Deep Boosting

Joint work with Corinna Cortes (Google Research)
Vitaly Kuznetsov (Courant Institute)
Umar Syed (Google Research)

MEHRYAR MOHRI MOHRI@
COURANT INSTITUTE & GOOGLE RESEARCH

Deep Boosting Essence

Ensemble Methods in ML

- Combining several base classifiers to create a more accurate one.
  - Bagging (Breiman 1996).
  - AdaBoost (Freund and Schapire 1997).
  - Stacking (Smyth and Wolpert 1999).
  - Bayesian averaging (MacKay 1996).
  - Other averaging schemes e.g., (Freund et al. 2004).
- Often very effective in practice.
- Benefit of favorable learning guarantees.

Convex Combinations

- Base classifier set H.
    - boosting stumps.
    - decision trees with limited depth or number of leaves.
- Ensemble combinations: convex hull of base classifier set.

$$\operatorname{conv}(H) = \left\{ \sum_{t=1}^{T} \alpha_t h_t \colon \alpha_t \geq 0; \sum_{t=1}^{T} \alpha_t \leq 1; \forall t, h_t \in H \right\}.$$

Ensembles - Margin Bound

(Koltchinskii and Panchencko, 2002)

- Theorem: let H be a family of real-valued functions. Fix $\rho > 0$. Then, for any $\delta > 0$, with probability at least $1-\delta$, the following holds for all $f = \sum_{t=1}^{T} \alpha_t h_t \in \text{conv}(H)$:

$$R(f) \leq \widehat{R}_{S,\rho}(f) + \frac{2}{\rho}\mathfrak{R}_m(H) + \sqrt{\frac{\log \frac{1}{\delta}}{2m}},$$

- where $\widehat{R}_{S,\rho}(f) = \frac{1}{m}\sum_{i=1}^{m} 1_{y_i f(x_i) \leq \rho}$.

Questions

- Can we use a much richer or deeper base classifier set?
  - richer families needed for difficult tasks.
  - but generalization bound indicates risk of overfitting.

AdaBoost

(Freund and Schapire, 1997)

- Description: coordinate descent applied to

$$F(\boldsymbol{\alpha}) = \sum_{i=1}^{m} e^{-y_i f(x_i)} = \sum_{i=1}^{m} \exp\left(-y_i \sum_{t=1}^{T} \alpha_t h_t(x_i)\right).$$

- Guarantees: ensemble margin bound.
  - but AdaBoost does not maximize the margin!
  - some margin maximizing algorithms such as arc-gv are outperformed by AdaBoost! (Reyzin and Schapire, 2006)

Suspicions

- Complexity of hypotheses used:
  - arc-gv tends to use deeper decision trees to achieve a larger margin.
- Notion of margin:
  - minimal margin perhaps not the appropriate notion.
  - margin distribution is key.

→ can we shed more light on these questions?

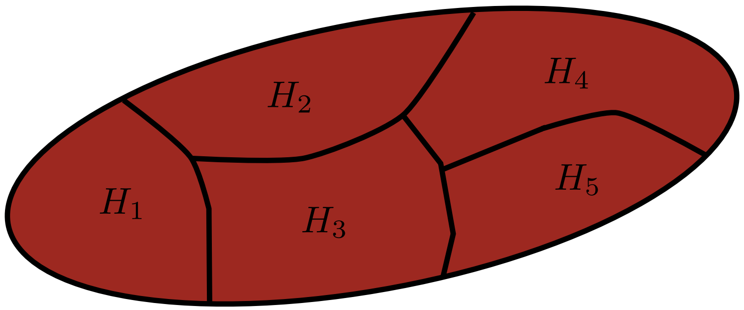
Question

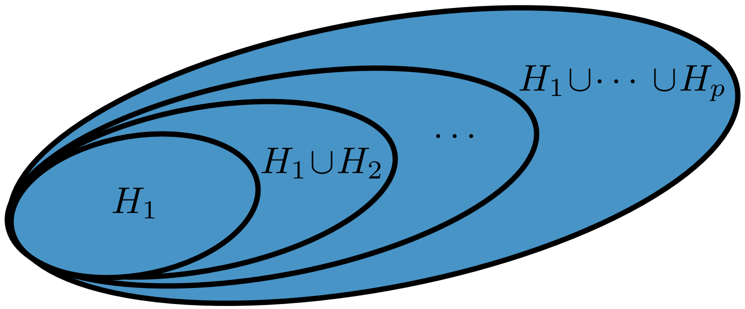
- Main question: how can we design ensemble algorithms that can succeed even with very deep decision trees or other complex sets?
    - theory.
    - algorithms.
    - experimental results.
    - model selection.

