1. LPBoost

Some details have been omitted from the main presentation for space reasons and clarity of presentation. Here we additionally provide a description of the LPBoost algorithm and a detailed motivation and derivation of the 1.5-class $\nu$-LPBoost variant.

We also provide a larger image of the unsupervised ranking results.

1.1. LPBoost Algorithm

The LPBoost algorithm is summarized in Algorithm 1. We use $\mathcal{H}$ to denote the space of possible hypothesis. For weighted substructure mining applications this is $\mathcal{H} = \{h(\cdot; t, \omega)| (t, \omega) \in T \times \Omega\}$. We denote by $h_i$ the hypothesis selected at iteration $i$.

### Algorithm 1 Linear Programming Boosting (LPBoost)

<table>
<thead>
<tr>
<th>Input:</th>
<th>Training set $X = {x_1, \ldots, x_\ell}, x_i \in X$, labels $Y = {y_1, \ldots, y_\ell}, y_i \in {-1, 1}$, convergence threshold $\theta \geq 0$.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Output:</td>
<td>The classification function $f(x) : X \rightarrow \mathbb{R}$.</td>
</tr>
<tr>
<td>1:</td>
<td>$\lambda_n \leftarrow \frac{1}{\ell}, \forall n = 1, \ldots, \ell$</td>
</tr>
<tr>
<td>2:</td>
<td>$\gamma \leftarrow 0, J \leftarrow 1$</td>
</tr>
<tr>
<td>3: loop</td>
<td>$\hat{h} \leftarrow \arg\max_{h \in \mathcal{H}} \sum_{n=1}^{\ell} y_n \lambda_n h(x_n)$</td>
</tr>
<tr>
<td>4:</td>
<td>if $\sum_{n=1}^{\ell} y_n \lambda_n \hat{h}(x_n) \leq \gamma + \theta$ then break</td>
</tr>
<tr>
<td>7:</td>
<td>end if</td>
</tr>
<tr>
<td>8:</td>
<td>$h_J \leftarrow \hat{h}$</td>
</tr>
<tr>
<td>9:</td>
<td>$J \leftarrow J + 1$</td>
</tr>
<tr>
<td>10:</td>
<td>$(\lambda, \gamma) \leftarrow$ solution to the dual of the LP problem, where $\gamma$ is the objective function value.</td>
</tr>
<tr>
<td>11:</td>
<td>$\alpha \leftarrow$ Lagrangian multipliers of solution to dual LP problem</td>
</tr>
<tr>
<td>12:</td>
<td>end loop</td>
</tr>
<tr>
<td>13:</td>
<td>$f(x) := \text{sign} \left( \sum_{j=1}^{J} \alpha_j h_j(x) \right)$</td>
</tr>
</tbody>
</table>

1.2. 1.5-class LPBoost

In Figures 1-3 the behaviour of the 1-class, 2-class and 1.5-class classifiers is shown schematically for a 2D toy example.

Given a set of positive samples $X_1 = \{x_{1,1}, \ldots, x_{1,N}\}$
and a set of negative samples $X_2 = \{x_{2,1}, \ldots, x_{2,M}\}$, we derive the following new “1.5-class LPBoost” formulation.

\[
\begin{align*}
\min \quad & \rho_2 - \rho_1 + \frac{1}{\nu N} \sum_{n=1}^{N} \xi_{1,n} + \frac{1}{\nu M} \sum_{m=1}^{M} \xi_{2,m} \\
\text{s.t.} \quad & \sum_{t \in T} \alpha_t h(x_{1,n}; t) \geq \rho_1 - \xi_{1,n}, \quad n = 1, \ldots, N \\
& \sum_{t \in T} \alpha_t h(x_{2,m}; t) \leq \rho_2 + \xi_{2,m}, \quad m = 1, \ldots, M \\
& \sum_{t \in T} \alpha_t = 1, \\
& \bm{\alpha} \in \mathbb{R}^{|T|}_+, \rho_1, \rho_2 \in \mathbb{R}_+, \xi_1 \in \mathbb{R}^N_+, \xi_2 \in \mathbb{R}^M_+
\end{align*}
\] (1)

where we directly maximize a soft-margin $(\rho_1 - \rho_2)$ that separates positive from negative training samples as illustrated in Figure 4. The hypotheses are decision stumps that reward the presence of a pattern:

\[
h(x; t) = \begin{cases} 
1 & t \subseteq x \\
0 & \text{otherwise.}
\end{cases}
\] (2)

The class decision function is given by thresholding at the margin’s center $(\rho_1 + \rho_2)/2$, such that

\[
f(x) = \text{sign} \left( \sum_{t \in T} \alpha_t h(x; t) - \frac{\rho_1 + \rho_2}{2} \right).
\] (3)

Problem (1) can be solved by the LPBoost algorithm [1] using the following dual LP problem.

\[
\begin{align*}
\max \quad & \lambda, \mu, \gamma \\
\text{s.t.} \quad & \sum_{n=1}^{N} \lambda_n h(x_{1,n}; t) \geq \sum_{m=1}^{M} \mu_m h(x_{2,m}; t) \leq \gamma, \\
& 0 \leq \lambda_n \leq \frac{1}{\nu N}, \quad n = 1, \ldots, N \\
& 0 \leq \mu_m \leq \frac{1}{\nu M}, \quad m = 1, \ldots, M.
\end{align*}
\] (4)

For solving Problem (4) we again use column-generation techniques, incrementally adding the most violated constraint.

Similarly to the original 2-class LPBoost formulation we derive the gain function from the constraints on the hypotheses outputs of the dual of (1) to obtain

\[
\hat{h} = \arg\max_{h \in \mathcal{H}} \left[ \sum_{n=1}^{N} \lambda_n h(x_{1,n}) - \sum_{m=1}^{M} \mu_m h(x_{2,m}) \right],
\] (6)

which is the same as for the 2-class case, except that the set of samples are explicitly split into two sums. For performing weighted substructure mining efficiently we need a bound on the gain for a pattern $t'$. The bound shall be evaluated using only $t$, where $t \subseteq t'$ is subpattern of $t'$; this allows efficient pruning in the mining algorithm. For the new formulation we derive the following new bound on the gain function. Using the anti-monotonicity property [2] for
any \( t \subseteq t' \) we have

\[
\text{gain}(t') = \sum_{n=1}^{N} \lambda_n h(x_{1,n}; t') - \sum_{m=1}^{M} \mu_m h(x_{2,m}; t') \\
= \sum_{n=1}^{N} \lambda_n I(t' \subseteq x_{1,n}) - \sum_{m=1}^{M} \mu_m I(t' \subseteq x_{2,m}) \\
\leq \sum_{n=1}^{N} \lambda_n I(t' \subseteq x_{1,n}) \\
\leq \sum_{n=1}^{N} \lambda_n I(t \subseteq x_{1,n}).
\]

A drawback of the new formulation (1) is the violation of the closed-under-complementation assumption of Demiriz et al. [1], hence we are not guaranteed to obtain the optimal solution \((H, \alpha)\) among all possible sets and weightings. In practice this never caused any problems and the convergence behavior measured by an independent test error is very similar to the 2-class case.

The 1.5-class formulation (1) is a generalization of 1-class \(\nu\)-Boosting in Rätsch et al. [3] and we recover the original formulation when \(M = 0\), that is, when no negative samples are available.

References


