

# Machine Learning

## Lecture 6: Gradient boosting

Iurii Efimov



girafe  
ai

# Outline

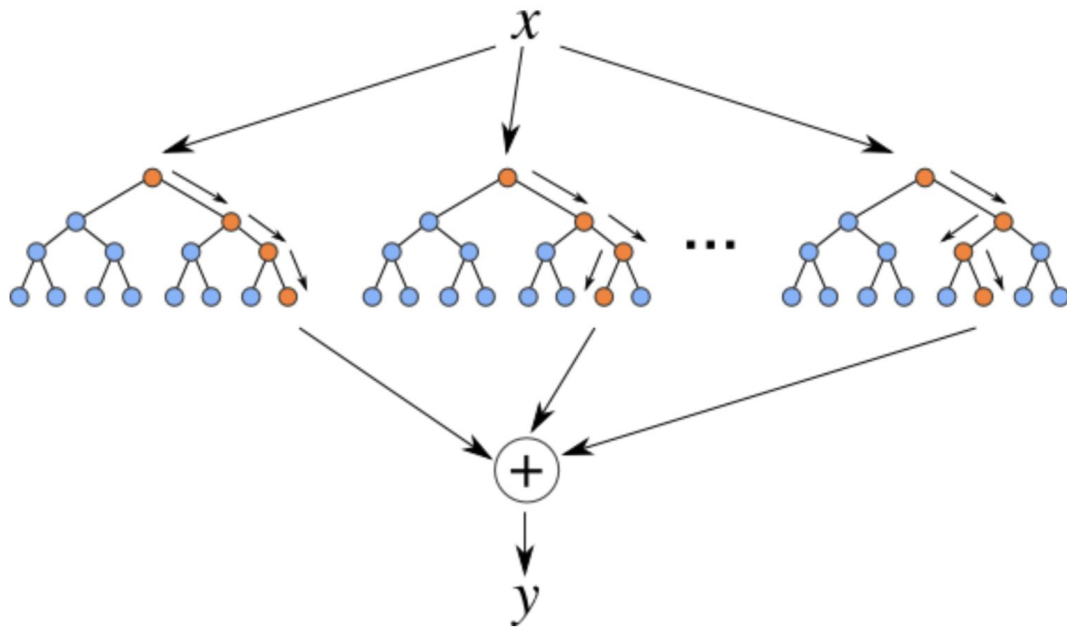
1. Boosting intuitions
2. Gradient boosting
- 3.

Based on: ml-mipt 2017 by Victor Kantor,  
Evgeny Sokolov HSE 2018, mlcourse\_open by ODS

