New understanding of boosting methods

Chunhua Shen, School of Computer Science, University of Adelaide

April 2013
Talk agenda

What to talk today

- A duality view of boosting algorithms
- A fully corrective boosting framework for minimizing regularized risk with arbitrary loss and regularization
  - Optimally training a cascade classifier with boosting
- A direct approach to mult-class boosting
Talk agenda

What to talk today

- A duality view of boosting algorithms
- A fully corrective boosting framework for minimizing regularized risk with arbitrary loss and regularization
  - Optimally training a cascade classifier with boosting
- A direct approach to mult-class boosting
New understanding of boosting methods

Talk agenda

What to talk today

- A duality view of boosting algorithms
- A fully corrective boosting framework for minimizing regularized risk with arbitrary loss and regularization
  - Optimally training a cascade classifier with boosting
- A direct approach to mult-class boosting
New understanding of boosting methods

What is boosting

- Boosting builds a very accurate classifier by combining rough and only moderately accurate classifiers.

- Boosting procedures
  - Given a set of labeled training examples
  - On each round
    1. The booster devises a distribution (importance) over the example set
    2. The booster requests a weak hypothesis/classifier/learner with low error
  - Upon convergence, the booster combine the weak hypothesis into a single prediction rule.
What is boosting

- Boosting builds a very accurate classifier by combining rough and only moderately accurate classifiers.

- **Boosting procedures**
  - Given a set of labeled training examples
  - On each round
    1. The booster devises a distribution (importance) over the example set
    2. The booster requests a weak hypothesis/classifier/learner with low error
  - Upon convergence, the booster combines the weak hypothesis into a single prediction rule.
Why boosting works

- Let $\mathcal{H}$ be a class of base classifier
  $\mathcal{H} = \{h_j(\cdot): \mathcal{X} \rightarrow \mathbb{R}\}, j = 1 \cdots N$, a boosting algorithm seeks for a convex combination:

  $$F(\mathbf{w}) = \sum_{j=1}^{N} w_j h_j(\mathbf{x})$$

- Statistical view [Friedman et al. 2000], maximum margin [Schapire et al. 1998], still there are open questions [Mease & Wyner 2008]

- The Lagrange dual problems of AdaBoost, LogitBoost and soft-margin LPBoost with generalized hinge loss are all entropy maximization problems [Shen & Li 2010 TPAMI]
Why boosting works

- Let $\mathcal{H}$ be a class of base classifier $\mathcal{H} = \{h_j(\cdot) : \mathcal{X} \to \mathbb{R}\}, j = 1 \cdots N$, a boosting algorithm seeks for a convex combination:

$$F(w) = \sum_{j=1}^{N} w_j h_j(x)$$

- Statistical view [Friedman et al. 2000], maximum margin [Schapire et al. 1998], still there are open questions [Mease & Wyner 2008]

- The Lagrange dual problems of AdaBoost, LogitBoost and soft-margin LPBoost with generalized hinge loss are all entropy maximization problems [Shen & Li 2010 TPAMI]
Why boosting works

- Let $\mathcal{H}$ be a class of base classifier
  $\mathcal{H} = \{h_j(\cdot) : \mathcal{X} \rightarrow \mathbb{R}\}, j = 1 \cdots N$, a boosting algorithm seeks for a convex combination:

  $$F(\mathbf{w}) = \sum_{j=1}^{N} w_j h_j(\mathbf{x})$$

- Statistical view [Friedman et al. 2000], maximum margin [Schapire et al. 1998], still there are open questions [Mease & Wyner 2008]

- The Lagrange dual problems of AdaBoost, LogitBoost and soft-margin LPBoost with generalized hinge loss are all entropy maximization problems [Shen & Li 2010 TPAMI]
A duality view of boosting

Explicitly find a meaningful Lagrange dual for some boosting algorithms

Dual of AdaBoost

The Lagrange dual of AdaBoost is a Shannon entropy maximization problem:

$$\max_{r,u} \frac{r}{T} - \sum_{i=1}^{M} u_i \log u_i, \quad \text{s.t.} \sum_{i=1}^{M} y_i u_i H_i \leq -r 1^\top, \ u \geq 0, 1^\top u = 1.$$  

Here $H_i = [H_{i1} \ldots H_{iN}]$ denotes $i$-th row of $H$, which constitutes the output of all weak classifiers on $x_i$. 

