

# Boosting on the Convex Hull

**Yongdai Kim**

*Department of Statistics*

*Ewha Womans University, Korea*

# 1. Introduction

## Objective

- Boosting (Freund and Schapire, 1997) is one of the most significant inventions in supervised learning.
- The basic idea of boosting makes a highly accurate learner by combining many base learners (weak learners).
- However, boosting has some critical deficiencies such as
  - vulnerability to output noise
  - overfitting when combining too many base learners.
- In this talk, I propose a new regularized boosting algorithm which overcomes such deficiencies.

## Main idea

- Mason et al. (2000) and Friedman (2001) showed that boosting can be understood as a gradient descent method in a function space.
- This implies that the final model of boosting is a linear combination of base learners.
- The complexity of a linear combination of base learners keeps increasing as more base learners are combined, which results in overfitting in boosting.
- The main idea of the proposed algorithm is to construct a convex combination of base learners instead of a linear combination.

## Contents of the talk

- Propose an algorithm for constructing the optimal convex combination of base learners
- Develop a PAC-style bound to explain how the proposed algorithm overcomes the aforementioned deficiencies in boosting
- Implement the proposed algorithm with decision trees
- Perform empirical comparison of the proposed algorithm and boosting.

## Literature Reviews

- Lee et al. (1996) developed an algorithm to find the optimal convex combination of base learners with  $L_2$  loss. But, this algorithm is hardly extended to other than  $L_2$  loss function.
- Mason et al. (2000) proposed a gradient descent method in the convex hull of base classifiers (called  $L_1$  boosting). However, this algorithm does not work well for regression problems.
- Kim et al. (2002) and Kim (2003) proposed two heuristic regularized boosting algorithms on the convex hull of base learners, but theoretical backgrounds are weak.
- The proposed algorithm works well both classification and regression problems with any convex loss functions (satisfying some regularity condition) and has well established theoretical reasonings.

## 2. Basic set-up of Statistical learning

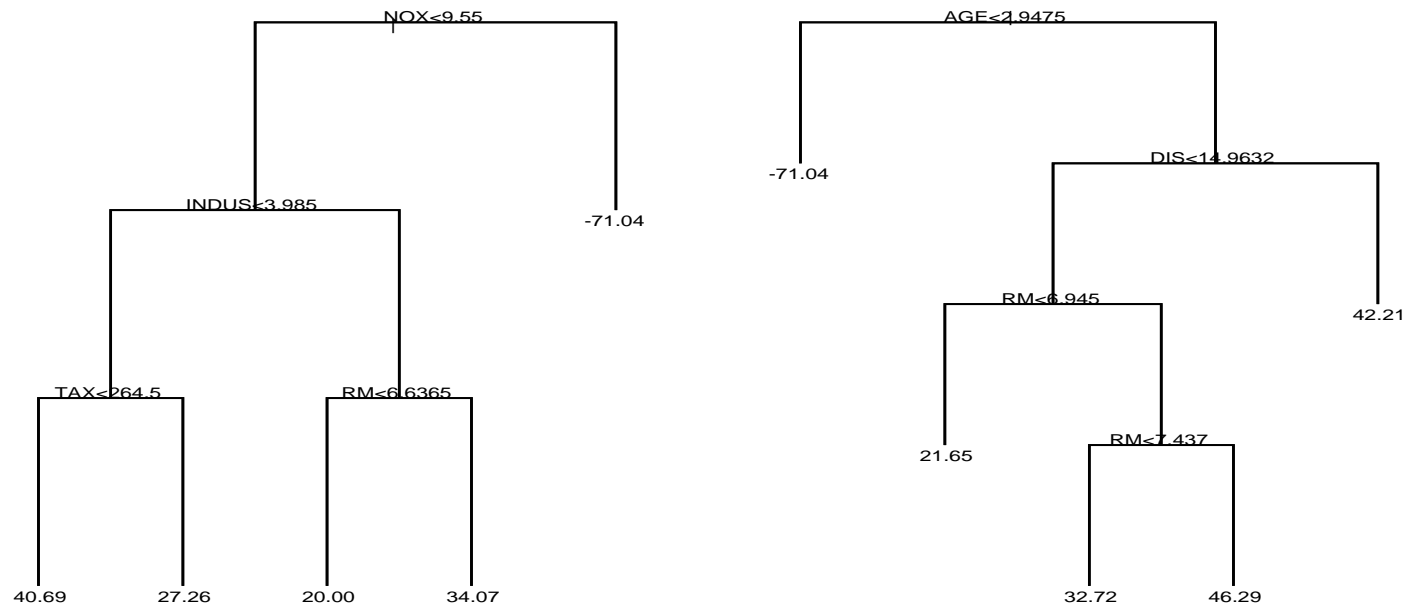
- Input(Covariate) :  $\mathbf{x} \in R^p$
- Output(Response) :  $y$
- System (Model):  $y = f(\mathbf{x}, \epsilon)$
- Loss function:  $l(y, a)$
- Assumption :  $f$  belongs to a family of functions  $\mathcal{F}$ .
- Learning set (Data):  $\mathcal{L} = \{(y_i, \mathbf{x}_i, i = 1, \dots, n\}$  assumed to be a random sample of  $(Y, \mathbf{X}) \sim P$
- Objective: Find  $f^0 = \arg \min_{f \in \mathcal{F}} E_{(Y, \mathbf{X})} l(Y, f(\mathbf{X}))$ .
- Estimation: Estimate  $f^0$  by  $\hat{f} = \arg \min_{f \in \mathcal{F}} \sum_{i=1}^n l(y_i, f(\mathbf{x}_i))$ .
- Prediction: If new input is  $\mathbf{x}$ , predict unknown  $y$  by  $\hat{f}(\mathbf{x})$ .

