n^{t-1} e^{-\alpha(u_n^{t}-\rho)}
\]
or keep
\[
d^t(\alpha) \cdot u^t = \rho
\]

Approximated as
\[
Z_t(\alpha) \leq \left(1 + \gamma_t^- (1 - e^\alpha) - \gamma_t^+ (1 - e^{-\alpha})\right) e^{\rho \alpha}
\]
Instead to minimize RHS get step above.
Simple approx. step for AdaBoost $\rho^*$

**Step:**

$$\hat{\alpha}_t = \frac{1}{2} \ln \frac{\gamma_t^+}{-\gamma_t^-} - \frac{1}{2} \ln \frac{1+\rho}{1-\rho}$$

**Motivation:** Let $b=0$.

We have:

$$0 < \alpha_t \leq \hat{\alpha}_t$$

and:

$$\bar{Z}_t(\alpha_t) \leq \bar{Z}_t(\hat{\alpha}_t)$$
Experimental results (1)

--- Hypothesis with the range \{-1,1\}

All the steps are same.

Email Spam data →
Experimental results(2)

----Hypothesis with the range [-1,1]

data with 100 dimensional and containing 98 nuisance dimensions with uniform noise.

20000 data results averaged over 200 splits $m$ training 20000-$m$ testing
Averaged margin achieved for 200 split problems with different training size
Margin obtained for different split problems with AdaBoost

- Original AdaBoost (Eq. (3))
- New step (Eq. (8))
- Optimal step obtained by line-search

Problems split from the same data
Margin obtained for different split problems with AdaBoost $\rho^*$
Conclusion

- For small size problem, the original step of AdaBoost works better than others;
- For a little larger size problems, our new steps work well the original ones;
Future works

☐ Apply the new steps to nonlinear problems;
☐ Prove the bounds of the corresponding algorithms;
☐ Do more experiments with practical dataset.
Reference