New understanding of boosting methods
Primal of AdaBoost (Note the auxiliary variables $z_i, i = 1, \cdots$)

\[
\min_w \log \left( \sum_{i=1}^{M} \exp z_i \right),
\]

s.t. $z_i = -y_i H_i w \ (\forall i = 1, \cdots, M),$

\[w \geq 0, \ 1^\top w = \frac{1}{T}.\]
A duality view of boosting

Summary:

<table>
<thead>
<tr>
<th>algorithm</th>
<th>loss in primal</th>
<th>entropy reg. LPBoost in dual</th>
</tr>
</thead>
<tbody>
<tr>
<td>adaboost</td>
<td>exponential loss</td>
<td>Shannon entropy</td>
</tr>
<tr>
<td>logitboost</td>
<td>logistic loss</td>
<td>binary relative entropy</td>
</tr>
<tr>
<td>soft-margin $\ell_p(p &gt; 1)$ LPBoost</td>
<td>generalized hinge loss</td>
<td>Tsallis entropy</td>
</tr>
</tbody>
</table>
New understanding of boosting methods

Average margin vs. margin variance

Why AdaBoost just works?

Theorem:
AdaBoost approximately maximizes the average margin and at the same time minimizes the variance of the margin distribution under the assumption that the margin follows a Gaussian distribution.

Proof: See [Shen & Li 2010 TPAMI]. Main tools used:

1. Central limit theorem;
2. Monte Carlo integral.
What this theorem tells us:

1. We should focus on optimizing the overall margin distribution. Almost all previous work on boosting has focused on a large minimum margin.

2. Answered an open question in [Reyzin & Schapire 2006], [Mease & Wyner 2008]

3. We can design new boosting algorithm to directly maximize the average margin and minimize the margin variance [Shen & Li, 2010 TNN]
New understanding of boosting methods

Average margin vs. margin variance

What this theorem tells us:

1. We should focus on optimizing the overall margin distribution. Almost all previous work on boosting has focused on a large minimum margin.
2. Answered an open question in [Reyzin & Schapire 2006], [Mease & Wyner 2008]
3. We can design new boosting algorithm to directly maximize the average margin and minimize the margin variance [Shen & Li, 2010 TNN]
New understanding of boosting methods

Average margin vs. margin variance

What this theorem tells us:

1. We should focus on optimizing the overall margin distribution. Almost all previous work on boosting has focused on a large minimum margin.

2. Answered an open question in [Reyzin & Schapire 2006], [Mease & Wyner 2008]

3. We can design new boosting algorithm to directly maximize the average margin and minimize the margin variance [Shen & Li, 2010 TNN]
New understanding of boosting methods

**A better margin distribution: MDBBoost**

\[
\max_w \bar{\rho} - \frac{1}{2} \sigma^2, \text{s.t. } w \geq 0, 1^\top w = T.
\]

It is equivalent to

\[
\min_{w, \rho} \frac{1}{2} \rho^\top A \rho - 1^\top \rho,
\]

s.t. \( w \geq 0, 1^\top w = T, \rho_i = y_i H_i w, \forall i = 1, \ldots, M. \)

Its dual is

\[
\min_{r, u} r + 1/(2T)(u - 1)^\top A^{-1}(u - 1), \text{s.t., } \sum_{i=1}^{M} y_i u_i H_i \leq r 1^\top.
\]
Totally corrective boosting using column generation

We can design column generation based boosting based on the primal problems and their corresponding dual problems.
Results of Wilcoxon Signed-Ranks Test (WSRT) on 13 UCI machine learning datasets.

<table>
<thead>
<tr>
<th></th>
<th>AdaBoost</th>
<th>AdaBoost-CG</th>
<th>LPBoost</th>
<th>MDBoost</th>
</tr>
</thead>
<tbody>
<tr>
<td>AdaBoost</td>
<td>–</td>
<td>Not Better (12*)</td>
<td>Not Better (43)</td>
<td>Not Better (7)</td>
</tr>
<tr>
<td>AdaBoost-CG</td>
<td>Better (66*)</td>
<td>–</td>
<td>Better (78*)</td>
<td>Not Better (18)</td>
</tr>
<tr>
<td>LPBoost</td>
<td>Not Better (48)</td>
<td>Not Better (0*)</td>
<td>–</td>
<td>Not Better (5)</td>
</tr>
<tr>
<td>MDBoost</td>
<td>Better (84)</td>
<td>Better (73)</td>
<td>Better (86)</td>
<td>–</td>
</tr>
</tbody>
</table>