$y$  is categorical  $\Rightarrow$  Classification  
is continuous  $\Rightarrow$  Regression

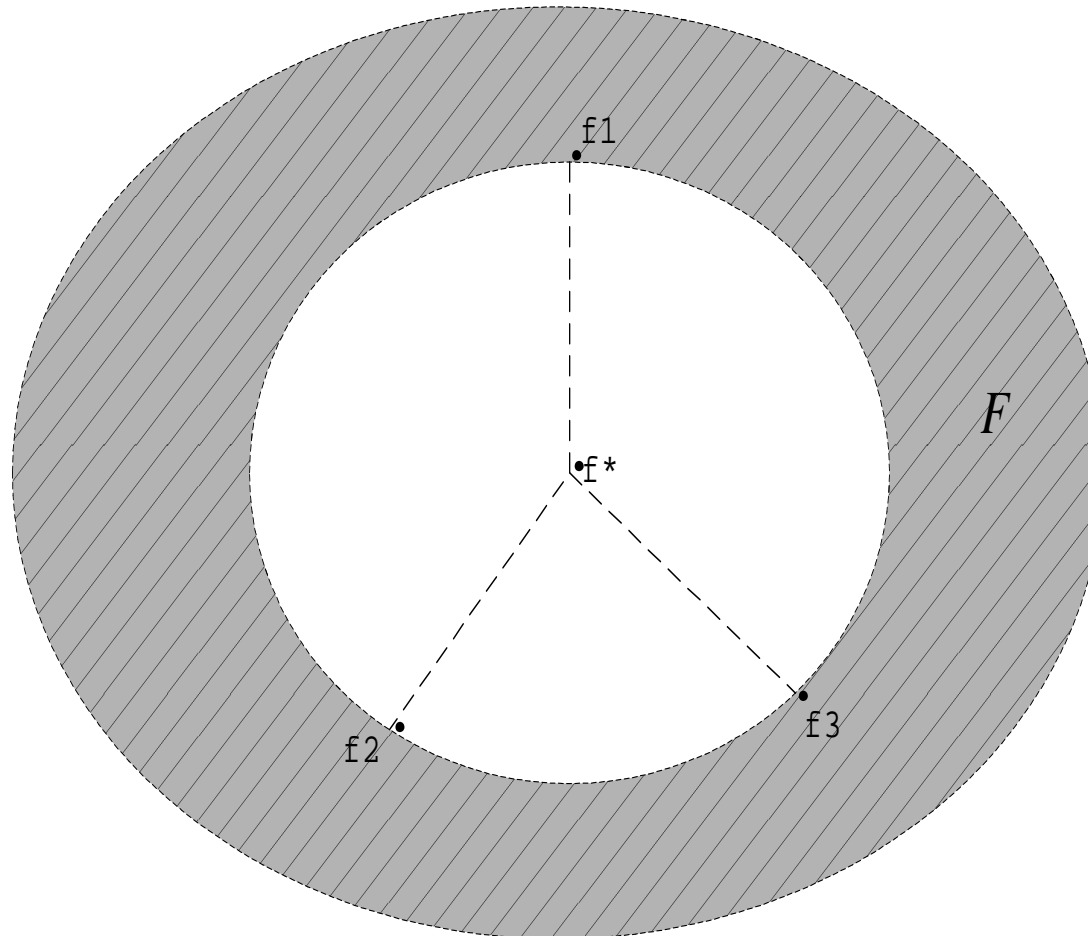
### 3. Instability in decision trees

#### Example of instability

- Two trees from two bootstrap samples of the Boston Housing data with node size 5



