



Group Meeting: Speed and Sparsity of Regularized Boosting

Yongxin Taylor Xi
Zhen James Xiang
July 11, 2008

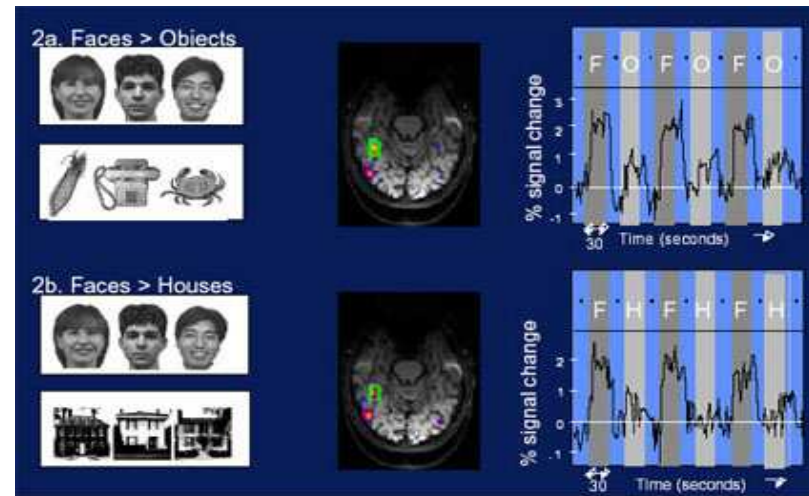
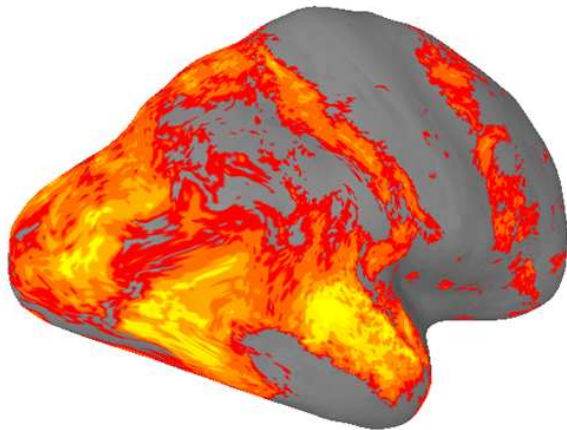
Courtesy slides from Peter Ramadge.

Sparse Boosting

Learning a classifier from labelled examples.

Identify locus of brain activity for specific processing task.

(Work in progress)



Classifiers, Classifier Edge

Binary classifier: $h: X \rightarrow \{-1, +1\}$

Labelled training examples:

$$S = \langle (x_1, y_1), \dots, (x_m, y_m) \rangle$$

Distribution on examples: $d \in \Delta_m$

Edge of h wrt d :

$$e(h, d) = E_{i \sim d}[y_i h(x_i)] = \sum_{i=1}^m d_i y_i h(x_i).$$

$$e(h, d) \in [-1, 1]$$

$$\Pr_{i \sim d}[y_i \neq h(x_i)] = \frac{1}{2}(1 - e(h, d))$$

Weak Classifiers

Weak classifiers:

$$\mathcal{H} = \{h_1, h_2, \dots, h_N\}$$

S **weakly learnable** under \mathcal{H} :

$$\exists \theta > 0: \quad \forall d \in \Delta_m \quad \exists h_j \in \mathcal{H}: e(h_j, d) \geq \theta$$

i.e., $\exists h_j \in \mathcal{H}$ that performs better than chance.

Define θ^* : $\theta^* > 0$ is the largest such θ .

The Idea of Boosting

Find a “good” composite classifier:

$$h_{\alpha}(x) = \sum_{j=1}^N \alpha_j h_j(x) \quad z = \text{sign}(h_{\alpha}(x))$$

How to select α ?

- a. Use the labelled examples.
- b. Want to use few weak classifiers.
- c. In general, N is very large, m is small.
- d. Do not know θ^* .
- e. Want good generalization.