Theory

Base Classifier Set H

- Decomposition in terms of sub-families or their union.



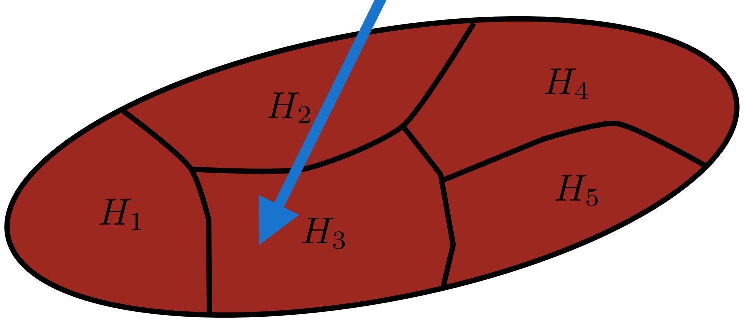


Ensemble Family

- Non-negative linear ensembles $\mathcal{F} = \mathrm{conv}(\cup_{k=1}^{p} H_k)$:

$$f = \sum_{t=1}^{T} \alpha_t h_t$$

with $\alpha_t \geq 0, \sum_{t=1}^{T} \alpha_t \leq 1, h_t \in H_{k_t}.$



Ideas

- Use hypotheses drawn from H_ks with larger ks but allocate more weight to hypotheses drawn from smaller ks.
  - how can we determine quantitatively the amounts of mixture weights apportioned to different families?
  - can we provide learning guarantees guiding these choices?

Learning Guarantee

(Cortes, MM, and Syed, 2014)

- Theorem: Fix $\rho > 0$. Then, for any $\delta > 0$, with probability at least $1 - \delta$, the following holds for all $f = \sum_{t=1}^{T} \alpha_t h_t \in \mathcal{F}$:

$$R(f) \leq \widehat{R}_{S,\rho}(f) + \frac{4}{\rho} \sum_{t=1}^{T} \alpha_t \mathfrak{R}_m(H_{k_t}) + \widetilde{O}\left(\sqrt{\frac{\log p}{\rho^2 m}}\right).$$

Consequences

- Complexity term with explicit dependency on mixture weights.
  - quantitative guide for controlling weights assigned to more complex sub-families.
  - bound can be used to inspire, or directly define an ensemble algorithm.

Algorithms

Set-Up

- $H_1, \ldots, H_p$: disjoint sub-families of functions taking values in $[-1, +1]$.
- Further assumption (not necessary): symmetric sub-families, i.e. $h \in H_k \Leftrightarrow -h \in H_k$.
- Notation:
  - $r_j = \mathfrak{R}_m(H_{k_j})$ with $h_j \in H_{k_j}$.

Derivation

- Learning bound suggests seeking $\boldsymbol{\alpha} \geq 0$ with $\sum_{t=1}^{T} \alpha_t \leq 1$ to minimize

$$\frac{1}{m} \sum_{i=1}^{m} 1_{y_i \sum_{t=1}^{T} \alpha_t h_t(x_i) \leq \rho} + \frac{4}{\rho} \sum_{t=1}^{T} \alpha_t r_t.$$

Convex Surrogates

- Let $u \mapsto \Phi(-u)$ be a decreasing convex function upper bounding $u \mapsto 1_{u \leq 0}$, with Φ differentiable.
- Two principal choices:
  - Exponential loss: $\Phi(-u) = \exp(-u)$.
  - Logistic loss: $\Phi(-u) = \log_2(1 + \exp(-u))$.

Optimization Problem

(Cortes, MM, and Syed, 2014)

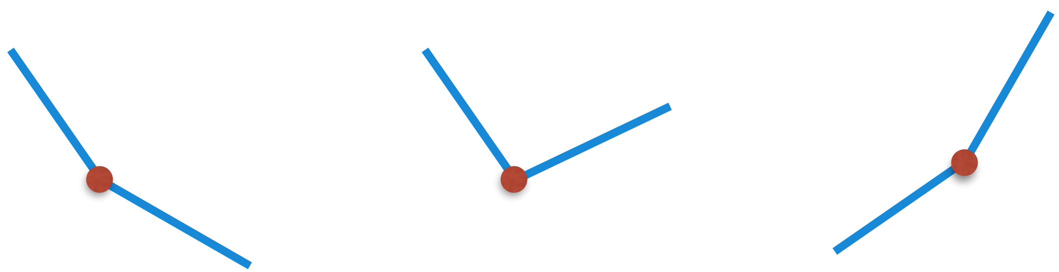
- Moving the constraint to the objective and using the fact that the sub-families are symmetric leads to:

