

академия  
больших  
данных



Machine Learning, Lecture 6

# Gradient boosting

Radoslav Neychev, Fall 2021

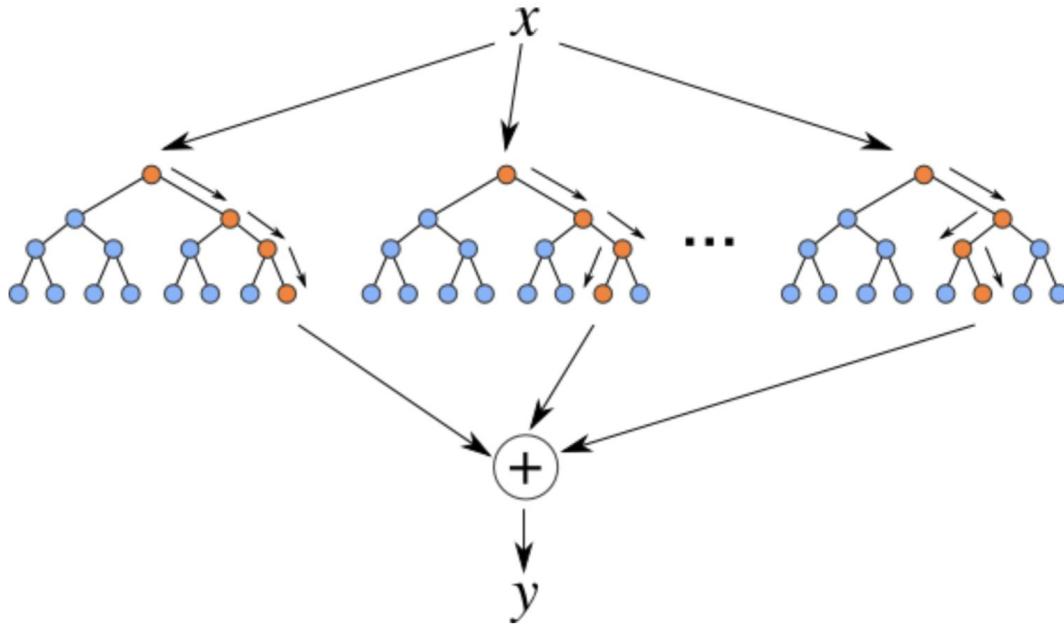
# Outline

1. Trees and ensembles recap
2. Boosting intuitions
3. Gradient boosting

# Random Forest



Bagging + RSM = Random Forest





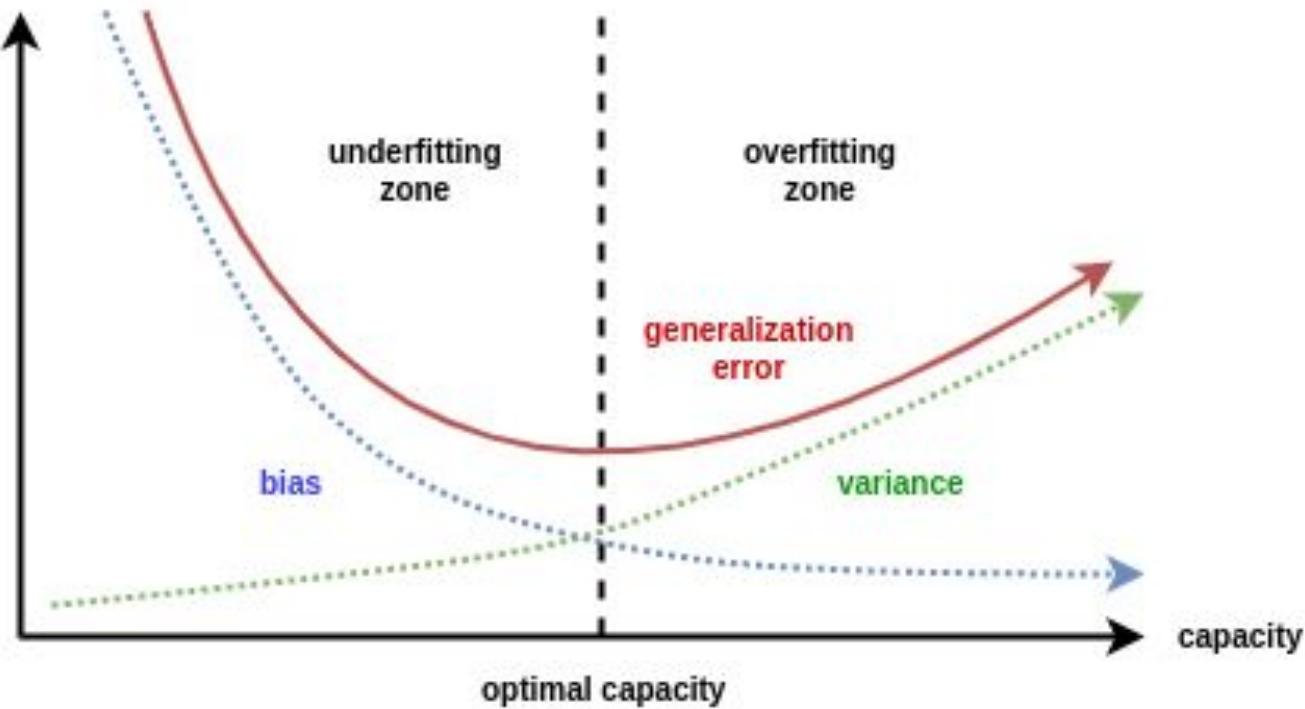
# Random Forest

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