# Geometry



## Idea of boosting

- Instead of finding the best model on  $\mathcal{F}$ , find the best model on the convex hull of  $\mathcal{F}$ .
- That is, boosting extends the function space from  $\mathcal{F}$  to the convex hull of  $\mathcal{F}$ .
- Similar arguments can be applied to other ensemble methods such as bagging and random forest.
- Now, the questions are
  - how to find the best model on the convex hull of  $\mathcal{F}$ ?
  - how to control the complexity of the model (i.e. the size of the convex hull)?

## 4. Algorithm: Convex Hull Boost (CHB)

- Let  $\mathcal{F}$  be the set of base learners, a subset of a certain Hilbert space.
- Let  $co(\mathcal{F})$  be the convex hull of  $\mathcal{F}$ , that is

$$co(\mathcal{F}) = \left\{ \sum_{i=1}^M w_i f_i(\mathbf{x}), w_i \geq 0, \sum_{i=1}^M w_i = 1, M = 1, 2, \dots \right\}.$$

- Let  $\overline{co}(\mathcal{F})$  is the closure of  $co(\mathcal{F})$ .
- Let  $l(y, f(\mathbf{x}))$  be a given loss function.
- The objective of the CHB algorithm is to find  $\hat{H}$  in  $\overline{co}(\mathcal{F})$  defined by

$$\hat{H} = \arg \min_{H \in \overline{co}(\mathcal{F})} \sum_{i=1}^n l(y_i, H(\mathbf{x}_i)).$$

## Simple forward stagewise function approximation algorithm

1. Set  $H_0(\mathbf{x}) \equiv 0$ .

2. For  $m = 1$  to  $M$

(a) Let

$$(\hat{\alpha}, \hat{f}) = \arg \min_{\alpha \in [0,1], f \in \mathcal{F}} \sum_{i=1}^n l(y_i, (1-\alpha)H_{m-1}(\mathbf{x}_i) + \alpha f(\mathbf{x}_i)).$$

(b)  $H_m(\mathbf{x}) = (1 - \hat{\alpha})H_{m-1}(\mathbf{x}) + \hat{\alpha}\hat{f}(\mathbf{x})$

3. Prediction using  $H_M(\mathbf{x})$ .

### Problem in the simple forward stagewise function approximation

- Obtaining  $(\hat{\alpha}, \hat{f})$  is numerically very hard. In particular, the object function to be minimized is not convex with respect to  $(\alpha, f)$ .
- CHB algorithm is devised to overcome such deficiency in the simple forward stagewise function approximation algorithm.

## Idea of CHB algorithm

1. First, fix  $\alpha$ .
2. Get  $\hat{f} = \arg \min_{f \in \mathcal{F}} C_n((1 - \alpha)H_{m-1} + \alpha f)$  where

$$C_n(f) = \sum_{i=1}^n l(y_i, f(\mathbf{x}_i))/n.$$

3. Let  $\tilde{H} = (1 - \alpha)H_{m-1} + \alpha \hat{f}$ .
4. If  $C_n(\tilde{H}) \geq C_n(H_{m-1})$ , then reduce  $\alpha = \alpha/2$  and return to 2.
5. Otherwise, get  $\hat{\alpha} = \arg \min_{\alpha \in [0,1]} C_n((1 - \alpha)H_{m-1} + \alpha \hat{f})$  and update  $H_m = (1 - \hat{\alpha})H_{m-1}(\mathbf{x}_i) + \hat{\alpha} \hat{f}(\mathbf{x}_i)$ .