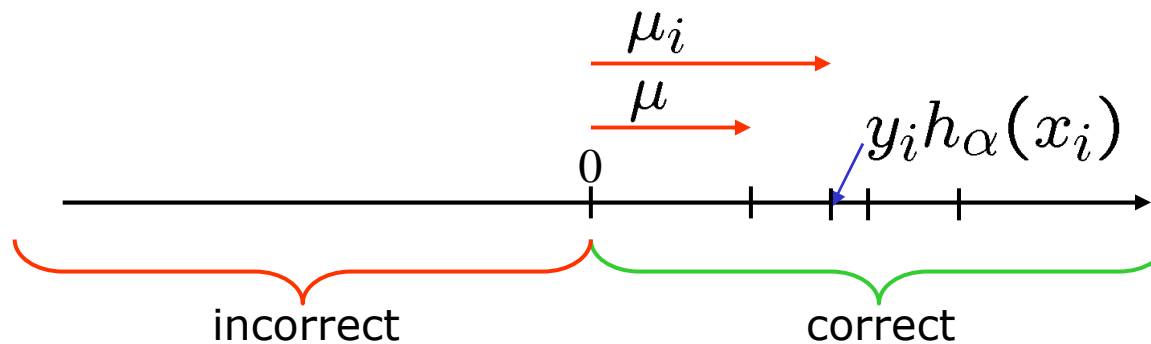
Margin

Normalized composite classifier:

$$h_{\alpha}(x) = \sum_{j=1}^N \alpha_j h_j(x), \quad \|\alpha\|_1 = 1$$

Margin on i th example: $\mu_i = y_i h_{\alpha}(x_i)$.

Margin of classifier: $\mu(h_{\alpha}) = \min_i \mu_i$.



How Good can a Comp. Classifier be?

How good can h_α be?

The maximum margin classifier has:

$$\mu = \max_{\alpha} \left[\min_i \sum_j \alpha_j y_i h_j(x_i) \right]$$

By von Neumann min-max theorem:

$$\max_{\alpha} \mu(h_{\alpha}) = \theta^*$$

$$\exists \alpha^*: \mu(h_{\alpha^*}) = \theta^* > 0$$

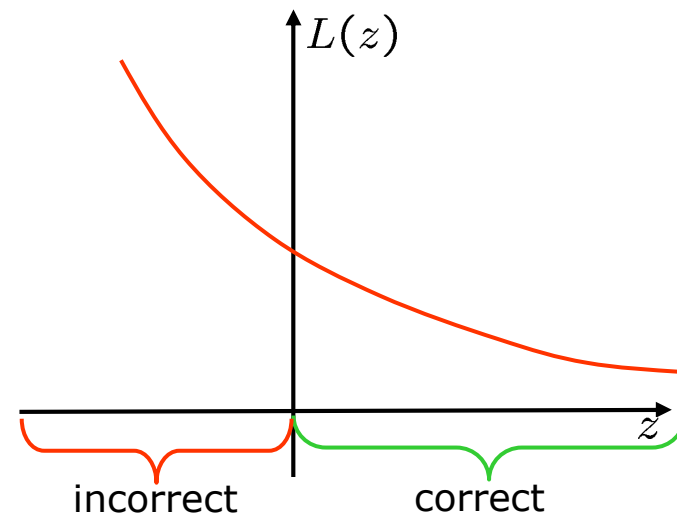
Loss Minimization

Approach:

