Different Boosting Algorithms and Underlying Optimization Problems

Maya Hristakeva

University of California, Santa Cruz

March 21, 2008
1. Boosting Setting
2. LPBoost
3. Entropy Regularized LPBoost
4. TotalBoost and SoftBoost
5. Symmetric SoftMaxBoost
6. Experiments
7. Conclusion
Outline

1. Boosting Setting
2. LPBoost
3. Entropy Regularized LPBoost
4. TotalBoost and SoftBoost
5. Symmetric SoftMaxBoost
6. Experiments
7. Conclusion
Protocol of Boosting

- Given N training examples $(x_1, y_1), \ldots, (x_n, y_n)$
- $y_i \in \{-1, 1\}$ correct label of instance $x_i \in X$
- Maintain a distribution $d^t \in [0, 1]^n$ on the examples
- In each iteration $t = 1, 2, \ldots, T$
  - Weak learner provides a new base hypothesis $h_t$
  - Update $d^t$ to $d^{t+1}$, put more weights on “hard examples”
- Output the convex combination of the weak hypothesis

\[ f_\alpha(x) = \sum_{t=1}^{T} \alpha_t h_t(x) \]
Margin vs. Edge

**Edge**

- Measurement of “goodness” of a hypothesis \((h_t)\) w.r.t. a distribution

\[
\gamma_h(d) = \sum_{n=1}^{N} d_n y_n h(x_n) \quad d \in P^n
\]

\[
\gamma_h(d) = u^t \cdot d \quad u^t = yh_t(x)
\]

**Margin**

- Measurement of “confidence” in prediction for a hypothesis weighting

\[
\rho_n(\alpha) = y_n \sum_{t=1}^{T} \alpha_t h_t(x_n) \quad \alpha \in P^t
\]
Outline

1. Boosting Setting
2. LPBoost
3. Entropy Regularized LPBoost
4. TotalBoost and SoftBoost
5. Symmetric SoftMaxBoost
6. Experiments
7. Conclusion
LPBoost (Hard Margin)

- Given a set of hypothesis \{h^1, ..., h^t\}
- Predict with any distribution that minimizes the maximum edge of the t hypothesis seen so far (Totally Corrective update).

\[ d^t \in \arg\min_{d \in S^N} \max_{t=1,\ldots,T} u^t \cdot d \]

- By duality

\[ \gamma^* = \min_{d \in S^N} \max_{t=1,\ldots,T} u^t \cdot d = \max_{\alpha} \min_{n=1,\ldots,N} y_n f_\alpha(x_n) = \rho^* \]
LPBoost (Soft Margin)

- Allow for some examples to lie below the margin
- Penalize via slack variables $\psi_n$

$$\max_{\alpha \in S, n=1, \ldots, N} \min_{\psi \geq 0} \left( \sum_{t=1}^{T} u^n_t \alpha_t + \psi_n \right) - \frac{1}{\nu} \sum_{n=1}^{N} \psi_n$$

$$d^t \in \min_{d \in S^N} \max_{t=1, \ldots, T} u^t \cdot d\quad \text{for} \quad d \leq \frac{1}{\nu} 1$$

- Distribution is capped by $1/\nu$ for $\nu \in [1, N]$
Outline

1. Boosting Setting
2. LPBoost
3. Entropy Regularized LPBoost
4. TotalBoost and SoftBoost
5. Symmetric SoftMaxBoost
6. Experiments
7. Conclusion
Entropy Regularized LPBoost

- Given a set of hypothesis \( \{h^1, \ldots, h^t\} \) the optimization problem is:

\[
\min_{d \in S^N} \max_{t=1,\ldots,T} u^t \cdot d + \frac{1}{\eta} \triangle (d, d^1)
\]

- Objective function is strictly convex, thus optimization problem has unique solution
- Has logarithmic bound
- \( \eta \to \infty \): the Entropy Regularized LPBoost turns into the totally corrective LPBoost with soft margin
Outline

1. Boosting Setting
2. LPBoost
3. Entropy Regularized LPBoost
4. TotalBoost and SoftBoost
5. Symmetric SoftMaxBoost
6. Experiments
7. Conclusion
**Motivation**

- **LPBoost**: predicts with any distribution that minimizes the maximum edge of the \( t \) hypothesis seen so far
- **TotalBoost and SoftBoost**: motivated by the minimum entropy principle of Jaynes
  - Among the solutions satisfying some linear constraint choose the one that minimizes \( \triangle(d, d^1) \)
  - Ensures the optimization problems have unique solutions
TotalBoost solves:

\[ \mathbf{d}^{t+1} = \min_{\mathbf{d}} \triangle (\mathbf{d}, \mathbf{d}^1) \]

s.t. \( \mathbf{d} \cdot \mathbf{u}^m \leq \hat{\gamma}_t - \epsilon \), for \( 1 \leq m \leq t \); \( \mathbf{d} \in P^N \)

SoftBoost solves:

\[ \mathbf{d}^{t+1} = \min_{\mathbf{d}} \triangle (\mathbf{d}, \mathbf{d}^1) \]

s.t. \( \mathbf{d} \cdot \mathbf{u}^m \leq \hat{\gamma}_t - \epsilon \), for \( 1 \leq m \leq t \); \( \mathbf{d} \in P^N \), \( \mathbf{d} \leq \frac{1}{\nu} \mathbf{1} \)

where \( \hat{\gamma}_t = \min_{m=1,...,t} \mathbf{d}^m \cdot \mathbf{u}^m \).
Outline