$$\min_{\boldsymbol{\alpha} \in \mathbb{R}^N} \frac{1}{m} \sum_{i=1}^{m} \Phi\Big(1 - y_i \sum_{j=1}^{N} \alpha_j h_j(x_i)\Big) + \sum_{t=1}^{N} (\lambda r_j + \beta) |\alpha_j|,$$

where $\lambda, \beta \geq 0$, and for each hypothesis, keep either *h* or *-h*.

DeepBoost Algorithm

- Coordinate descent applied to convex objective.
  - non-differentiable function.
  - definition of maximum coordinate descent.



Direction & Step

- Maximum direction: definition based on the error

$$\epsilon_{t,j} = \frac{1}{2}\Big[1 - \mathop{\mathrm{E}}_{i \sim \mathcal{D}_t}[y_i h_j(x_i)]\Big],$$

where D_t is the distribution over sample at iteration t.

- Step:
  - closed-form expressions for exponential and logistic losses.
  - general case: line search.

Pseudocode

DeepBoost$(S = ((x_1, y_1), \ldots, (x_m, y_m)))$

for $i \leftarrow 1$ **to** m **do**
  $D_1(i) \leftarrow \frac{1}{m}$
for $t \leftarrow 1$ **to** T **do**
  for $j \leftarrow 1$ **to** N **do**
    if $(\alpha_{t-1,j} \neq 0)$ **then**
      $d_j \leftarrow \left(\epsilon_{t,j} - \frac{1}{2}\right) + \operatorname{sgn}(\alpha_{t-1,j}) \frac{\Lambda_j m}{2S_t}$
    elseif $\left(\left|\epsilon_{t,j} - \frac{1}{2}\right| \leq \frac{\Lambda_j m}{2S_t}\right)$ **then**
      $d_j \leftarrow 0$
    else $d_j \leftarrow \left(\epsilon_{t,j} - \frac{1}{2}\right) - \operatorname{sgn}\left(\epsilon_{t,j} - \frac{1}{2}\right) \frac{\Lambda_j m}{2S_t}$
  $k \leftarrow \underset{j \in [1,N]}{\operatorname{argmax}} |d_j|$
  $\epsilon_t \leftarrow \epsilon_{t,k}$
  if $\left(\left|(1 - \epsilon_t) e^{\alpha_{t-1,k}} - \epsilon_t e^{-\alpha_{t-1,k}}\right| \leq \frac{\Lambda_k m}{S_t}\right)$ **then**
    $\eta_t \leftarrow -\alpha_{t-1,k}$
  elseif $\left((1 - \epsilon_t) e^{\alpha_{t-1,k}} - \epsilon_t e^{-\alpha_{t-1,k}} > \frac{\Lambda_k m}{S_t}\right)$ **then**
    $\eta_t \leftarrow \log\left[-\frac{\Lambda_k m}{2\epsilon_t S_t} + \sqrt{\left[\frac{\Lambda_k m}{2\epsilon_t S_t}\right]^2 + \frac{1-\epsilon_t}{\epsilon_t}}\right]$
  else $\eta_t \leftarrow \log\left[+\frac{\Lambda_k m}{2\epsilon_t S_t} + \sqrt{\left[\frac{\Lambda_k m}{2\epsilon_t S_t}\right]^2 + \frac{1-\epsilon_t}{\epsilon_t}}\right]$
  $\boldsymbol{\alpha}_t \leftarrow \boldsymbol{\alpha}_{t-1} + \eta_t \mathbf{e}_k$
  $S_{t+1} \leftarrow \sum_{i=1}^{m} \Phi'\left(1 - y_i \sum_{j=1}^{N} \alpha_{t,j} h_j(x_i)\right)$
  for $i \leftarrow 1$ **to** m **do**
    $D_{t+1}(i) \leftarrow \frac{\Phi'\left(1 - y_i \sum_{j=1}^{N} \alpha_{t,j} h_j(x_i)\right)}{S_{t+1}}$
$f \leftarrow \sum_{j=1}^{N} \alpha_{t,j} h_j$
return f

$$\Lambda_j = \lambda r_j + \beta.$$

Connections with Previous Work

- For $\lambda = \beta = 0$, DeepBoost coincides with
  - AdaBoost (Freund and Schapire 1997), run with union of sub-families, for the exponential loss.
  - additive Logistic Regression (Friedman et al., 1998), run with union of sub-families, for the logistic loss.
- For $\lambda = 0$ and $\beta \neq 0$, DeepBoost coincides with
  - L1-regularized AdaBoost (Raetsch, Mika, and Warmuth 2001), for the exponential loss.
  - L1-regularized Logistic Regression (Duchi and Singer 2009), for the logistic loss.