# Random Forest



Bagging + RSM = Random Forest



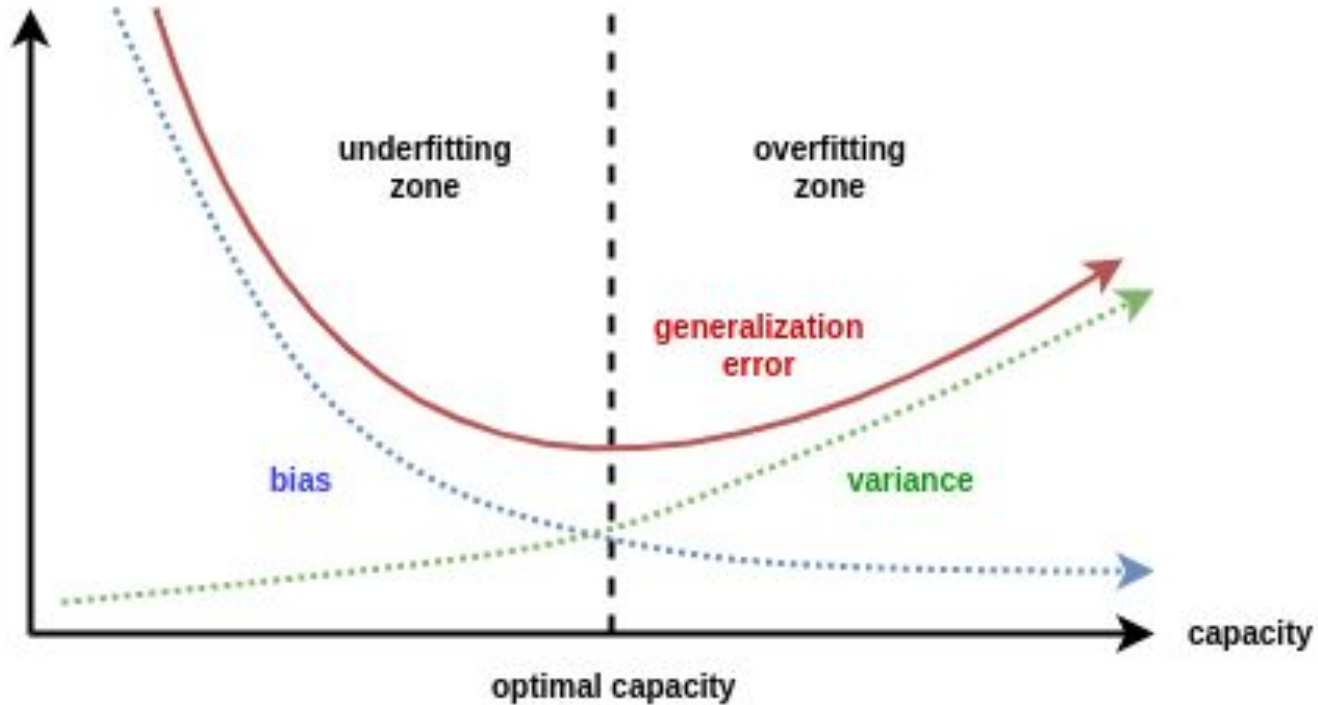


# Random Forest

- One of the greatest “universal” models.
- There are some modifications: Extremely Randomized Trees, Isolation Forest, etc.
- Allows to use train data for validation: OOB

$$\text{OOB} = \sum_{i=1}^{\ell} L \left( y_i, \frac{1}{\sum_{n=1}^N [x_i \notin X_n]} \sum_{n=1}^N [x_i \notin X_n] b_n(x_i) \right)$$

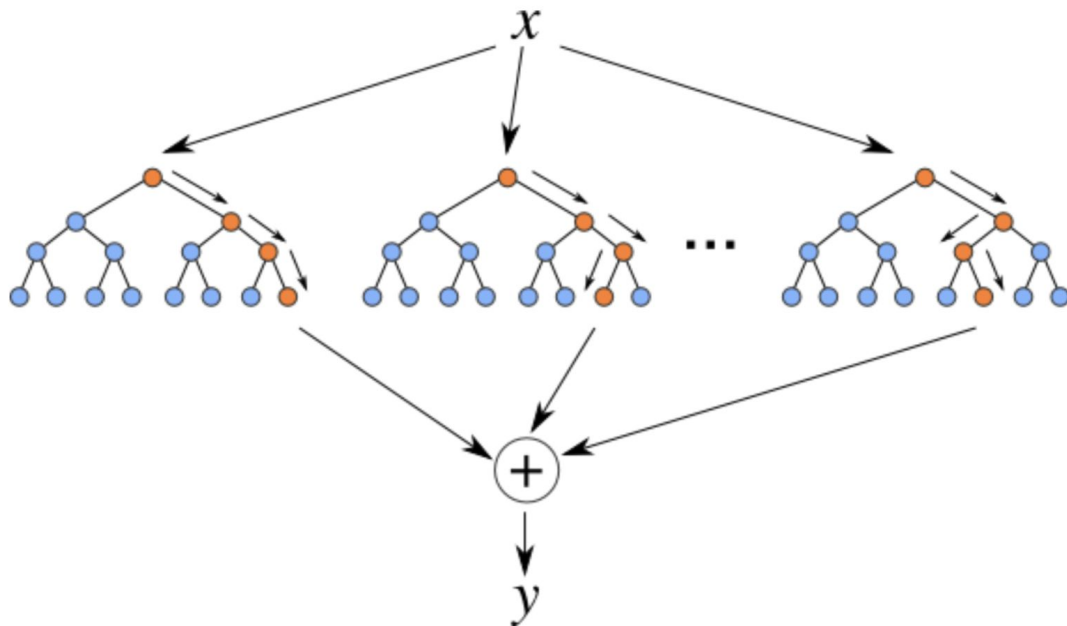
# Bias-variance tradeoff



# Random Forest



Is Random Forest decreasing bias or variance by building the trees ensemble?