1. Our boosting algorithms AdaBoost-CG and the MDBoost outperform AdaBoost and LPBoost in terms of test error.

2. In general, CG based totally corrective boosting converges much faster than conventional stage-wise boosting.

Chunhua Shen, Hanxi Li. “Boosting through optimization of margin distributions” 2010 IEEE Trans. Neural Networks
Totally corrective boosting for regularized risk minimization

1. A general framework that can be used to designed new boosting algorithms.

2. The proposed boosting framework, termed CGBoost, can accommodate various loss functions and different regularizers in a totally-corrective optimization way.
The primal problem of $\ell_1$ norm regularized risk minimization:

$$\min_{\mathbf{w}, \gamma} \sum_{i=1}^{m} \lambda(\gamma_i) + \nu \cdot \mathbf{1}^\top \mathbf{w}$$

s.t. $\gamma_i = y_i H_i \mathbf{w}$ ($\forall i = 1 \cdots m$), $\mathbf{w} \geq 0$.

The dual problem is:

$$\min_{\mathbf{u}} \sum_{i=1}^{m} \lambda^*(u_i), \; \text{s.t.} \; \mathbf{u}^\top \text{diag}(\mathbf{y}) H \geq -\nu \mathbf{1}^\top.$$
The $\ell_\infty$-norm regularized boosting can be written as

$$
\min_{w, \gamma} \sum_{i=1}^{m} \lambda(\gamma_i) \\
\text{s.t. } 0 \leq w \leq r1, \gamma_i = y_i H_i w \ (\forall i = 1 \ldots m).
$$

Its corresponding Lagrange dual is

$$
\min_{u, s} \sum_{i=1}^{m} \lambda^*(-u_i) + r1^T s \\
\text{s.t. } u^T \text{diag}(y) H \leq s^T, \\
s \geq 0.
$$
\[ \ell_2 \text{ norm regularized CGBoost} \]

The primal problem is

\[ \min_w \sum_{i=1}^{m} \lambda(\gamma_i) + \frac{1}{2} \nu \|w\|_2^2, \text{ s.t. } \gamma_i = y_i H_i w, \ w \geq 0. \]

The Lagrange dual can be written into an unconstrained problem again:

\[ \min_u \sum_{i=1}^{m} \lambda^*(-u_i) + r \|d^+\|_2^2, \]

with \( r = 0.5/\nu. \)
**Summary: CGBoost**

1. Samples’ margins $\gamma$ and weak classifiers’ clipped edges $d^+$ are dual to each other.

2. $\ell_p$ regularization in primal corresponds to $\ell_q$ regularization in dual with $1/p + 1/q = 1$.

<table>
<thead>
<tr>
<th></th>
<th>Primal</th>
<th>Dual</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\ell_1$</td>
<td>$\min \sum_{i=1}^{m} \lambda(\gamma_i) + \nu |w|_1$</td>
<td>$\min \sum_{i=1}^{m} \lambda^*(-u_i) + r |d^+|_\infty$</td>
</tr>
<tr>
<td>$\ell_2$</td>
<td>$\min \sum_{i=1}^{m} \lambda(\gamma_i) + \nu |w|_2^2$</td>
<td>$\min \sum_{i=1}^{m} \lambda^*(-u_i) + r |d^+|_2^2$</td>
</tr>
<tr>
<td>$\ell_\infty$</td>
<td>$\min \sum_{i=1}^{m} \lambda(\gamma_i) + \nu |w|_\infty$</td>
<td>$\min \sum_{i=1}^{m} \lambda^*(-u_i) + r |d^+|_1$</td>
</tr>
<tr>
<td>$\lambda(\gamma)$: loss in primal</td>
<td>$|d^+|_q$: loss in dual</td>
<td></td>
</tr>
<tr>
<td>$|w|_p$: regularization in primal</td>
<td>$\lambda^*(u)$: regularization in dual</td>
<td></td>
</tr>
</tbody>
</table>
Results of the Wilcoxon Signed-Ranks Test (WSRT). The conclusion is that no statistical significance between various boosting methods.