Pick a loss function, e.g. $L(z) = \exp(-z)$

Minimize total loss on the examples

$$\mathcal{L}(\alpha) = \sum_{i=1}^m L(y_i h_{\alpha}(x_i))$$



AdaBoost: Iterative coordinate descent

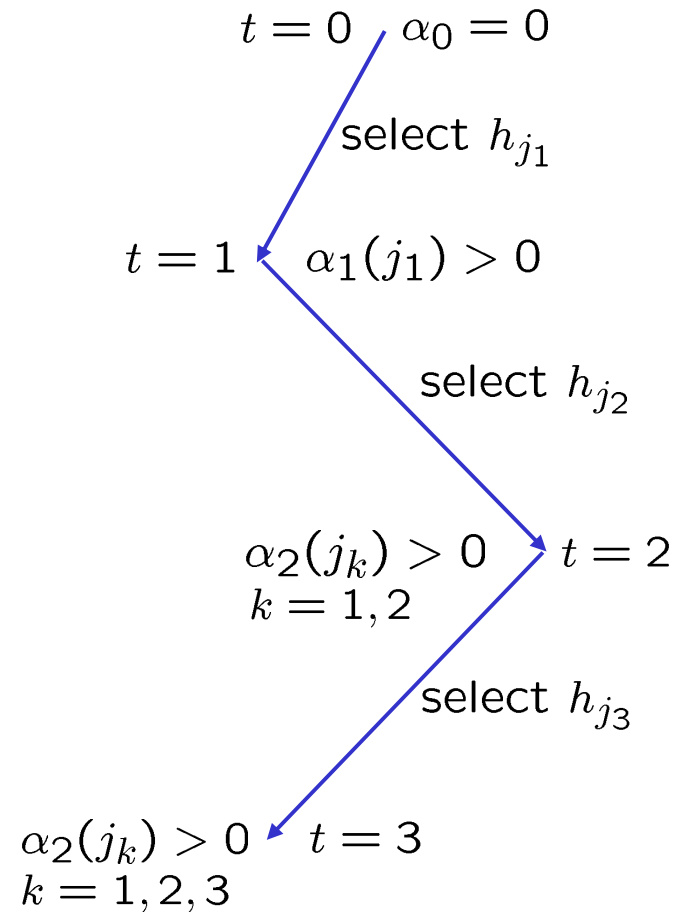
$$L(z) = \exp(-z)$$

$$\alpha_0(j) = 0 \in \mathbb{R}^N,$$

$$d_0 = \frac{1}{m} \mathbf{1} \in \mathbb{R}^m.$$

For $t = 1, 2, \dots$

- * Select a good weak classifier (large edge).
- * Increase its coefficient by line search.
- * Reweight the m training examples.



AdaBoost: attributes

- Fast.
- Counter t controls the complexity of the composite classifier.
- Empirically:
 - achieves large margins (but simple examples known where it does not achieve the maximum margin).
 - Generalizes well to new data.
- Often used metrics:
 - size of the margin
 - Complexity of the composite classifier

Regularized Loss Minimization: Sparsity

$$\begin{aligned} \min_{\alpha} \mathcal{L}(\alpha) &= \sum_{i=1}^m \exp \left(-y_i \sum_{j=1}^N \alpha_j h_j(x_i) \right) \\ \text{s.t. } \|\alpha\|_1 &\leq r \\ \alpha_j &\geq 0, \quad j = 1, 2, \dots, N \end{aligned}$$

T. Hastie et al., *"The Elements of Statistical Learning"*, 2001

Advantages/Disadvantages:

1. sparse solutions
2. margin maximizing as regularization relaxed [Rosset et al., 2004]
3. large computational burden (\Rightarrow epsilon-boosting).

Convergence Rate

Convergence Rate Theorem

(Xi, Xiang, Ramadge, Schapire, 2008)

Let $h_{\alpha(r)}$ be the composite classifier resulting from solution $\alpha(r)$ of the regularized loss minimization problem. Then

$$r > \frac{\ln(m\delta^{-1}(1 - \theta^*))}{\delta} \quad \Rightarrow \quad 0 \leq \theta^* - \mu(h_{\alpha(r)}) \leq \delta$$

Convergence Rate: more general case

Convergence Rate Theorem (Xi, Xiang, Ramadge, Schapire, 2008)

Let function L be convex and strictly monotone decreasing. Let a solution $\alpha(r)$ of the L_1 regularized boosting problem define a composite classifier $h_{\alpha(r)}$. Then the margin of $h_{\alpha(r)}$ is at least $(\theta^* - \delta)$ when

$$\frac{L'(r\theta^*)}{L'(r(\theta^* - \delta))} < \frac{\delta}{m(1 - \theta^*)}.$$

Moreover, if

$$\forall \epsilon > 0, \lim_{x \rightarrow \infty} \frac{L'(x)}{L'(x(1 - \epsilon))} = 0,$$

then $\lim_{r \rightarrow \infty} \mu(h_{\alpha(r)}) = \theta^*$.

