Active Learning with Boosting for Spam Detection

Enela Pema
University of California, Santa Cruz
epema@soe.ucsc.edu
Active Learning with Boosting for Spam Detection

- Given a small set of labeled examples and a set of unlabeled examples
- Pick the most significant examples from the unlabeled ones
- Ask the user to label them for you
Active Learning with Boosting for Spam Detection

GOAL: Find a good hypothesis with a small set of labeled examples
Active Learning with Boosting for Spam Detection

Why Boosting?
Good at distinguishing hard examples
Active Learning with Boosting for Spam Detection

In my project:

Active Learning with AdaBoost
Active Learning with AdaBoost*
Active Learning with LPBoost
Active Learning with ERLPBoost
Active Learning Framework

Active Learning Framework with Confidence-Based Sampling (using Boosting)

Given a dataset of labeled examples \((x_1, y_1), \ldots, (x_{Ntr}, y_{Ntr})\) and unlabeled examples \((x_{Ntr+1}), \ldots, (x_N)\)

Train on \((x_1, y_1), \ldots, (x_{Ntr}, y_{Ntr})\) with boosting and provide a strong classifier \(H\)

Compute scores for \((x_{Ntr+1}), \ldots, (x_N)\) using the formula:

\[
score(x_i) = \frac{|\sum_{j=1}^{T} \alpha_j h_j(x_i)|}{\sum_{j=1}^{T} \alpha_j}
\]

Ask the user to label the \(k\) examples with smaller score.

Add the new labeled examples to the training set.

\(Ntr = Ntr + k\)

If \(N_{tr}\) smaller than the maximum size of the training set (given as input), repeat from the beginning.

Table 1: Active Learning Framework with Confidence-Based Sampling
Comparator: Active Learning with Random Sampling

<table>
<thead>
<tr>
<th>Table 2: Active Learning Framework with Random Sampling</th>
</tr>
</thead>
<tbody>
<tr>
<td>Given a dataset of labeled examples ((x_1, y_1), \ldots, (x_{N_{tr}}, y_{N_{tr}})) and unlabeled examples ((x_{N_{tr}+1}), \ldots, (x_N))</td>
</tr>
<tr>
<td>Train on ((x_1, y_1), \ldots, (x_{N_{tr}}, y_{N_{tr}})) with boosting and provide a classifier (H)</td>
</tr>
<tr>
<td>Pick randomly (k) unlabeled examples. Ask the user to label the (k) chosen examples.</td>
</tr>
<tr>
<td>Add the new labeled examples to the training set.</td>
</tr>
<tr>
<td>(N_{tr} = N_{tr} + k)</td>
</tr>
<tr>
<td>If (N_{tr}) smaller than the maximum size of the training set (given as input) then repeat from the beginning.</td>
</tr>
</tbody>
</table>
AdaBoost algorithm

Given $(x_1, y_1), ..., (x_N, y_N) \in S_t$ where $y_i \in \{-1, +1\}$
Initialize distribution over the examples $D_1 = \frac{1}{N}$ for $i = 1..N$

For $t=1$ to $T$ (number of boosting rounds)
Get weak hypothesis $h_t$ such that $h_t = \arg \min \{ \epsilon_j \}$
where $\epsilon_j$ is the error of weak learner $h_j$ given distribution $D_t$, $\epsilon_j = \sum_{i=1}^{N} D_t^i [h_j(x_i) \neq y_i]$.

Update distribution $D_{t+1}(x_i) = \frac{D_t(x_i) e^{-\alpha_t y_t h_t (x_i)}}{Z}$ where $Z$ is a normalizing factor.
and $\alpha_t = \frac{1}{2} \log \frac{1-\epsilon_t}{\epsilon_t}$

Output strong classifier $H(x_i) = \text{sign} \left( \sum_{k=1}^{t} \alpha_k y_k h_k(x_i) \right)$

Table 3: Table 1: AdaBoost Algorithm
AdaBoost
Active Learning with AdaBoost
Active Learning with AdaBoost

Active Learning with AdaBoost: Confidence-Based vs Random Sampling

Accuracy

Size of Training Set
AdaBoost*

**AdaBoost* Algorithm**

Given \((x_1, y_1), \ldots, (x_N, y_N) \in S_t\) where \(y_i \in \{-1, +1\}\)
Initialize distribution over the examples \(D_1 = \frac{1}{N}\) for \(i = 1..N\)

For \(t = 1\) to \(T\) (number of boosting rounds)
Get weak hypothesis \(h_t\) such that \(h_t = \arg \max \gamma_j\)
where \(\gamma_j\) is the edge of weak learner \(h_j\) given distribution \(D_t\).

\[
\gamma_j = \frac{1}{N} \sum_{i=1}^{N} D_t^i y_i h_t(x_i).
\]