<table>
<thead>
<tr>
<th></th>
<th>AdaBoost</th>
<th>CG exp, ( \ell_1 )</th>
<th>CG exp, ( \ell_2 )</th>
<th>CG exp, ( \ell_\infty )</th>
<th>CG hinge, ( \ell_1 )</th>
<th>CG hinge, ( \ell_2 )</th>
<th>CG hinge, ( \ell_\infty )</th>
</tr>
</thead>
<tbody>
<tr>
<td>AdaBoost</td>
<td>–</td>
<td>No (46 &lt; 70)</td>
<td>No (53 &lt; 61)</td>
<td>No (19 &lt; 70)</td>
<td>No (44 &lt; 70)</td>
<td>No (38 &lt; 70)</td>
<td>No (62 &lt; 70)</td>
</tr>
<tr>
<td>CG exp, ( \ell_1 )</td>
<td>No (45 &lt; 70)</td>
<td>–</td>
<td>No (54 &lt; 70)</td>
<td>No (24 &lt; 61)</td>
<td>No (44 &lt; 70)</td>
<td>No (32 &lt; 70)</td>
<td>No (60 &lt; 70)</td>
</tr>
<tr>
<td>CG exp, ( \ell_2 )</td>
<td>No (25 &lt; 61)</td>
<td>No (37 &lt; 70)</td>
<td>–</td>
<td>No (30 &lt; 70)</td>
<td>No (32 &lt; 70)</td>
<td>No (24 &lt; 70)</td>
<td>No (45 &lt; 70)</td>
</tr>
<tr>
<td>CG exp, ( \ell_\infty )</td>
<td>Better (72 &gt; 70)</td>
<td>No (54 &lt; 61)</td>
<td>No (61 &lt; 70)</td>
<td>–</td>
<td>No (61 &lt; 70)</td>
<td>No (47 &lt; 70)</td>
<td>Better (70 = 70)</td>
</tr>
<tr>
<td>CG hinge, ( \ell_1 )</td>
<td>No (47 &lt; 70)</td>
<td>No (47 &lt; 70)</td>
<td>No (59 &lt; 70)</td>
<td>No (30 &lt; 70)</td>
<td>–</td>
<td>No (11 &lt; 61)</td>
<td>No (64 &lt; 70)</td>
</tr>
<tr>
<td>CG hinge, ( \ell_2 )</td>
<td>No (53 &lt; 70)</td>
<td>No (59 &lt; 70)</td>
<td>No (67 &lt; 70)</td>
<td>No (44 &lt; 70)</td>
<td>Better (67 &gt; 61)</td>
<td>–</td>
<td>Better (82 &gt; 70)</td>
</tr>
<tr>
<td>CG hinge, ( \ell_\infty )</td>
<td>No (29 &lt; 70)</td>
<td>No (31 &lt; 70)</td>
<td>No (46 &lt; 70)</td>
<td>No (21 &lt; 70)</td>
<td>No (27 &lt; 70)</td>
<td>No (9 &lt; 70)</td>
<td>–</td>
</tr>
</tbody>
</table>
CGBoost

- We now have a method to design a boosting algorithm for minimizing any regularized risk in a totally corrective fashion.
- C. Shen et al., “Fully corrective boosting with arbitrary loss and regularization” Neural Networks.
Application 1: Optimally training a cascade classifier

Cascade classifiers (1) standard cascade (2) multi-exit cascade. Only those classified as true detection by all nodes will be true targets.

\[ h_1, h_2, \ldots, h_j, h_{j+1}, \ldots, h_{n-1}, h_n \]
The detection rate and false positive rate of a cascade are: 
\[ F_{dr} = \prod_{t=1}^{N} d_t \] and \[ F_{fp} = \prod_{t=1}^{N} f_t \], respectively; if we have \( N \) nodes.

**Node Learning Objective:** Each node should have an extremely high detection rate \( d_t \) (e.g., 99.7%) and a moderate false positive rate \( f_t \) (e.g., 50%). For example, \( N = 20 \), \( F_{dr} \approx 94\% \) and \( F_{fp} \approx 10^{-6} \).
New understanding of boosting methods

Boosting for achieving the Node Learning Objective

Minimax Probability Machines [Lanckriet et al. 2002 JMLR]:

$$\max_{w,b,\gamma} \gamma \quad \text{s.t.} \quad \left[ \inf_{x_1 \sim (\mu_1, \Sigma_1)} \Pr\{w^\top x_1 \geq b\} \right] \geq \gamma,$$

$$\left[ \inf_{x_2 \sim (\mu_2, \Sigma_2)} \Pr\{w^\top x_2 \leq b\} \right] \geq \gamma.$$
New understanding of boosting methods