Convergence Rate: proof

Proof sketch:

1. uniform learning lemma: Equal edge ($> \theta^*$) of active base classifiers.
2. splitting margins of all examples into 3 groups:
 $\mu_i \leq \theta^* - \delta$, $\theta^* - \delta < \mu_i < \theta^*$, $\mu_i \geq \theta^*$.
3. bound the number of margins below $\theta^* - \delta$.

New Algorithm: AdaBoost+L1

$$\alpha_0 = \mathbf{0} \in \mathbf{R}^N, \quad d_0 = \frac{1}{m}\mathbf{1}, \\ U_0 = \emptyset, \quad r_0 = 0.$$

For $t = 1, 2, \dots$

1) Find $h_k \in \mathcal{H}$: $e(h_k, d_{t-1}) \geq \theta^*$.

2a) $U_t = U_{t-1} \cup \{k\}$,

2b) $r_t = r_{t-1} + \frac{1}{2} \ln \frac{1+e(h_k, d_{t-1})}{1-e(h_k, d_{t-1})}$.

3) Solve the (small) convex problem over $\{\alpha_j\}_{j \in U_t}$:

$$\min \sum_{i=1}^m \exp \left(-y_i \sum_{j \in U_t} \alpha_j h_j(x_i) \right) \\ \text{s.t.} \quad \sum_{j \in U_t} \alpha_j \leq r_t, \quad \alpha_j \geq 0$$

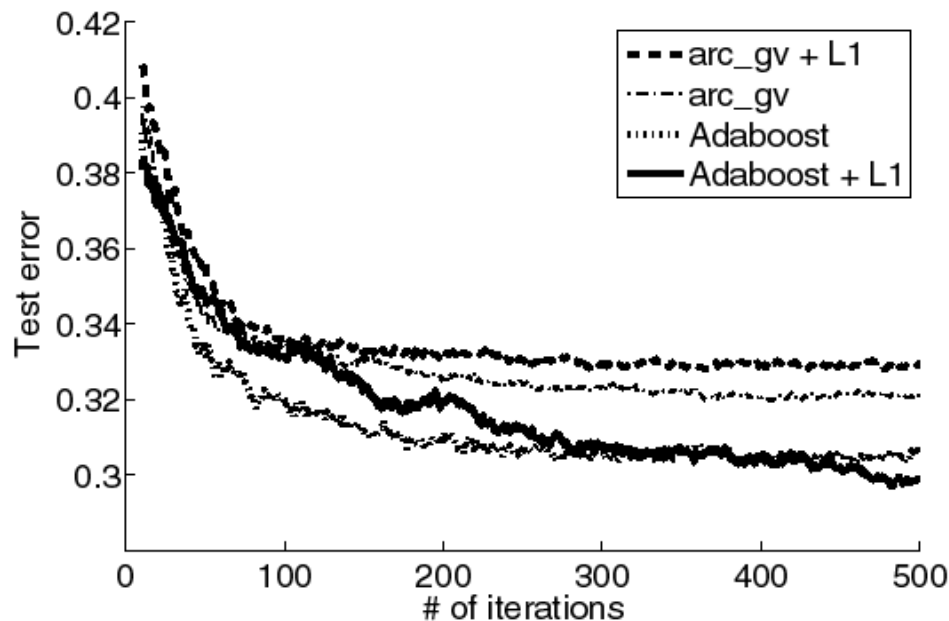
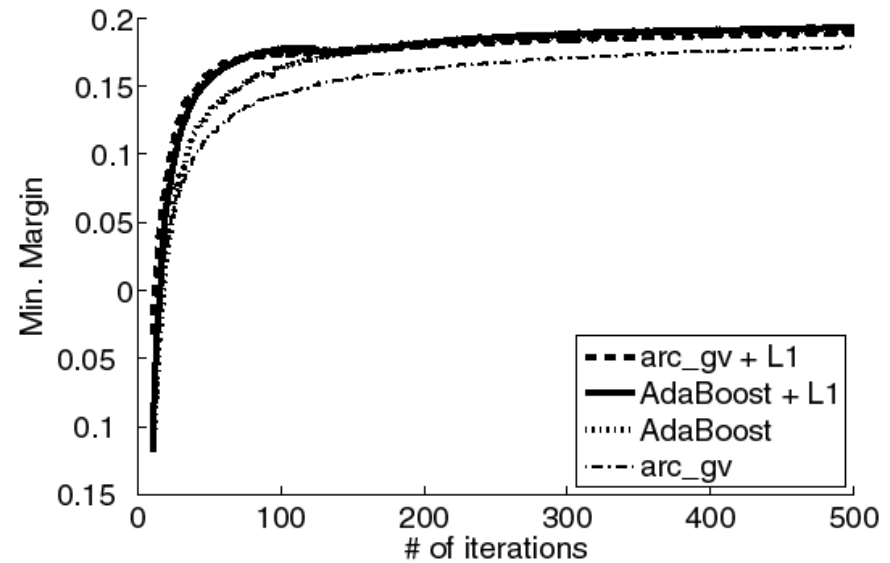
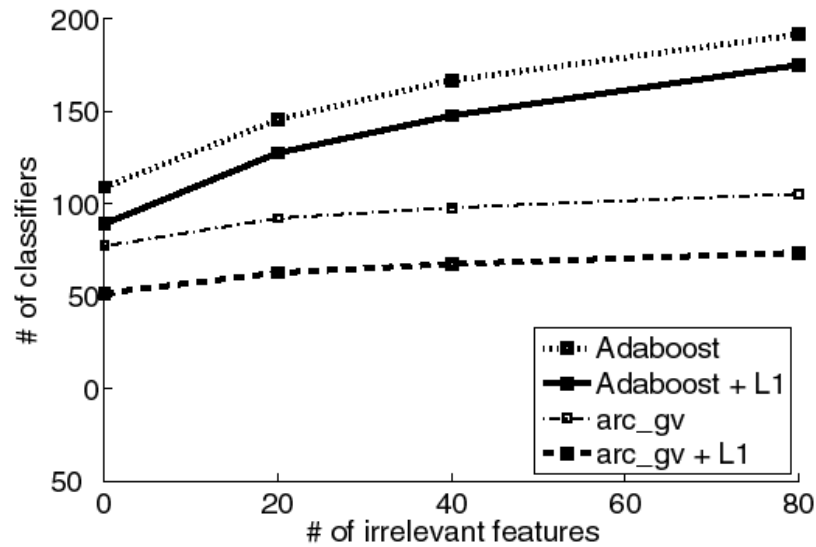
4) $\alpha_t(j) = \alpha_j^*$, if $j \in U_t$; 0, otherwise.