# Boosting

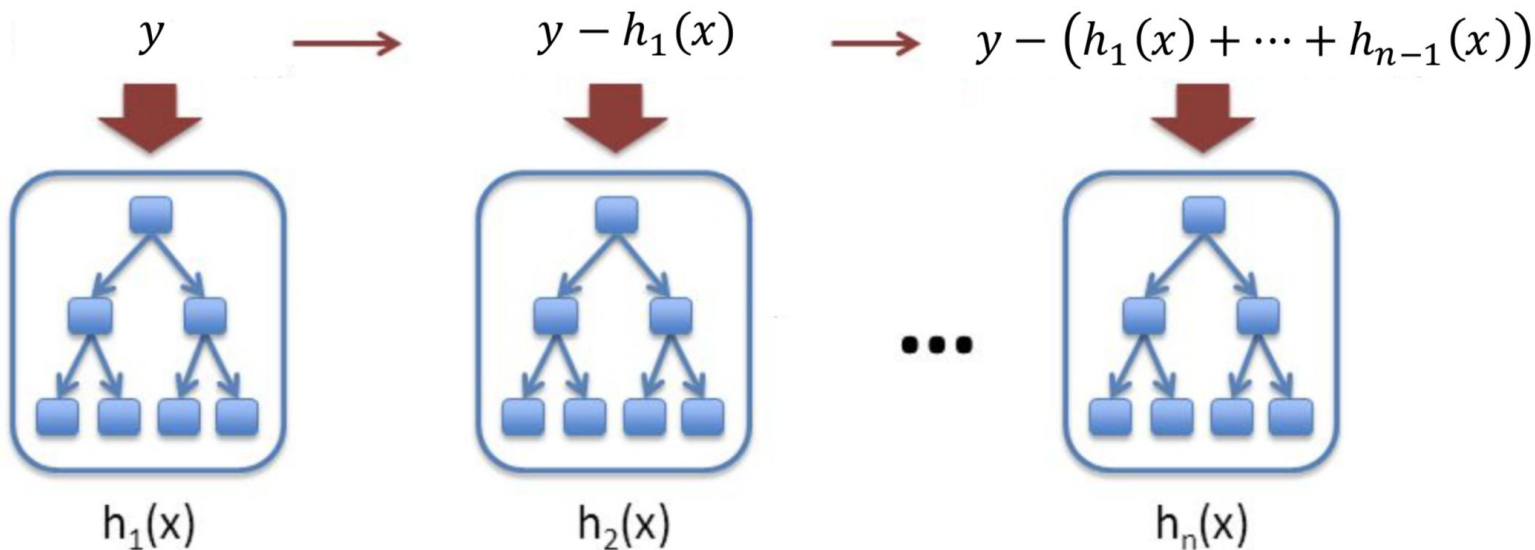
---

girafe  
ai

# Boosting



$$a_n(x) = h_1(x) + \dots + h_n(x)$$

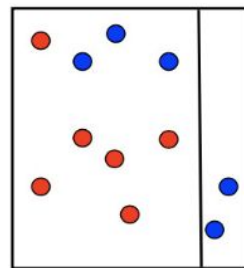
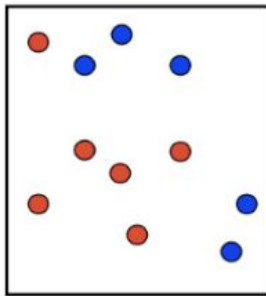




# Boosting: intuition

Binary classification

Use decision stumps.



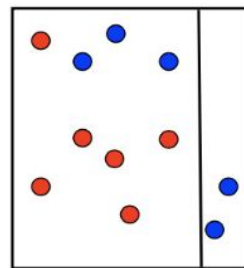
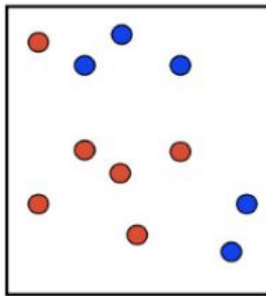
$t = 1$



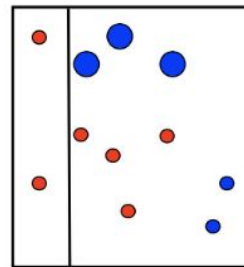
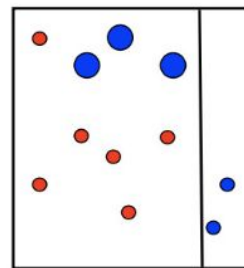
# Boosting: intuition

Binary classification

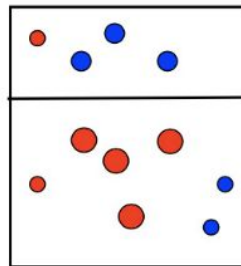
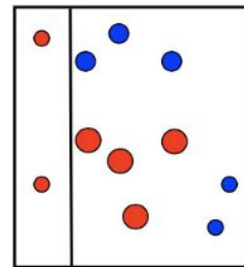
Use decision stumps.