Boosting for achieving the Node Learning Objective

Biased Minimax Probability Machines:

\[
\max_{w,b,\gamma} \gamma \quad \text{s.t.} \quad \left[ \inf_{x_1 \sim (\mu_1, \Sigma_1)} \Pr\{w^\top x_1 \geq b\} \right] \geq \gamma, \\
\left[ \inf_{x_2 \sim (\mu_2, \Sigma_2)} \Pr\{w^\top x_2 \leq b\} \right] \geq \gamma_0.
\]

Let’s consider a special case: \(\gamma_0 = 0.5\): The 2nd class will have a classification accuracy around 50%.
New understanding of boosting methods

Boosting from the simplified biased MPM

We can easily rewrite the original problem into:

$$\min_{w, \rho} \frac{1}{2} \rho^\top Q \rho - \theta e^\top \rho,$$

s.t. $w \geq 0, 1^\top w = 1,

$$\rho_i = (Aw)_i, i = 1, \ldots, m.$$ 

Here $Q = \begin{bmatrix} Q_1 & 0 \\ 0 & Q_2 \end{bmatrix}$ is a block matrix with

$$Q_1 = \begin{bmatrix}
\frac{1}{m} & -\frac{1}{m(m_1-1)} & \cdots & -\frac{1}{m(m_1-1)} \\
-\frac{1}{m(m_1-1)} & \frac{1}{m} & \cdots & -\frac{1}{m(m_1-1)} \\
\vdots & \vdots & \ddots & \vdots \\
-\frac{1}{m(m_1-1)} & -\frac{1}{m(m_1-1)} & \cdots & \frac{1}{m}
\end{bmatrix},$$

and $Q_2$ is similarly defined by replacing $m_1$ with $m_2$ in $Q_1$. 

The Lagrange dual is:

\[
\max_{u,r} - r - \frac{1}{2} (u - \theta e)^\top Q^{-1} (u - \theta e), \quad \text{s.t.} \quad \sum_{i=1}^{m} u_i A_i \leq r 1^\top.
\]

We then apply the column generation idea to obtain a new boosting algorithm.
At each iteration of Column Generation, solve the primal problem (QP subject to Simplex constraints) using entropic gradient (EG) descent.

Use KKT condition to recover the dual variable $u^*$.

EG takes 0.054 seconds to solve the primal QP with 1000 variables. Mosek takes 1.22 seconds: 20 times faster.

To train a complete cascade with 2900 weak classifiers, it takes 4 hours on a workstation with 8 Intel Xeon E5520 Cores and 32GB RAM (OpenMP used).
Object detection

AdaBoost and FisherBoost on toy data:
New understanding of boosting methods

Object detection (node performance)

![Graph showing false negative rate for different node numbers and different methods: multi-exit cascade (Ada), multi-exit cascade with LDA (Ada), multi-exit cascade with FisherBoost, multi-exit cascade with LAC (Ada), multi-exit cascade with LACBoost.](image-url)
New understanding of boosting methods

Object detection (node performance)

![Graph showing false negative rate for different node numbers and methods]

- multi-exit cascade (Asym)
- multi-exit cascade with LDA (Asym)
- multi-exit cascade with FisherBoost
- multi-exit cascade with LAC (Asym)
- multi-exit cascade with LACBoost
New understanding of boosting methods

Object detection (ROC)

Detection rate

Number of false positives

Viola–Jones cascade
multi–exit cascade (Ada)
multi–exit cascade with LDA (Ada)
multi–exit cascade with FisherBoost

Viola–Jones cascade
multi–exit cascade (Ada)
multi–exit cascade with LAC (Ada)
multi–exit cascade with LACBoost
New understanding of boosting methods

Object detection (ROC)

- multi−exit cascade (Asym)
- multi−exit cascade with LDA (Asym)
- multi−exit cascade with FisherBoost
- multi−exit cascade (Asym)
- multi−exit cascade with LAC (Asym)
- multi−exit cascade with LACBoost
New understanding of boosting methods

Object detection: examples

Figure: Detection results on the MIT+CMU test set.
New understanding of boosting methods

Human detection (ROC)