## Explanation of CHB algorithm

- Suppose  $C_n$  is smoothly convex. Then, unless  $H_{m-1}$  is the global minimizer of  $C_n$ , there exists  $\alpha^* > 0$  such that  $\min_{f \in \mathcal{F}} C_n((1 - \alpha)H_{m-1} + \alpha f) < C_n(H_{m-1})$  for all  $\alpha \in (0, \alpha^*)$ .
- Hence, for sufficiently small  $\alpha$ , we can always find  $\hat{f}$  in Step 2 which satisfies  $C_n(\tilde{H}) < C_n(H_{m-1})$ .
- However, too small  $\alpha$  results in too small improvement of  $H_m$  compared to  $H_{m-1}$ .
- Hence, we start from a larger value of  $\alpha$  and reduce it by half.

## $\alpha^*$ and the optimality

- If  $H_{m-1}$  is the optimum,  $\alpha^* = 0$ .
- Furthermore, we have

$$C_n(H_{m-1}) - C_n(H^*) \leq D\alpha^*$$

for some constant  $D$  under regularity conditions where  $H^*$  is the optimum.

- The above inequality implies that if we cannot find  $\tilde{H}$  improving  $H_{m-1}$  until sufficiently small  $\alpha$ , we can consider  $H_{m-1}$  as a near optimum without losing much information.

## CHB algorithm

Set  $H_0(\mathbf{x}) \equiv 0$ ,  $m = 0$  and  $Stop = F$ .

(1) While  $Stop = F$

$m = m + 1$  and  $\alpha = \alpha_m$ .

(2) Get  $\hat{f} = \arg \min_{f \in \mathcal{F}} C_n((1 - \alpha)H_{m-1} + \alpha f)$

    Let  $\tilde{H} = (1 - \alpha)H_{m-1} + \alpha \hat{f}$ .

    If  $C_n(\tilde{H}) \geq C_n(H_{m-1})$  then

$\alpha = \alpha/2$

        If  $\alpha > \epsilon$  then

            Goto (2)

        Else

$Stop = T$  and Goto (1)

        End If

    Else

        Get  $\hat{\alpha} = \arg \min_{\alpha \in [0,1]} C_n((1 - \alpha)H_{m-1} + \alpha \hat{f})$

        Update  $H_m = (1 - \hat{\alpha})H_{m-1}(\mathbf{x}_i) + \hat{\alpha} \hat{f}(\mathbf{x}_i)$ .

        If  $m \geq M$ ,  $Stop = T$ .

    End If

End While

Prediction using  $H_m(\mathbf{x})$ .

### Remark.

- The algorithm has two stopping criteria: one with  $\alpha$  and the other for the number of base hypotheses  $m$ .
- The starting values of  $\alpha$  ( $\alpha_m$  in the algorithm) depend on the number of hypotheses. Empirical studies show that  $\alpha_m = 1/m$  works well in most cases.

Convergence of CHB algorithm.

**Theorem 1** *Let  $l'(y, a) = \partial l(y, a) / \partial a$ . Suppose (1)  $l'$  satisfies a Lipschitz condition and (2)  $\sup_{f \in \mathcal{F}} |f| < \infty$  where  $|f| = \sup_{\mathbf{x} \in R^p} |f(\mathbf{x})|$ . Then, CHB algorithm converges to the optimum if  $\epsilon = 0$  and  $M = \infty$ .*

## 5. Learning Theory

- Suppose
  - Pseudo dimension of  $\mathcal{F}$  is finite ( $= V$ )
  - $\sup_{f \in \mathcal{F}} |f| (= \gamma) < \infty$ ;
  - $Y \in [-B, B]$  where  $B < \infty$ ;
  - $|l(y, a_1) - l(y, a_2)| \leq L_\gamma |a_1 - a_2|$  for all  $y \in [-B, B]$  and  $a_1, a_1 \in [-\gamma, \gamma]$ .
- Then, we have

$$\Pr \left\{ \sup_{H \in \overline{\text{co}}(\mathcal{F})} |C(H) - C_n(H)| \leq \beta(V, \gamma, \delta) / \sqrt{n} \right\} \geq 1 - \delta$$

where  $C(H) = E_{(Y, \mathbf{X})} l(Y, H(\mathbf{X}))$ ,  $\beta$  is an increasing function in both  $V$  and  $\gamma$  but decreasing in  $\delta$  (Koltchinskii and Panchenko 2002).

### Remark

- The above inequality implies that the complexity can be controlled by  $V$  and  $\gamma$ .
- Usually, we fix  $V$  and control  $\gamma$  to find the best model.
- By doing this, we can achieve consistency.