If \(|\gamma_t| = 1\), then \(\alpha_t = \text{sign} (\gamma_t), h_1 = h_t, T = 1\), break
\(\gamma_t^{\text{min}} = \min \{ \gamma_r \}\) for \(r = 1..t\), \(\rho_t = \gamma_t^{\text{min}} - \nu\)
\(\alpha_t = \frac{1}{2} \ln \frac{1 + \gamma_t}{1 - \gamma_t}\)
Update distribution \(D_{t+1}(x_i) = \frac{D_t e^{-\alpha_t y_t h_t(x_t)}}{Z_t}\) where \(Z_t\) is a normalizing factor

Output strong classifier \(H(x_i) = \text{sign}(\sum_{k=1}^{t} \frac{\alpha_k y_k h_k(x_i)}{\sum_{r=1}^{T} \alpha_r})\)

Table 4: AdaBoost* Algorithm
AdaBoost*
Active Learning with AdaBoost*
Active Learning with AdaBoost*

Active Learning using AdaBoost*: Confidence-Based Sampling vs Random Sampling

Accuracy vs Size of Training Set

- Random-Sampling : Test Set
- Confidence-Based Sampling : Test Set
AL with AdaBoost vs AL with AdaBoost*

Active Learning with AdaBoost vs Active Learning with AdaBoost*

Accuracy vs Size of Training Set

- AdaBoost: Testing Set
- AdaBoost: Full Set
- AdaBoost*: Testing Set
- AdaBoost*: Full Set
**LPBoost Algorithm**

Given \((x_1, y_1), \ldots, (x_N, y_N) \in S_t\) where \(y_i \in \{-1, +1\}\), and accuracy parameter \(\epsilon\)

Initialize: \(d^0\) to the uniform distribution and \(\gamma_0\) to 1.

Do for \(t = 1, \ldots\)

Send \(d^{t-1}\) to oracle and obtain hypothesis \(h^t\)

Set \(u^t_n = h^t(x_n)y_n\)

Assume \(u^t d^{t-1} \geq g\), where \(g\) need not be known

Update the distribution to any \([d^t, P_{LP}^t] \in \arg\min \gamma\)

such that \(u^q d \leq \gamma\), for \(1 \leq q \leq t\), \(d_n \leq 1\), for \(1 \leq n \leq N\), and \(\sum_n d_n = 1\)

If \(\min_{q=1 \ldots t} \frac{u^q d^{q-1} - P_{LP}^t}{\epsilon} \leq \epsilon\) then set \(T = t\) and break.

Output: \(f_w(x) = \sum_{q=1}^{T} w_q h^q(x)\), where the coefficients \(w\) maximize the soft margin over the hypothesis set \(\{h^1, h^2, \ldots, h^T\}\) using the LP problem.

Table 5: LPBoost Algorithm
LPBoost

Performance of LPBoost

Generalization Error

Number of Boosting Rounds

Training Set

Testing Set
Active Learning with LPBoost
**ERLPBoost Algorithm**

Given \((x_1, y_1), \ldots, (x_N, y_N) \in S_t\) where \(y_i \in \{-1, +1\}\), accuracy parameter \(\varepsilon\) and capping parameter \(\nu \in [1, N]\)

Initialize: \(d^1\) to the uniform distribution.

Do for \(t = 1, \ldots\)

Send \(d^t\) to oracle and obtain hypothesis \(h^t\)

Set \(u^t_n = h^t(x_n)y_n\)

Set \(\delta^t = \min_{q=1..t} P^q(d^q) - P^{t-1}(d^t)\) If \(\delta^t \leq \frac{\varepsilon}{2}\) then set \(T = t - 1\) and break

Else update distribution to:

\[
[d^{t+1}, \gamma^t] \in \min_{d, \gamma} \gamma + \frac{1}{\eta} \Delta(d, d^1)
\]

s.t. \(d u^m \leq \gamma\) for \(1 \leq m \leq t\); \(d \in P_N, d \leq \frac{1}{\nu} 1\)

Assume \(u^t d^{t-1} \geq g\), where \(g\) need not be known

Update the distribution to any \([d^t, P^t_{LP}] \in \arg \min_{d, \gamma} \gamma\)

such that \(u^q d \leq \gamma\), for \(1 \leq q \leq t\), \(d_n \leq 1\), for \(1 \leq n \leq N\), and \(\sum_n d_n = 1\)

If \(\min_{q=1..t} u^q d^{q-1} - P^t_{LP} \leq \varepsilon\) then set \(T = t\) and break.

Output: \(f_w(x) = \sum_{q=1}^{T} w_q h^q(x)\), where the coefficients \(w\) maximize the soft margin over the hypothesis set \(\{h^1, h^2, \ldots h^T\}\) using the Linear Programming problem.

---

**Table 6: ERLPBoost Algorithm**
ERLPPBoost
Active Learning with ERLPBoost

![Graph showing the accuracy of ERLPBoost with respect to the size of the training set. The graph compares the training set and testing set accuracies. The accuracy increases as the size of the training set increases.](image-url)