1. Boosting Setting
2. LPBoost
3. Entropy Regularized LPBoost
4. TotalBoost and SoftBoost
5. Symmetric SoftMaxBoost
6. Experiments
7. Conclusion
Given a set of hypothesis \( \{ h^1, ..., h^t \} \) the optimization problem is:

\[
\min_{\mathbf{d}} \frac{1}{\eta} \sum_{i=1}^{n} d_i \log \frac{d_i}{d_1} + \lambda \log(\sum_{j=1}^{t} \exp(\lambda^{-1} \xi_j))
\]

s.t. \( \| \mathbf{d} \|_1 = 1 \)

\( \mathbf{u}^j \mathbf{d} = \xi_j \)

- Minimizes the softmax over all edges (not the maximum edge)
- Softmax function is convex, thus the optimization problem has a unique solution
Dual Problem

- Maximizes the softmin over the margins of all examples
  \[
  \max_{\alpha} - \frac{1}{\eta} \log \sum_{i=1}^{n} d_i^1 \exp(-\eta u_i; \alpha) - \lambda \sum_{j=1}^{t} \alpha_j \log(\alpha_j)
  \]
  s.t. \( ||\alpha||_1 = 1, \alpha \geq 0 \)

- Symmetry between the primal and the dual
Symmetric SoftMaxBoost

Connection to LPBoost and Entropy Regularized LPBoost

- If $\lambda \to 0$ then get the maximum edge from the softmax function
- Symmetric SoftMaxBoost $\to$ Entropy Regularized LPBoost

\[
\max_{\mathbf{u}^{i} \mathbf{d}} = \lim_{\lambda \to 0} \lambda \log \left( \sum_{j=1}^{t} \exp \left( \lambda^{-1} \xi_j \right) \right)
\]

- If $\lambda \to 0$ and $\eta \to \infty$ then Symmetric SoftMaxBoost $\to$ totally corrective LPBoost with hard margin
Algorithm

Input: \( S = \langle (x_1, y_1), ..., (x_n, y_n) \rangle \) with parameters \( \eta \) and \( \lambda \)

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

Do for \( t = 1, ..., T \)

- Send \( d^t \) to weak learner and obtain hypothesis \( h^t \).
  - Set \( u_n^t = h^t(x_n)y_n \).
- Update the distribution to
  \[
  d^{t+1} = \min_d \frac{1}{\eta} \sum_{i=1}^n d_i \log \frac{d_i}{d_i^t} + \lambda \log(\sum_{j=1}^t \exp(\lambda^{-1} \xi_j))
  \]
  s.t. \( \|d\|_1 = 1 \), \( u^t d = \xi_t \)
- If above problem is infeasible then \( T = t \) and break.

Output: \( f_\alpha(x) = \sum_{t=1}^T \alpha_t h_t(x) \), where the coefficients \( \alpha_t \) maximize the hard margin over the hypothesis using LP problem (1).
Outline

1. Boosting Setting
2. LPBoost
3. Entropy Regularized LPBoost
4. TotalBoost and SoftBoost
5. Symmetric SoftMaxBoost
6. Experiments
7. Conclusion
Datasets

- Sonar dataset
  - 208 objects and 60 attributes
  - from the UCI benchmark repository
- Simulated data using the following model:
  \[ P(Y = 1|x) = q + (1 - 2q)I \left[ \sum_{j=1}^{J} x^{(j)} > J/2 \right] \]
  - \( q \) - the Bayes error
  - \( J \leq d \) - the number of effective dimensions

Parameters

- \( \epsilon = 0.01 \)
- \( \nu/N = 0.1 \) or \( \nu = 2 \times q \)
- \( \eta = \frac{2}{\epsilon} \ln \frac{N}{\nu} \)
Hard Margin (Sonar Data)

\[
\lambda = 0.01 \text{ and } \frac{1}{\eta} = 0.025
\]
Experiments

Hard Margin (Simulated Data)

$\lambda = 10$, and $\frac{1}{\eta} = 0.025$
Soft Margin (Sonar Data)

Symmetric SoftMaxBoost with $\lambda = 0.01$ and $\frac{1}{\eta} = 0.025$. 
Soft Margin (Simulated Data)

Data model parameters: \( q = 0.1, n = 100, J = 20, d = 5 \)

Symmetric SoftMaxBoost with \( \lambda = 10 \) and \( \frac{1}{\eta} = 0.025 \).
Generalization Error (Sonar Data)

Symmetric SoftMaxBoost with $\lambda = 0.01$ and $\frac{1}{\eta} = 0.025$. 
Generalization Error (Simulated Data)

Data model parameters: $q = 0.1$, $n = 100$, $J = 20$, $d = 5$

Symmetric SoftMaxBoost with $\lambda = 0.01$ and $\frac{1}{\eta} = 0.025$. 
Experiments

Entropy Regularized LPBoost varying $\eta$

$$\eta = \frac{2}{\epsilon} \ln \frac{N}{\nu}, \quad \eta_2 = 100\eta, \text{ and } \eta_3 = 0.01\eta$$
Outline

1. Boosting Setting
2. LPBoost
3. Entropy Regularized LPBoost
4. TotalBoost and SoftBoost
5. Symmetric SoftMaxBoost
6. Experiments
7. Conclusion
Summary

- Symmetric SoftMaxBoost has performance comparable to LPBoost and Entropy Regularized LPBoost
  - starts as quickly and levels off as it reaches the softmax over the edges
  - quickly reaches low generalization error and does not overfit
- Do not know of any iteration bound
- Performance could vary for different data
- Many thanks to Karen Glocer for her continuous assistance regarding implementation issues
Questions

The End!