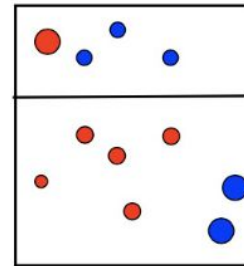
$t = 1$



$t = 2$



$t = 3$

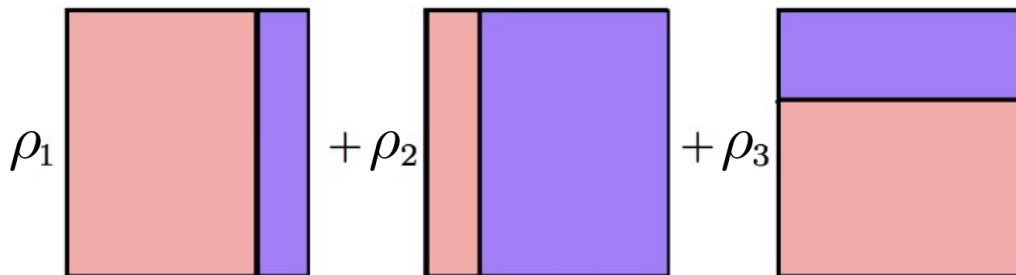
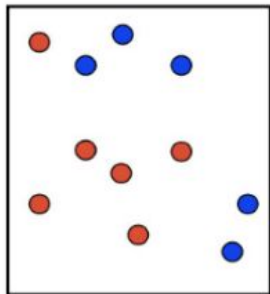


# Boosting: intuition

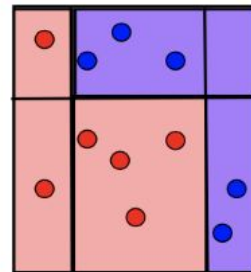


Binary classification

Use decision stumps.



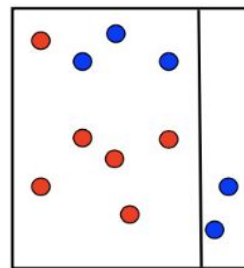
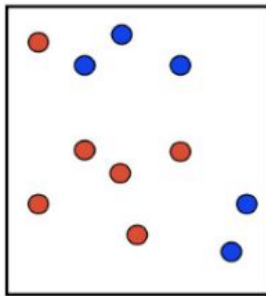
$$\hat{f}_T(x) = \sum_{t=1}^T \rho_t h_t(x) =$$



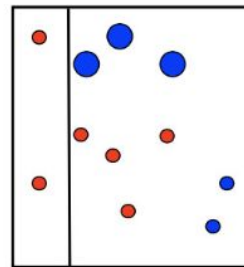
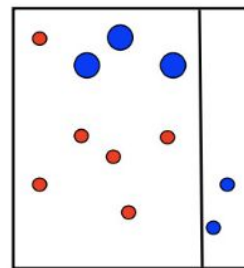
# Boosting: intuition

Binary classification

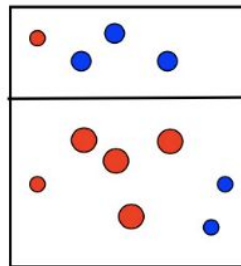
Use decision stumps.



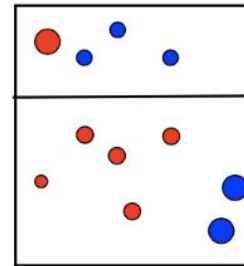
$t = 1$



$t = 2$



$t = 3$



# Gradient boosting

---

girafe  
ai

# Gradient boosting: theory



Denote dataset  $\{(x_i, y_i)\}_{i=1, \dots, n}$ , loss function  $L(y, f)$

Optimal model:

$$\hat{f}(x) = \arg \min_{f(x)} L(y, f(x)) = \arg \min_{f(x)} \mathbb{E}_{x,y}[L(y, f(x))]$$

Let it be from parametric family:

$$\begin{aligned}\hat{f}(x) &= f(x, \hat{\theta}), \\ \hat{\theta} &= \arg \min_{\theta} \mathbb{E}_{x,y}[L(y, f(x, \theta))]\end{aligned}$$



# Gradient boosting: theory

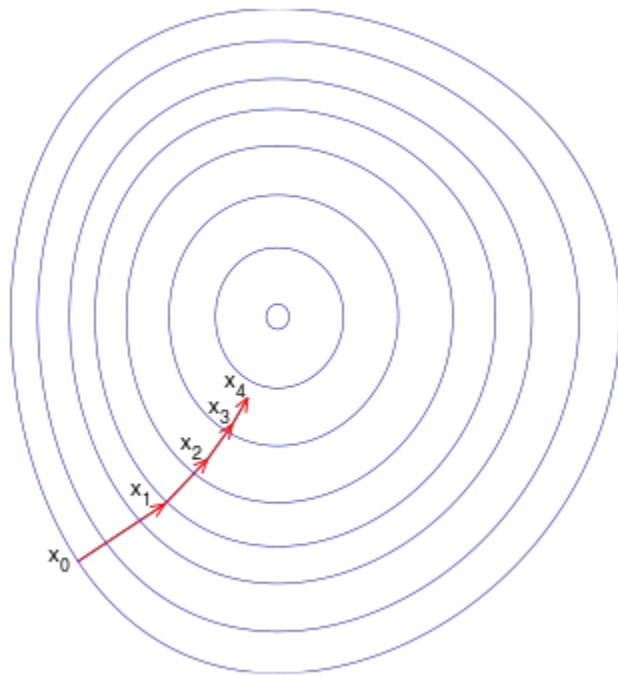
$$\hat{f}(x) = \sum_{i=0}^{t-1} \hat{f}_i(x),$$