## Consistency

- Suppose

- Let  $\mathcal{F}_\gamma = \{f : \sup |f| < \gamma, f \in \mathcal{F}\}$ .

- Let  $\mathcal{F}$  has finite pseudo dimension with  $\overline{\text{co}}(\mathcal{F}) = L_2(P)$

- Let  $\hat{H}_\gamma = \arg \min_{H \in \overline{\text{co}}(\mathcal{F}_\gamma)} C_n(H)$ .

- Let  $H^* = \arg \min_{H \in L_2(P)} C(H)$ .

- Then, we can find  $\{\gamma_n\}, \gamma_n \rightarrow \infty$  such that

$$C(\hat{H}_{\gamma_n}) \rightarrow C(H^*).$$

- Moreover, under regularity conditions, we can prove Bayes consistency for classification (Lugosi and Vayatis 2003) and  $L_2(P)$  convergence for regression problems.

- An example of such base learners is a set of decision trees with  $p + 1$  terminal nodes (Breiman, 2000).

## 6. CHB with decision trees

### Estimation of the terminal nodes

- Let  $\mathcal{F}$  be the class of decision trees with  $L$  many terminal nodes.
- That is  $f(\mathbf{x}) = \sum_{k=1}^L b_k I(\mathbf{x} \in R_k)$  where  $\{R_k\}$  are partitions of  $R^p$ .
- For given  $H_{m-1}, \alpha$  and  $\{R_k\}$ ,  $\hat{b}_k$  are estimated by

$$\hat{b}_k = \arg \min_{b: |b| \leq \gamma} \sum_{i=1}^n l(y_i, (1 - \alpha)H_{m-1}(\mathbf{x}_i) + \alpha b) I(\mathbf{x}_i \in R_k).$$

Choice of  $\{R_k\}$ .

- Start with the decision tree with 1 terminal node.
- For the current decision tree, choose the split rule which drops the value of  $C_n((1 - \alpha)H_{m-1} + \alpha f)$  most significantly.
- Repeat this procedure until the decision tree has  $L$  many terminal nodes.

## Gradient descent

- With some loss functions, finding the optimal split rule is computationally demanding.
- A remedy is a gradient descent as follows.
  - Let  $z_i = -l'(y_i, (1 - \alpha)H_{m-1}(\mathbf{x}_i))$  for  $i = 1, \dots, n$ .
  - Construct a regression tree with new responses  $\{z_i\}$ .
- In some cases, CHB with the gradient descent may be stuck on a sub-optimal solution.
- To avoid this, stochastic gradient descent can be used as is done by Friedman (1999).

## CHB algorithm with gradient descent

Set  $H_0(\mathbf{x}) \equiv 0$ ,  $m = 0$  and  $Stop = F$ .

(1) While  $Stop = F$

$m = m + 1$ ,  $\alpha = \alpha_m$  and  $bool = T$

(2) Get the negative gradient  $\{z_i\}$ .

Let  $z_i = z_i + \epsilon_i$  where  $\epsilon_i$  are mean 0 noises.

Construct a regression tree  $\hat{f}$  with  $\{z_i\}$  as responses.

Estimate the terminal nodes of  $\hat{f}$  by minimizing  $C_n((1 - \alpha)H_{m-1} + \alpha\hat{f})$ .

Let  $\tilde{H} = (1 - \alpha)H_{m-1} + \alpha\hat{f}$ .

If  $bool = F$  and  $C_n(\tilde{H}) \geq C_n(H_{m-1})$  then

If  $\alpha < \epsilon$  then

$bool = F$ ,  $\alpha = 1/m$  and Goto (2)

Else

$\alpha = \alpha/2$  and Goto (2)

End If

Else

Get  $\hat{\alpha} = \arg \min_{\alpha \in [0,1]} C_n((1 - \alpha)H_{m-1} + \alpha\hat{f})$

.....

## 7. Empirical Studies

- 6 classification data sets and 5 regression data sets are analyzed.
- The test errors are calculated based on 10 fold cross-validation unless the test data set is available.
- Stumps (decision trees with two terminal nodes) are used as base learners.
- For CHB,  $\gamma$  is selected using 10% validation set.
- To make the comparison fair, the optimal numbers of base learners in boosting algorithms are selected using 10% validation set.