Experiments

Rad. Complexity Estimates

- Benefit of data-dependent analysis:
  - empirical estimates of each $\mathfrak{R}_m(H_k)$.
  - example: for kernel function K_k,

$$\widehat{\mathfrak{R}}_S(H_k) \leq \frac{\sqrt{\mathrm{Tr}[\mathbf{K}_k]}}{m}.$$

  - alternatively, upper bounds in terms of growth functions,

$$\mathfrak{R}_m(H_k) \leq \sqrt{\frac{2 \log \Pi_{H_k}(m)}{m}}.$$

Experiments (1)

- Family of base classifiers defined by boosting stumps:
  - boosting stumps H_1^{stumps} (threshold functions).
    - in dimension d, $\Pi_{H_1^{\text{stumps}}}(m) \leq 2md$, thus

$$\mathfrak{R}_m(H_1^{\text{stumps}}) \leq \sqrt{\frac{2\log(2md)}{m}}.$$

  - decision trees of depth 2, H_2^{stumps}, with the same question at the internal nodes of depth 1.
    - in dimension d, $\Pi_{H_2^{\text{stumps}}}(m) \leq (2m)^2\,\frac{d(d-1)}{2}$, thus

$$\mathfrak{R}_m(H_2^{\text{stumps}}) \leq \sqrt{\frac{2\log(2m^2 d(d-1))}{m}}.$$

Experiments (1)

- Base classifier set: $H_1^{\text{stumps}} \cup H_2^{\text{stumps}}$.
- Data sets:
  - same UCI Irvine data sets as (Breiman 1999) and (Reyzin and Schapire 2006).
  - OCR data sets used by (Reyzin and Schapire 2006): ocr17, ocr49.
  - MNIST data sets: ocr17-mnist, ocr49-mnist.
- Experiments with exponential loss.
- Comparison with AdaBoost and AdaBoost-L1.

Experiments - Stumps Exp Loss

(Cortes, MM, and Syed, 2014)

Table 1: Results for boosted decision stumps and the exponential loss function.

breastcancer	AdaBoost H_1^{stumps}	AdaBoost H_2^{stumps}	AdaBoost-L1	DeepBoost
Error	0.0429	0.0437	0.0408	**0.0373**
(std dev)	(0.0248)	(0.0214)	(0.0223)	**(0.0225)**
Avg tree size	1	2	1.436	1.215
Avg no. of trees	100	100	43.6	21.6

ocr17	AdaBoost H_1^{stumps}	AdaBoost H_2^{stumps}	AdaBoost-L1	DeepBoost
Error	0.0085	0.008	0.0075	**0.0070**
(std dev)	0.0072	0.0054	0.0068	**(0.0048)**
Avg tree size	1	2	1.086	1.369
Avg no. of trees	100	100	37.8	36.9

ionosphere	AdaBoost H_1^{stumps}	AdaBoost H_2^{stumps}	AdaBoost-L1	DeepBoost
Error	0.1014	0.075	0.0708	**0.0638**
(std dev)	(0.0414)	(0.0413)	(0.0331)	**(0.0394)**
Avg tree size	1	2	1.392	1.168
Avg no. of trees	100	100	39.35	17.45

ocr49	AdaBoost H_1^{stumps}	AdaBoost H_2^{stumps}	AdaBoost-L1	DeepBoost
Error	0.0555	0.032	0.03	**0.0275**
(std dev)	0.0167	0.0114	0.0122	**(0.0095)**
Avg tree size	1	2	1.99	1.96
Avg no. of trees	100	100	99.3	96

german	AdaBoost H_1^{stumps}	AdaBoost H_2^{stumps}	AdaBoost-L1	DeepBoost
Error	0.243	0.2505	0.2455	**0.2395**
(std dev)	(0.0445)	(0.0487)	(0.0438)	**(0.0462)**
Avg tree size	1	2	1.54	1.76
Avg no. of trees	100	100	54.1	76.5