$$(\rho_t, \theta_t) = \arg \min_{\rho, \theta} \mathbb{E}_{x,y} [L(y, \hat{f}(x) + \rho \cdot h(x, \theta))],$$

$$\hat{f}_t(x) = \rho_t \cdot h(x, \theta_t)$$

What if we could use gradient descent in space of our models?

# Gradient boosting: theory



What if we could use gradient descent in space of our models?





# Gradient boosting: theory

$$\hat{f}(x) = \sum_{i=0}^{t-1} \hat{f}_i(x),$$

$$r_{it} = - \left[ \frac{\partial L(y_i, f(x_i))}{\partial f(x_i)} \right]_{f(x)=\hat{f}(x)}, \quad \text{for } i = 1, \dots, n,$$

$$\theta_t = \arg \min_{\theta} \sum_{i=1}^n (r_{it} - h(x_i, \theta))^2,$$

$$\rho_t = \arg \min_{\rho} \sum_{i=1}^n L(y_i, \hat{f}(x_i) + \rho \cdot h(x_i, \theta_t))$$

# Gradient boosting: theory



In linear regression case with MSE loss:

$$r_{it} = - \left[ \frac{\partial L(y_i, f(x_i))}{\partial f(x_i)} \right]_{f(x)=\hat{f}(x)} = -2(\hat{y}_i - y_i) \propto \hat{y}_i - y_i$$

# Gradient boosting: beautiful demo



Great demo:

[http://arogozhnikov.github.io/2016/06/24/gradient\\_boosting\\_explained.html](http://arogozhnikov.github.io/2016/06/24/gradient_boosting_explained.html)

# Gradient boosting



What we need:

- Data.
- Loss function and its gradient.
- Family of algorithms (with constraints if necessary).
- Number of iterations  $M$ .
- Initial value (GBM by Friedman): constant.



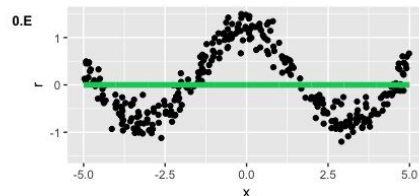
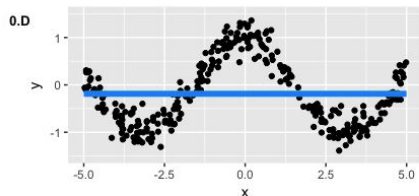
# Gradient boosting: example

What we need:

- Data: toy dataset  $y = \cos(x) + \epsilon, \epsilon \sim \mathcal{N}(0, \frac{1}{5}), x \in [-5, 5]$
- Loss function: MSE
- Family of algorithms: decision trees with depth 2
- Number of iterations  $M = 3$
- Initial value: just mean value

Example by ODS; source: <https://habr.com/ru/company/ods/blog/327250/>

# Gradient boosting: example

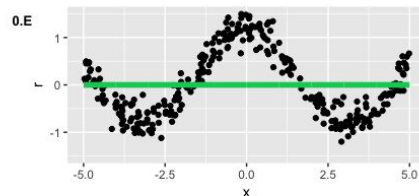
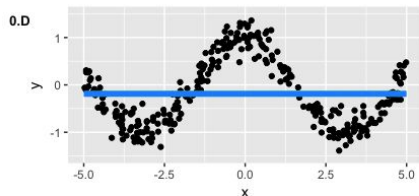


Left: full ensemble on each step.

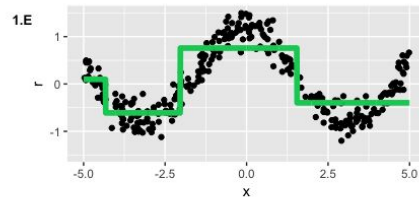
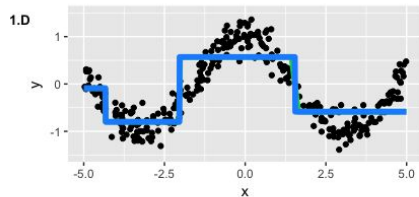
Right: additional tree decisions.