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



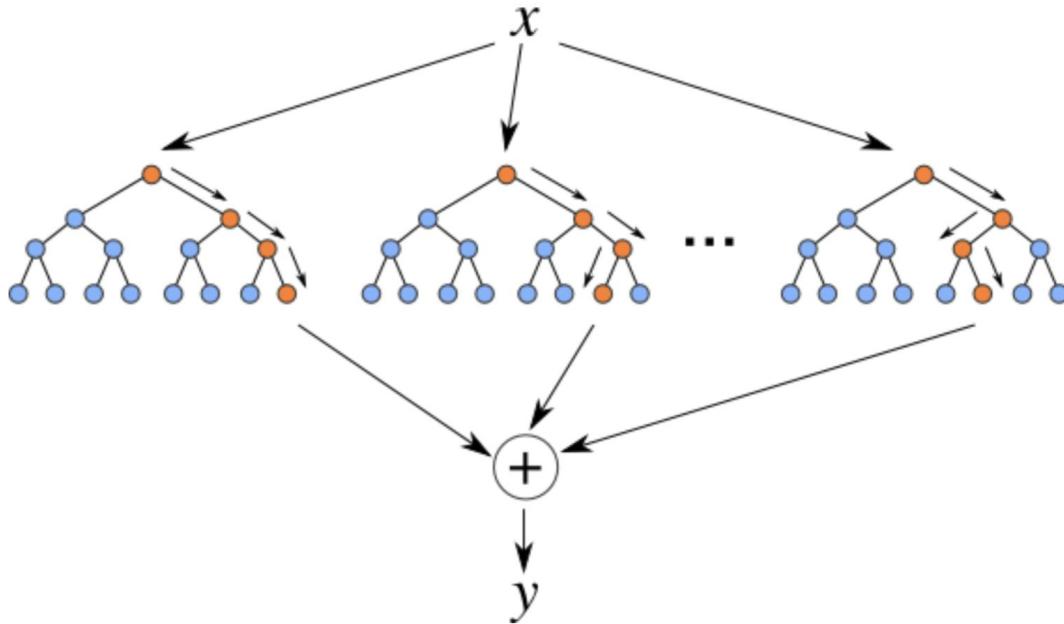
# Bias-variance tradeoff





# Random Forest

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



# Boosting

---

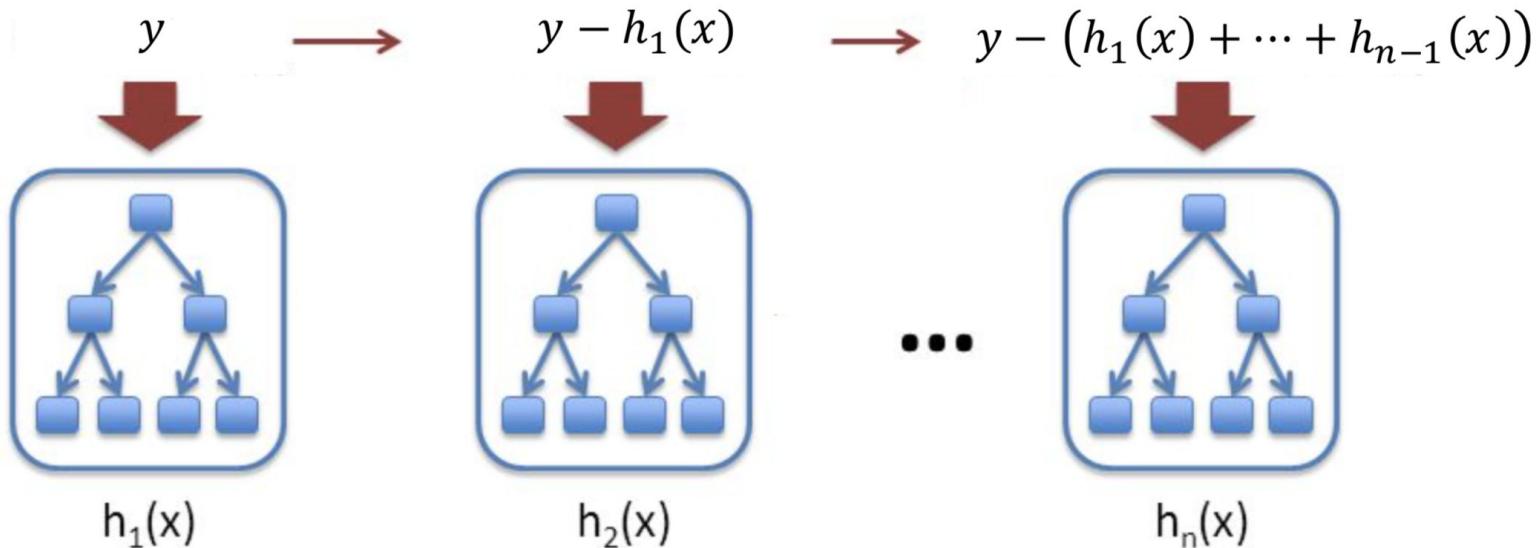
girafe  
ai

01

# Boosting



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

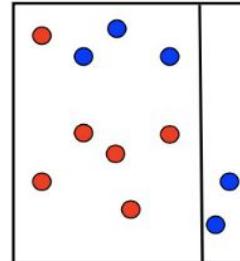
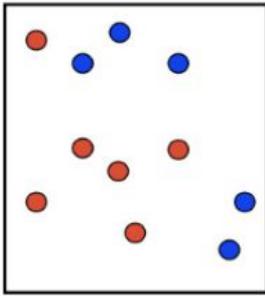