ocr17-mnist	AdaBoost H_1^{stumps}	AdaBoost H_2^{stumps}	AdaBoost-L1	DeepBoost
Error	0.0056	0.0048	0.0046	**0.0040**
(std dev)	0.0017	0.0014	0.0013	**(0.0014)**
Avg tree size	1	2	2	1.99
Avg no. of trees	100	100	100	100

diabetes	AdaBoost H_1^{stumps}	AdaBoost H_2^{stumps}	AdaBoost-L1	DeepBoost
Error	0.253	0.260	0.254	**0.253**
(std dev)	(0.0330)	(0.0518)	(0.04868)	**(0.0510)**
Avg tree size	1	2	1.9975	1.9975
Avg no. of trees	100	100	100	100

ocr49-mnist	AdaBoost H_1^{stumps}	AdaBoost H_2^{stumps}	AdaBoost-L1	DeepBoost
Error	0.0414	0.0209	0.0200	**0.0177**
(std dev)	0.00539	0.00521	0.00408	**(0.00438)**
Avg tree size	1	2	1.9975	1.9975
Avg no. of trees	100	100	100	100

Experiments (2)

- Family of base classifiers defined by decision trees of depth k. For trees with at most n nodes:

$$\mathfrak{R}_m(\mathsf{T}_n) \leq \sqrt{\frac{(4n+2)\log_2(d+2)\log(m+1)}{m}}.$$

- Base classifier set: $\cup_{k=1}^{K} H_k^{\text{trees}}$.
- Same data sets as with Experiments (1).
- Both exponential and logistic loss.
- Comparison with AdaBoost and AdaBoost-L1, Logistic Regression and L1-Logistic Regression.

Experiments - Trees Exp Loss

(Cortes, MM, and Syed, 2014)

breastcancer	AdaBoost	AdaBoost-L1	DeepBoost
Error	0.0267	0.0264	**0.0243**
(std dev)	(0.00841)	(0.0098)	**(0.00797)**
Avg tree size	29.1	28.9	20.9
Avg no. of trees	67.1	51.7	55.9

ocr17	AdaBoost	AdaBoost-L1	DeepBoost
Error	0.004	0.003	**0.002**
(std dev)	(0.00316)	(0.00100)	**(0.00100)**
Avg tree size	15.0	30.4	26.0
Avg no. of trees	88.3	65.3	61.8

ionosphere	AdaBoost	AdaBoost-L1	DeepBoost
Error	0.0661	0.0657	**0.0501**
(std dev)	(0.0315)	(0.0257)	**(0.0316)**
Avg tree size	29.8	31.4	26.1
Avg no. of trees	75.0	69.4	50.0

ocr49	AdaBoost	AdaBoost-L1	DeepBoost
Error	0.0180	**0.0175**	**0.0175**
(std dev)	(0.00555)	(0.00357)	**(0.00510)**
Avg tree size	30.9	62.1	30.2
Avg no. of trees	92.4	89.0	83.0

german	AdaBoost	AdaBoost-L1	DeepBoost
Error	0.239	0.239	**0.234**
(std dev)	(0.0165)	(0.0201)	**(0.0148)**
Avg tree size	3	7	16.0
Avg no. of trees	91.3	87.5	14.1

ocr17-mnist	AdaBoost	AdaBoost-L1	DeepBoost
Error	0.00471	0.00471	**0.00409**
(std dev)	(0.0022)	(0.0021)	**(0.0021)**
Avg tree size	15	33.4	22.1
Avg no. of trees	88.7	66.8	59.2

diabetes	AdaBoost	AdaBoost-L1	DeepBoost
Error	0.249	0.240	**0.230**
(std dev)	(0.0272)	(0.0313)	**(0.0399)**
Avg tree size	3	3	5.37
Avg no. of trees	45.2	28.0	19.0

ocr49-mnist	AdaBoost	AdaBoost-L1	DeepBoost
Error	0.0198	0.0197	**0.0182**
(std dev)	(0.00500)	(0.00512)	**(0.00551)**
Avg tree size	29.9	66.3	30.1
Avg no. of trees	82.4	81.1	80.9