Figure: Comparison between FisherBoost, LACBoost, HOG with the linear SVM of Dalal and Triggs, and Pyramid HOG with the histogram intersection kernel SVM (IKSVM) of [Maji et al, CVPR 2008]. 15× faster than [Maji 2008], 70× faster than [Dalal & Triggs 2005], → real-time
The proposed method can be immediately applied to other asymmetric/rare event detection problems (spam detection, web data mining, network intrusion detection)

Examples

**Figure:** Detected pedestrians in the INRIA data set by our method.
A direct approach to multi-class boosting

The basic idea is to learn classifiers by pairwise comparison. For a training example \((x, y)\), if we have a perfect classification rule, then the following holds

\[ F_y(x) > F_r(x), \text{ for any } r \neq y. \]
A direct approach to multi-class boosting

In the large margin framework with the hinge loss:

\[ F_y(x) \geq 1 + F_r(x), \text{ for any } r \neq y. \]

This means that the correct label is supposed to have a classification confidence that is larger by at least a unit than any of the confidences for the other predictions.
A direct approach to multi-class boosting

We generalize this idea to the entire training set and introduce slack variables $\xi$ to enable soft-margin. The primal problem that we want to optimize can then be written as

$$\min_{W,\xi} \sum_{i=1}^{m} \xi_i + \nu ||W||_1$$

s.t. $\delta_{r,y_i} + H_i : w_{y_i} \geq 1 + H_i : w_r - \xi_i, \forall i, r,$

$W \geq 0.$

Here $\nu > 0$ is the regularization parameter.
A direct approach to multi-class boosting

Lagrange dual:

\[
\begin{align*}
\min_U & \quad \sum_{r=1}^{k} \sum_{i=1}^{m} \delta_{r,y_i} U_{ir} \\
\text{s.t.} \quad & \sum_{i} (\delta_{r,y_i} - U_{ir}) H_i \leq \nu 1^T, \forall r, \\
& \sum_{r} U_{ir} = 1, \forall i; \quad U \geq 0.
\end{align*}
\]

Each row of the matrix $U$ is normalized.
The first set of constraints can be infinitely many:

$$\sum_{i} (\delta_{r,y_i} - U_{ir}) h(x_i) \leq \nu, \forall r, \text{ and } \forall h(\cdot) \in \mathcal{H}.$$ 

How to boost?
Find the most violated constraint (best weak learner)!
The subproblem for generating weak classifiers is

\[ h^*(\cdot) = \arg\max_{h(\cdot), r} \sum_{i=1}^{m} (\delta_{r, y_i} - U_{ir}) h(x_i). \]

The matrix \( U \in \mathbb{R}^{m \times k} \) plays the role of measuring importance of a training example.
Toy example
Given training samples, our goal is to minimize the multi-class hinge loss with $\ell_{1,2}$ mixed-norm regularization. The primal problem can be written as

$$\min_{W,\xi} \sum_{i=1}^{m} \xi_i + \nu \|W\|_{1,2}$$

s.t. $\delta_{r,y_i} + H_{i:y_i}w_{y_i} \geq 1 + H_{i:w_r} - \xi_i, \forall i, r$;

$W \geq 0; \xi \geq 0.$
Results

<table>
<thead>
<tr>
<th></th>
<th>MNIST</th>
<th>‘0 − 3’</th>
<th>‘4 − 5’</th>
<th>‘6 − 7’</th>
<th>‘8 − 10’</th>
</tr>
</thead>
<tbody>
<tr>
<td>MultiBoost</td>
<td>99.8% 0.2% 0% 0%</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Group Sparsity</td>
<td>4.5% 48.8% 40.9% 5.8%</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ABCDETC</td>
<td>‘0 − 15’ ‘16 − 30’ ‘31 − 45’ ‘46 − 62’</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MultiBoost</td>
<td>99.8% 0.2% 0% 0%</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Group Sparsity</td>
<td>0% 81.3% 18.7% 0%</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table: The distribution of shared weak classifiers. For example, ‘8 − 10’ indicates that the weak classifier is being shared among 8 to 10 classes. The table illustrates the feature sharing property of our algorithms, i.e., one weak classifier is being shared among multiple classes.
A direct approach to multi-class boosting


We can also generalize this idea to broader structured learning.
Thanks

more information:

cs.adelaide.edu.au/~chhshen/

Questions?