# Boosting: intuition



Binary classification

Use decision stumps.

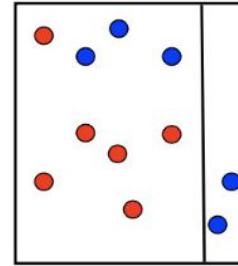
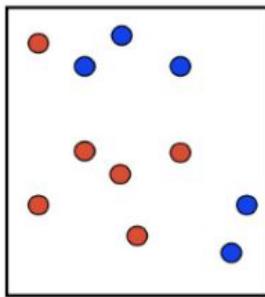


# Boosting: intuition

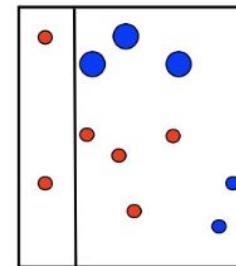


Binary classification

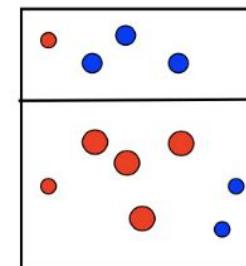
Use decision stumps.



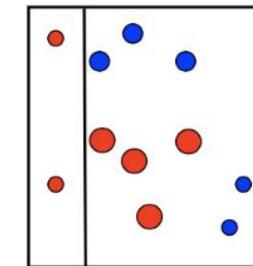
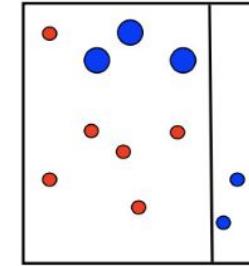
$t = 1$