Example by ODS; source:  
<https://habr.com/ru/company/ods/blog/327250/>

# Gradient boosting: example



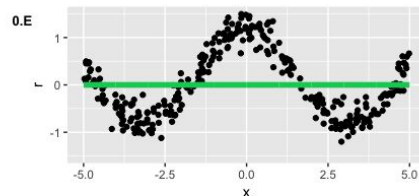
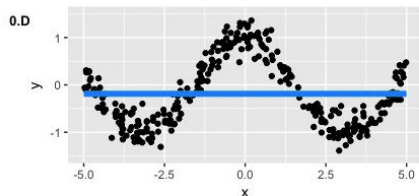
Left: full ensemble on each step.



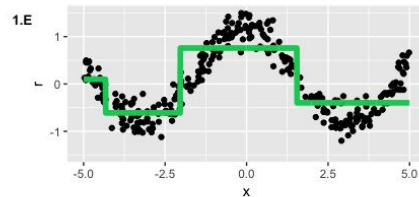
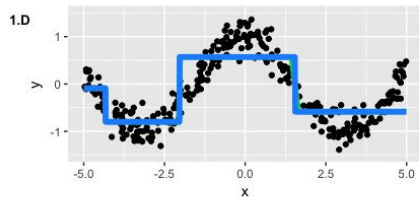
Right: additional tree decisions.

Example by ODS; source:  
<https://habr.com/ru/company/ods/blog/327250/>

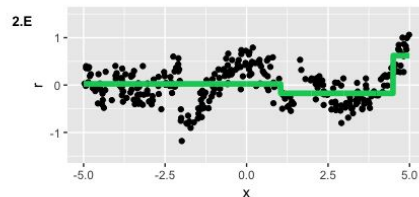
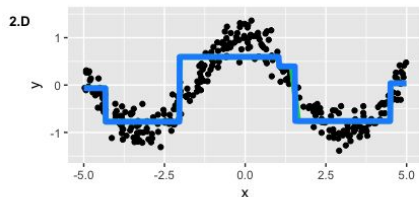
# Gradient boosting: example



Left: full ensemble on each step.



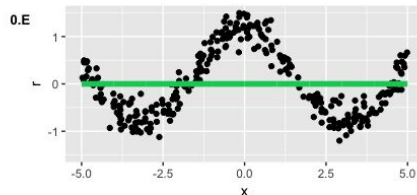
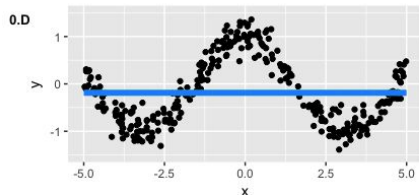
Right: additional tree decisions.



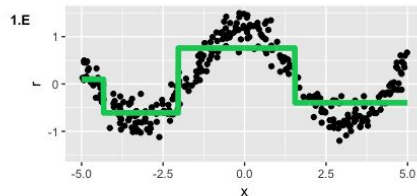
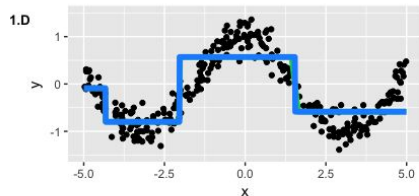
Example by ODS; source:  
<https://habr.com/ru/company/ods/blog/327250/>



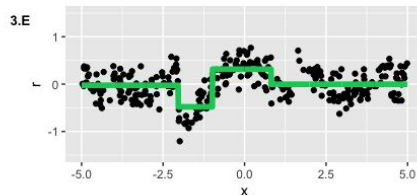
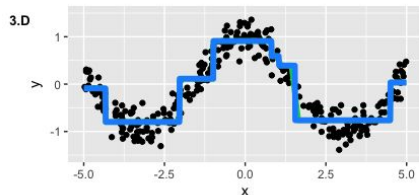
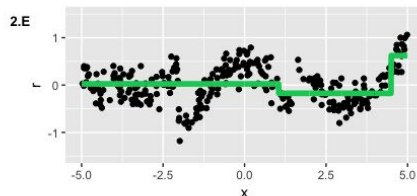
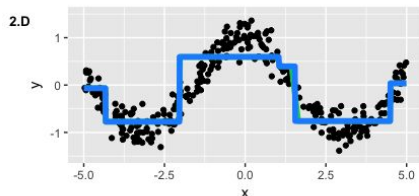
# Gradient boosting: example



Left: full ensemble on each step.