Data set description for classification

ID	Data Set	Training	Test	Nr. Inputs
1	Breast Cancer (BC)	683	CV	9
2	Pime-Indian-Diabetes (PD)	768	CV	8
3	German (GE)	1000	CV	20
4	House-vote-84 (HV)	435	CV	16
5	Ionosphere (IO)	351	CV	34
6	kr-vs-kp (KP)	3196	CV	36

## CHB versus $L_1$ Boost and AdaBoost

- I use the exponential loss for CHB and  $L_1$  Boost.
- Generalization errors

Data	BC	DP	GE	HV	IO	KR
CHB	0.0352	0.2381	0.249	0.0372	0.0914	0.0514
$L_1$	0.0382	0.2355	0.246	0.0348	0.1028	0.0498
ADA	0.0544	0.25	0.272	0.0372	0.0942	0.0344

- Apparently, CHB and  $L_1$  Boost are competitive and AdaBoost is inferior (except one).

## CHB with various loss functions

- Along with the exponential loss, I use two more losses;
  - Trimmed exponential

$$l(y, a) = \begin{cases} \exp(-ya) & \text{if } ya > 0 \\ -ya & \text{if } ya < 0 \end{cases}$$

- Hinge loss

$$l(y, a) = \begin{cases} 0 & \text{if } ya > 1 \\ 1 - ya & \text{if } ya < 1 \end{cases}$$

- Exponential loss  $\geq$  Trimmed exponential loss  $\geq$  Hinge loss  $\geq$  0-1 loss
- Hinge loss is the tightest convex upper bound of the 0-1 loss (Lin 2001).
- For trimmed exponential and Hinge losses, I use stochastic gradient descent algorithm.

- Generalization errors

Data	BC	DP	GE	HV	IO	KR
EXP	0.0352	0.2381	0.249	0.0372	0.0914	0.0514
EXP-R	0.0352	0.2381	0.248	0.0348	0.0799	0.0479
TEXP	0.0323	0.2394	0.235	0.0395	0.0685	0.0467
HINGE	0.0308	0.2447	0.307	0.0441	0.1057	0.0755

EXP-R : CHB with the exponential loss and some randomization

- Apparently, trimmed exponential loss is the best and Hinge is the worst (but not always).
- For the exponential loss, the idea of randomization works.

## Remark

- *Penalization* (CHB and  $L_1$  Boost) outperforms *early stopping* (AdaBoost).
- Bad performance of Hinge loss may be due to that the loss function is non-differentiable so forward-stagewise fitting may not work.
- Randomization (EXP-R) still improves the structural risk minimization principle (EXP). This may be related to Bayesian averaging, but not clear.

Data description for regression

ID	Data Set	Training	Test	Nr. Inputs
1	Bostong Housing	506	CV	12
2	Friedman 1	200	2000	20
3	Friedman 2	200	2000	16
4	Friedman 3	200	2000	34
5	Servo	330	CV	8

## CHB versus $L_1$ Boost and Gradient Boosting

- $L_2$  loss is used.
- Mean absolute deviations

Data	BO	F1	F2	F3	SV
CHB	1.6692	1.6253	23.7806	0.1568	0.6583
L1	2.6350	1.8670	25.0808	0.1603	0.6698
GB	1.7236	2.0671	27.7271	0.1787	0.6885

- CHB outperforms Gradient Boost as well as  $L_1$  Boost for the all data sets.

## CHB with various loss functions

- Along with  $L_2$  loss, I use the Huber loss and absolute loss.
- For Huber and absolute losses, I use the CHB with stochastic gradient descent.
- Mean absolute deviations

Data	BO	F1	F2	F3	SV
$L_2$	1.6692	1.6253	23.7806	0.1568	0.6583
$L_2$ -R	1.6849	1.6840	23.4802	0.1560	0.6672
Huber	1.6442	1.5738	24.4984	0.1531	0.6917
Abs	1.9461	1.6307	25.1105	0.1547	0.4458

$L_2$ -R : CHB with the  $L_2$  loss and some randomization

- Apparently,  $L_2$  and Huber losses are competitive and absolute loss is the worst.
- Apparently, the effect of randomization is not significant.

## 8. Conclusions

- Obvious facts
  - CHB works well for both of classification and regression problems with various loss functions.
  - CHB outperforms boosting algorithms.
- Less clear facts
  - CHB does not work well for non-differentiable convex loss functions.
  - Randomization improves the structural risk minimization principle only for classification problems.
- Future works
  - CHB with non-convex loss functions.