$t = 2$



$t = 3$

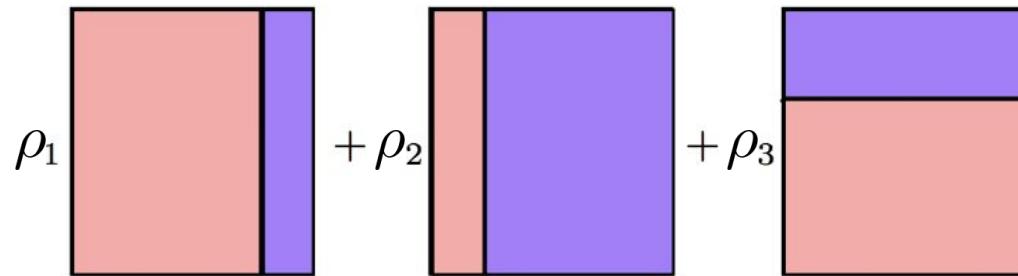
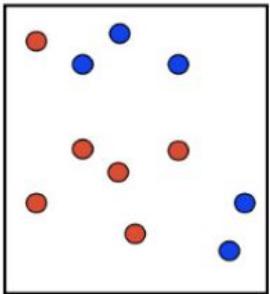




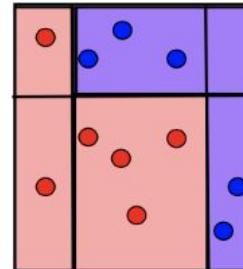
# Boosting: intuition

Binary classification

Use decision stumps.

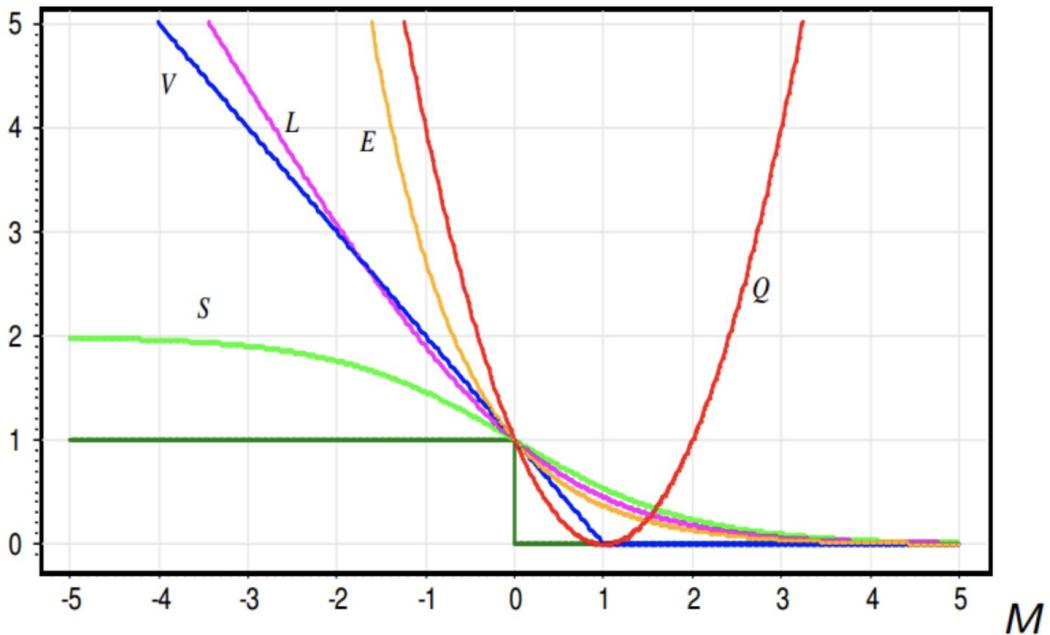


$$\hat{f}_T(x) = \sum_{t=1}^T \rho_t h_t(x)$$





# Recap: loss functions for classification



$$Q(M) = (1 - M)^2$$

$$V(M) = (1 - M)_+$$

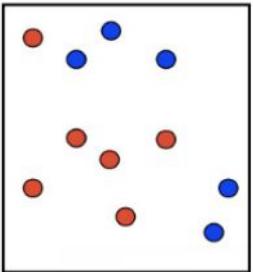
$$S(M) = 2(1 + e^M)^{-1}$$

$$L(M) = \log_2(1 + e^{-M})$$

$$E(M) = e^{-M}$$



# Boosting: AdaBoost



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

$$L(y_i, \hat{f}_T(x_i)) = \exp(-y_i \hat{f}_T(x_i)) = \exp\left(-y_i \sum_{t=1}^T \rho_t h_t(x_i)\right)$$