Experiments - Trees Log Loss

(Cortes, MM, and Syed, 2014)

breastcancer	LogReg	LogReg-L1	DeepBoost
Error	0.0351	0.0264	**0.0264**
(std dev)	(0.0101)	(0.0120)	**(0.00876)**
Avg tree size	15	59.9	14.0
Avg no. of trees	65.3	16.0	23.8

ocr17	LogReg	LogReg-L1	DeepBoost
Error	0.00300	0.00400	**0.00250**
(std dev)	(0.00100)	(0.00141)	**(0.000866)**
Avg tree size	15.0	7	22.1
Avg no. of trees	75.3	53.8	25.8

ionosphere	LogReg	LogReg-L1	DeepBoost
Error	0.074	0.060	**0.043**
(std dev)	(0.0236)	(0.0219)	**(0.0188)**
Avg tree size	7	30.0	18.4
Avg no. of trees	44.7	25.3	29.5

ocr49	LogReg	LogReg-L1	DeepBoost
Error	0.0205	0.0200	**0.0170**
(std dev)	(0.00654)	(0.00245)	**(0.00361)**
Avg tree size	31.0	31.0	63.2
Avg no. of trees	63.5	54.0	37.0

german	LogReg	LogReg-L1	DeepBoost
Error	0.233	0.232	**0.225**
(std dev)	(0.0114)	(0.0123)	**(0.0103)**
Avg tree size	7	7	14.4
Avg no. of trees	72.8	66.8	67.8

ocr17-mnist	LogReg	LogReg-L1	DeepBoost
Error	0.00422	0.00417	**0.00399**
(std dev)	(0.00191)	(0.00188)	**(0.00211)**
Avg tree size	15	15	25.9
Avg no. of trees	71.4	55.6	27.6

diabetes	LogReg	LogReg-L1	DeepBoost
Error	0.250	**0.246**	**0.246**
(std dev)	(0.0374)	**(0.0356)**	**(0.0356)**
Avg tree size	3	3	3
Avg no. of trees	46.0	45.5	45.5

ocr49-mnist	LogReg	LogReg-L1	DeepBoost
Error	0.0211	**0.0201**	**0.0201**
(std dev)	(0.00412)	**(0.00433)**	**(0.00411)**
Avg tree size	28.7	33.5	72.8
Avg no. of trees	79.3	61.7	41.9

Multi-Class Learning Guarantee

(Kuznetsov, MM, and Syed, 2014)

- Theorem: Fix $\rho > 0$. Then, for any $\delta > 0$, with probability at least $1 - \delta$, the following holds for all $f = \sum_{t=1}^{T} \alpha_t h_t \in \mathcal{F}$:

$$R(f) \leq \widehat{R}_{S,\rho}(f) + \frac{8c}{\rho} \sum_{t=1}^{T} \alpha_t \mathfrak{R}_m(\Pi_1(H_{k_t})) + \widetilde{O}\left(\sqrt{\frac{\log p}{\rho^2 m}}\right).$$

  - with c number of classes.
  - and $\Pi_1(H_k) = \{x \mapsto h(x, y) \colon y \in \mathcal{Y}, h \in H_k\}$.

Extension to Multi-Class

- Similar data-dependent learning guarantee proven for the multi-class setting.
  - bound depending on mixture weights and complexity of sub-families.
- Deep Boosting algorithm for multi-class:
  - similar extension taking into account the complexities of sub-families.
  - several variants depending on number of classes.
  - different possible loss functions for each variant.

Experiments - Multi-Class

Table 1: Empirical results for MDeepBoostSum, $\Phi = \exp$. AB stands for AdaBoost.

abalone	AB.MR	AB.MR-L1	MDeepBoost
Error	0.713	0.696	**0.677**
(std dev)	(0.0130)	(0.0132)	(0.0092)
Avg tree size	69.8	31.5	23.8
Avg no. of trees	17.9	13.3	15.3

handwritten	AB.MR	AB.MR-L1	MDeepBoost
Error	0.016	0.011	**0.009**
(std dev)	(0.0047)	(0.0026)	(0.0012)
Avg tree size	187.3	240.6	203.0
Avg no. of trees	34.2	21.7	24.2

letters	AB.MR	AB.MR-L1	MDeepBoost
Error	0.042	0.036	**0.032**
(std dev)	(0.0023)	(0.0018)	(0.0016)
Avg tree size	1942.6	1903.8	1914.6
Avg no. of trees	24.2	24.4	23.3

pageblocks	AB.MR	AB.MR-L1	MDeepBoost
Error	0.020	0.017	**0.013**
(std dev)	(0.0037)	(0.0021)	(0.0027)
Avg tree size	134.8	118.3	124.9
Avg no. of trees	8.5	14.3	6.6

pendigits	AB.MR	AB.MR-L1	MDeepBoost
Error	0.008	0.006	**0.004**
(std dev)	(0.0015)	(0.0023)	(0.0011)
Avg tree size	272.5	283.3	259.2
Avg no. of trees	23.2	19.8	21.4