5) $d_t(i) = \frac{\exp(-y_i \sum_{j=1}^N \alpha_t(j) h_j(x_i))}{\sum_{i=1}^m \exp(-y_i \sum_{j=1}^N \alpha_t(j) h_j(x_i))}$,
 $i = 1, \dots, N$.

Theorem

(Xi, Xiang, Ramadge, Schapire)

In the limit, Adaboost+ L_1 achieves the maximum margin θ^* .



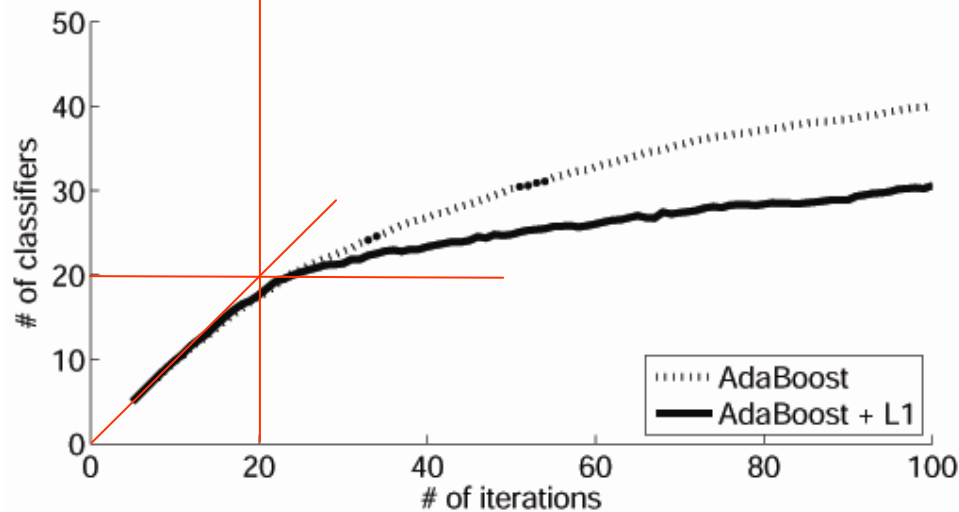
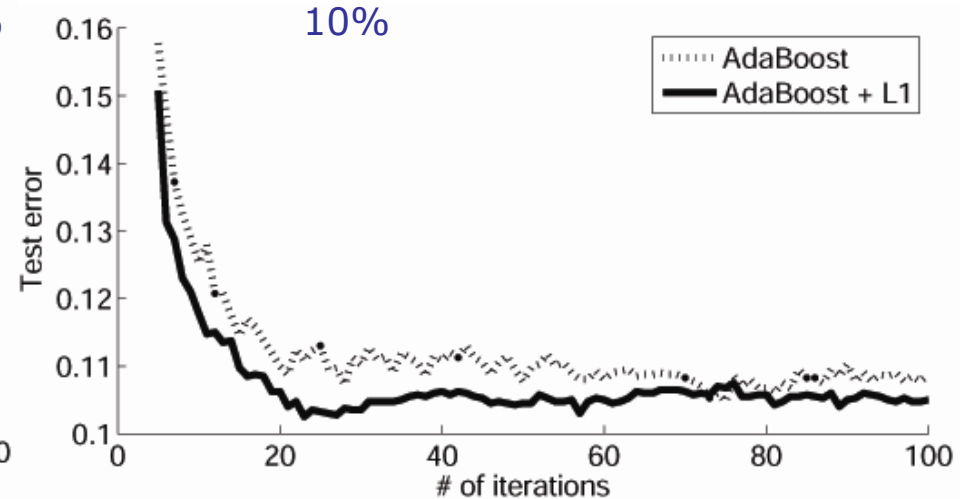
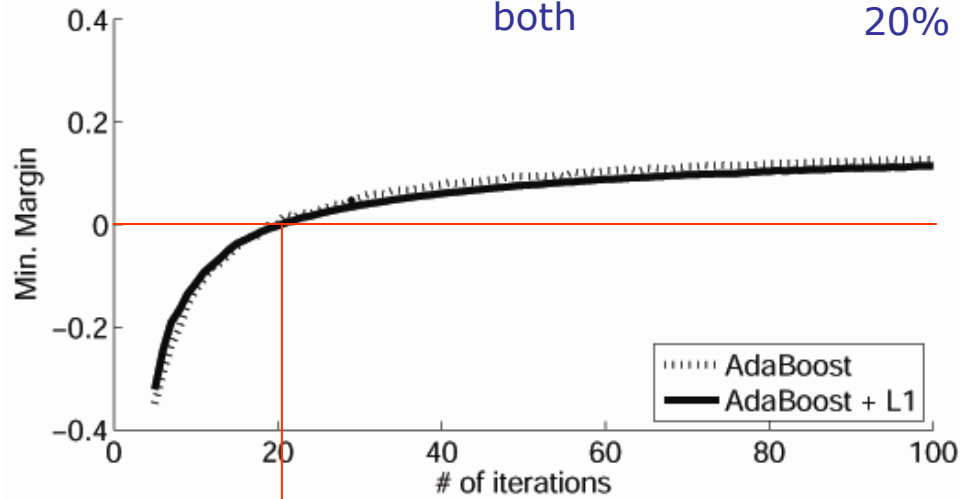
Example 1: Breiman's ringnorm.
 Synthetic dataset with 0, 20, 40, 80 irrelevant features added.

100 random training examples,
 500 random testing examples.
 Averaged results for 20 runs.

Weak classifiers are simple
 decision stumps.

Examples: free, \$, !, ..., you, mail, (, ..., all, people, re, ...

} both
} 20%
} 10%



Example 2: SPAM (UCI repository)

4601 training examples, 57 features (words and symbols from text email). Randomly selected 100 training examples, 200 testing examples and repeated 20 times.

Weak classifiers are simple decision stumps.