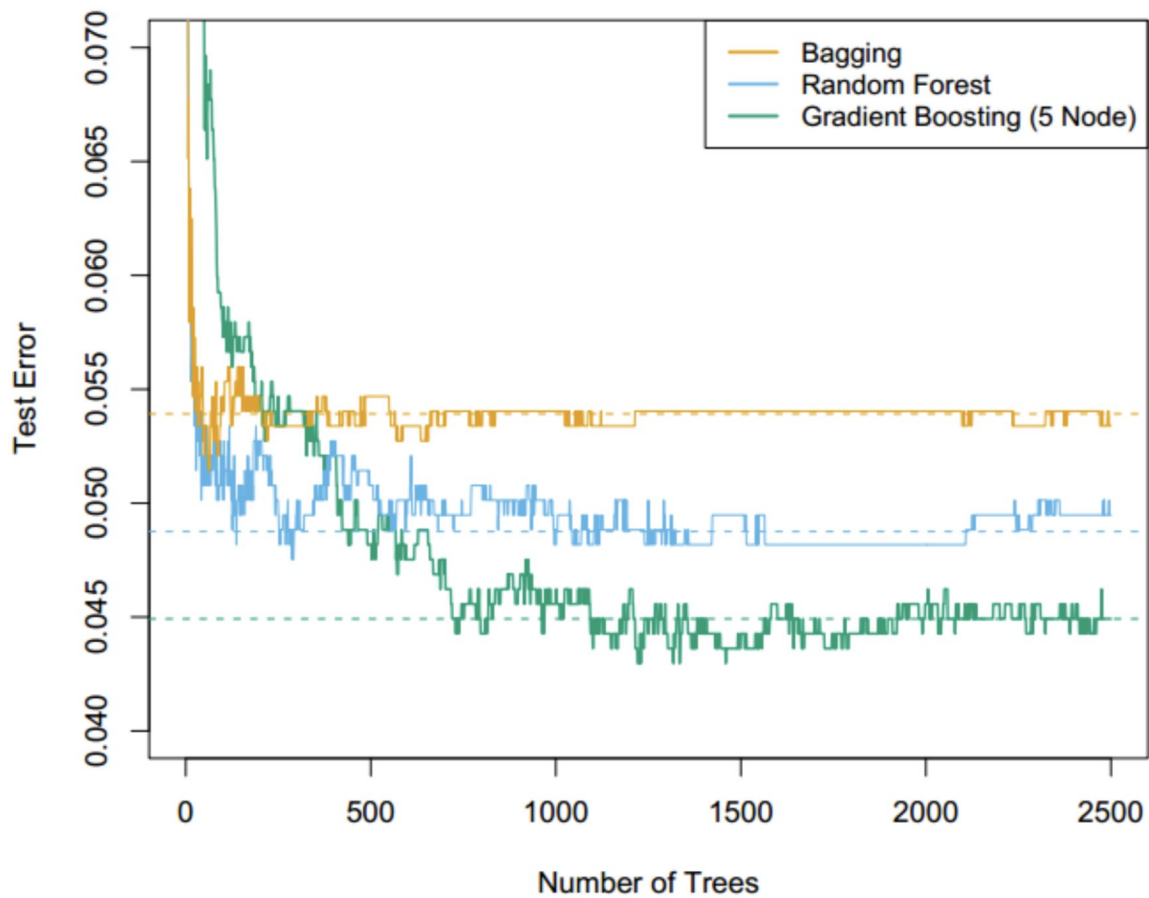


Right: additional tree decisions.