satimage	AB.MR	AB.MR-L1	MDeepBoost
Error	0.089	0.081	**0.073**
(std dev)	(0.0062)	(0.0040)	(0.0045)
Avg tree size	557.9	478.8	535.6
Avg no. of trees	7.6	7.3	7.6

statlog	AB.MR	AB.MR-L1	MDeepBoost
Error	0.011	0.006	**0.004**
(std dev)	(0.0059)	(0.0035)	(0.0030)
Avg tree size	74.8	79.2	61.8
Avg no. of trees	23.2	17.5	17.6

yeast	AB.MR	AB.MR-L1	MDeepBoost
Error	0.388	0.376	**0.352**
(std dev)	(0.0392)	(0.0431)	(0.0402)
Avg tree size	100.6	111.7	71.4
Avg no. of trees	8.7	6.5	7.7

Experiments - Multi-Class

Table 1: Empirical results for MDeepBoostCompSum, comparison with multinomial logistic regression.

abalone	LogReg	LogReg-L1	MDeepBoost
Error	0.710	0.700	**0.687**
(std dev)	(0.0170)	(0.0102)	(0.0104)
Avg tree size	162.1	156.5	28.0
Avg no. of trees	22.2	9.8	10.2

handwritten	LogReg	LogReg-L1	MDeepBoost
Error	0.016	0.012	**0.008**
(std dev)	(0.0031)	(0.0020)	(0.0024)
Avg tree size	237.7	186.5	153.8
Avg no. of trees	32.3	32.8	35.9

letters	LogReg	LogReg-L1	MDeepBoost
Error	0.043	0.038	**0.035**
(std dev)	(0.0018)	(0.0012)	(0.0012)
Avg tree size	1986.5	1759.5	1807.3
Avg no. of trees	25.5	29.0	27.2

pageblocks	LogReg	LogReg-L1	MDeepBoost
Error	0.019	0.016	**0.012**
(std dev)	(0.0035)	(0.0025)	(0.0022)
Avg tree size	127.4	151.7	147.9
Avg no. of trees	4.5	6.8	7.4

pendigits	LogReg	LogReg-L1	MDeepBoost
Error	0.009	0.007	**0.005**
(std dev)	(0.0021)	(0.0014)	(0.0012)
Avg tree size	306.3	277.1	262.7
Avg no. of trees	21.9	20.8	19.7

satimage	LogReg	LogReg-L1	MDeepBoost
Error	0.091	0.082	**0.074**
(std dev)	(0.0066)	(0.0057)	(0.0056)
Avg tree size	412.6	454.6	439.6
Avg no. of trees	6.0	5.8	5.8

statlog	LogReg	LogReg-L1	MDeepBoost
Error	0.012	0.006	**0.002**
(std dev)	(0.0054)	(0.0020)	(0.0022)
Avg tree size	74.3	71.6	65.4
Avg no. of trees	22.3	20.6	17.5

yeast	LogReg	LogReg-L1	MDeepBoost
Error	0.381	0.375	**0.354**
(std dev)	(0.0467)	(0.0458)	(0.0468)
Avg tree size	103.9	83.3	117.2
Avg no. of trees	14.1	9.3	9.3

Other Related Algorithms

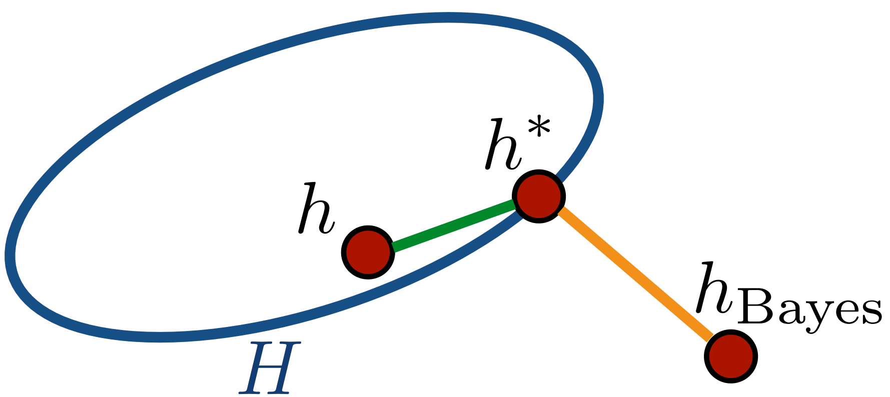
- Structural Maxent models (Cortes, Kuznetsov, MM, and Syed, ICML 2015): feature functions chosen from a union of very complex families.
- Deep Cascades (DeSalvo, MM, and Syed, ALT 2015): cascade of predictors with leaf predictors and node questions selected from very rich families.