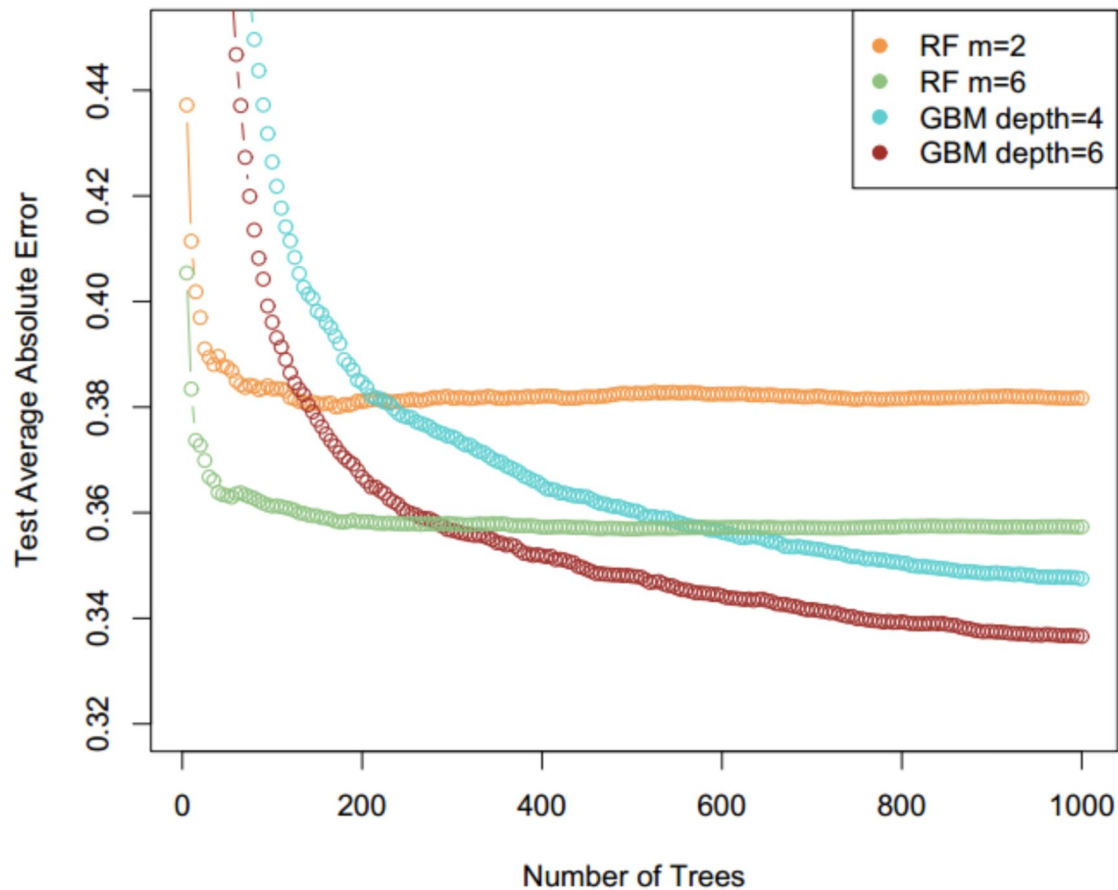


Example by ODS; source:  
<https://habr.com/ru/company/ods/blog/327250/>

## Spam Data



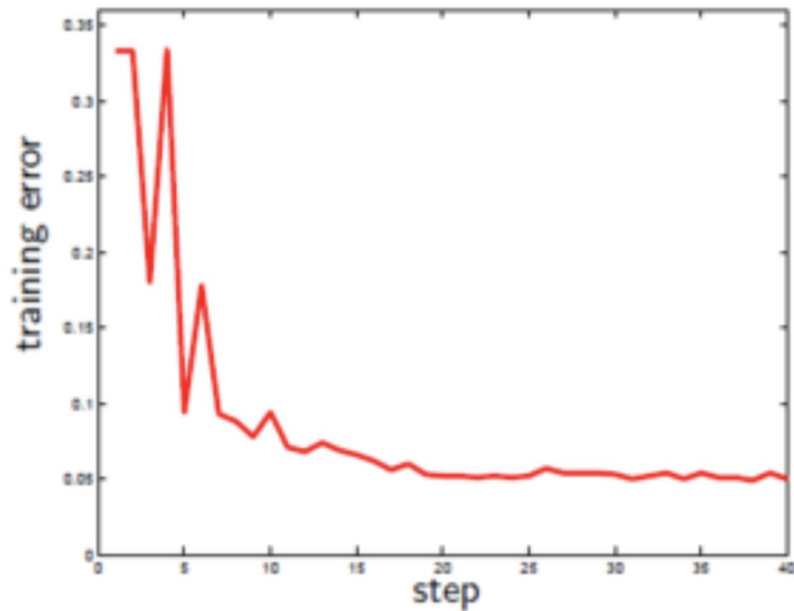
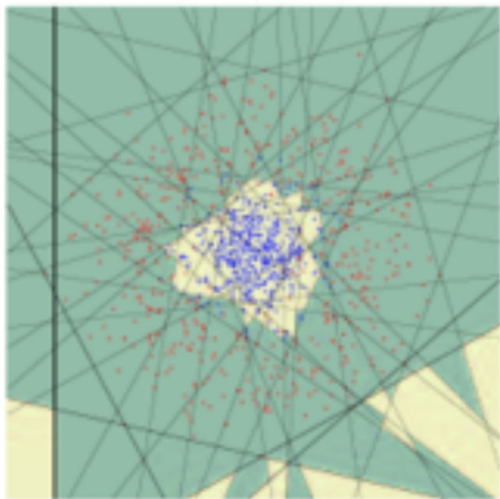
## California Housing Data



# Boosting with linear classification methods



$t = 40$



# Technical side: training in parallel



Which of the ensembling methods could be parallelized?

- Random Forest: parallel on the forest level (all trees are independent)
- Gradient boosting: parallel on one tree level

# Revise



1. Boosting intuitions
2. Gradient boosting

# Thanks for attention!

Questions?