Model Selection

Model Selection

- Problem: how to select hypothesis set H?
  - H too complex, no gen. bound, overfitting.
  - H too simple, gen. bound, but underfitting.

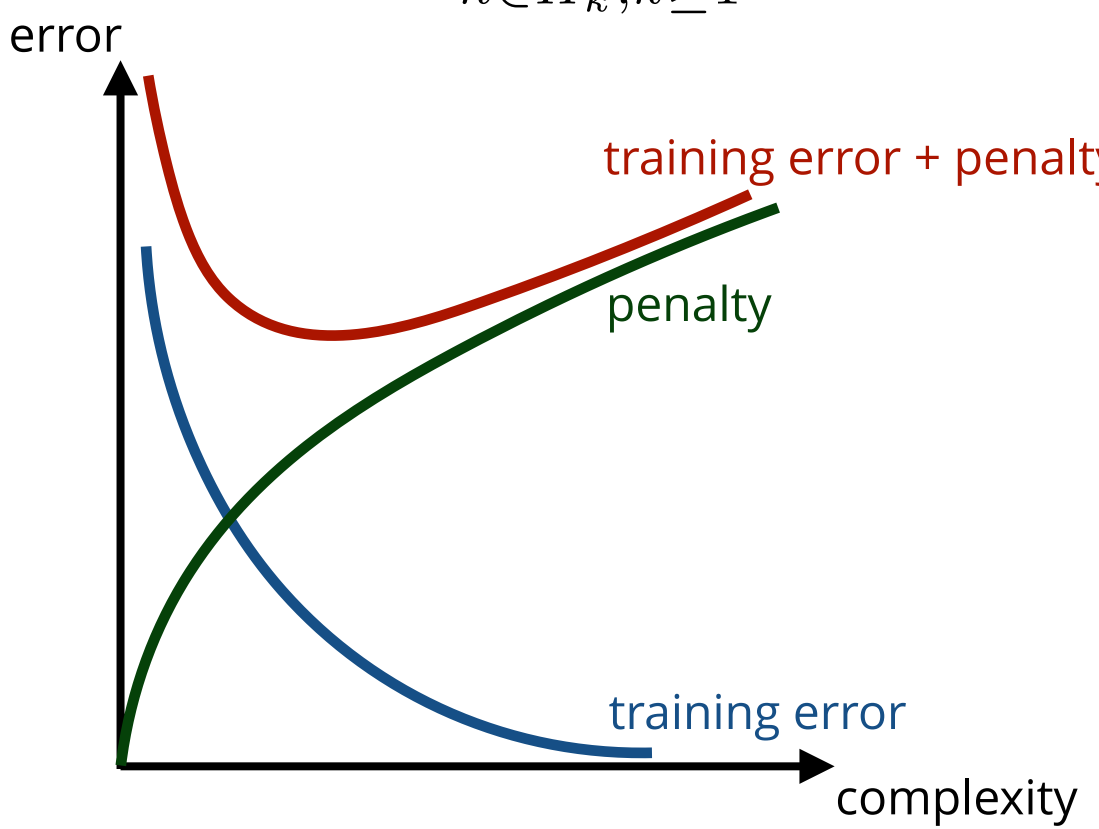
  → balance between estimation and approx. errors.



Structural Risk Minimization

(Vapnik and Chervonenkis, 1974; Vapnik, 1995)

- SRM: $H = \bigcup_{k=1}^{\infty} H_k$ with $H_1 \subset H_2 \subset \cdots \subset H_k \subset \cdots$
  - solution: $f^* = \operatorname*{argmin}_{h \in H_k, k \geq 1} \widehat{R}_S(h) + \mathrm{pen}(k, m)$.



Voted Risk Minimization

- Ideas:
  - no selection of specific H_k.
  - instead, use all H_ks: $h = \sum_{k=1}^{p} \alpha_k h_k$, $h_k \in H_k$, $\boldsymbol{\alpha} \in \Delta$.
  - hypothesis-dependent penalty:

$$\sum_{k=1}^{p} \alpha_k \mathfrak{R}_m(H_k).$$

→ Deep ensembles.

Conclusion

- Deep Boosting: ensemble learning with increasingly complex families.
  - data-dependent theoretical analysis.
  - algorithm based on learning bound.
  - extension to multi-class.
  - ranking and other losses.
  - enhancement of many existing algorithms.
  - compares favorably to AdaBoost and Logistic Regression or their L1-regularized variants in experiments.