$$= \boxed{\exp\left(-y_i \sum_{t=1}^{T-1} \rho_t h_t(x_i)\right)} \cdot \exp(-y_i \rho_T h_T(x_i))$$

const on step T

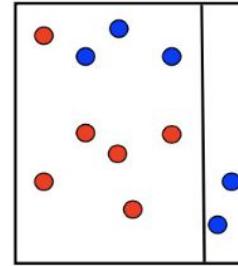
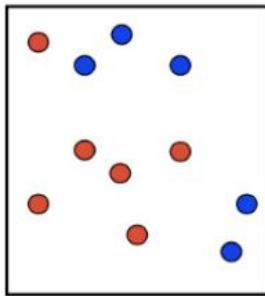
$$= w_i \cdot \exp(-y_i \rho_T h_T(x_i))$$

# Boosting: intuition

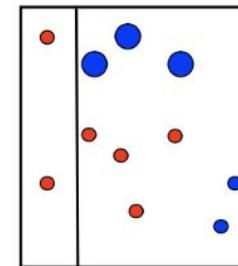


Binary classification

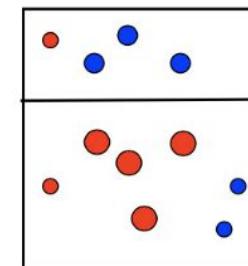
Use decision stumps.



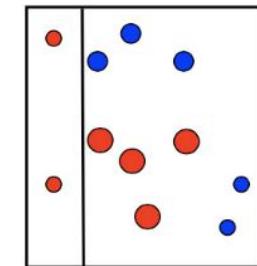
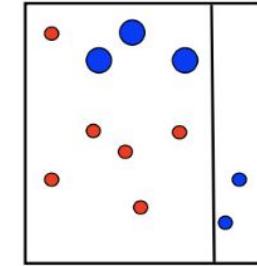
$t = 1$



$t = 2$



$t = 3$



# Gradient boosting

---

girafe  
ai

02



# 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:

$$\hat{f}(x) = f(x, \hat{\theta}),$$

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



# Gradient boosting: theory

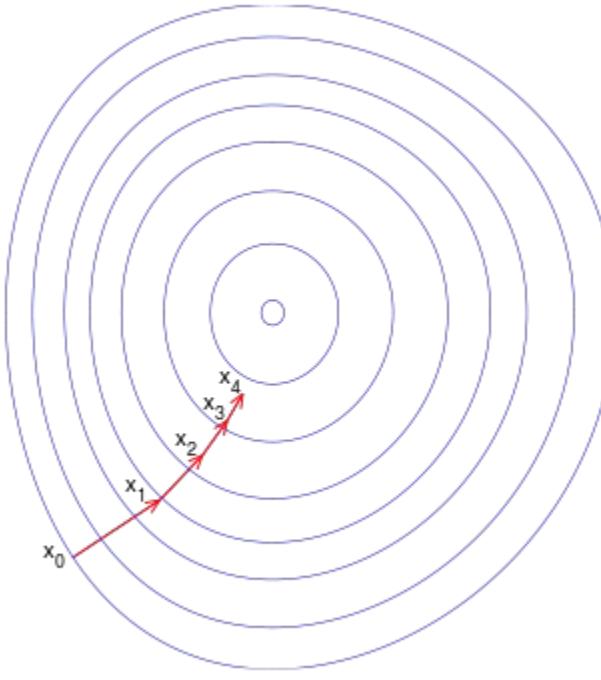
$$\hat{f}(x) = \sum_{i=0}^{t-1} \hat{f}_i(x),$$

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

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

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.



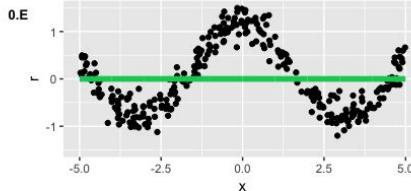
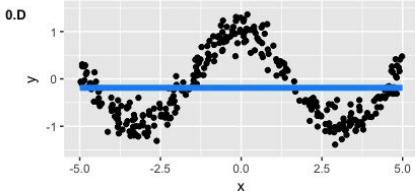
# Gradient boosting: example

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 valu



# Gradient boosting: example



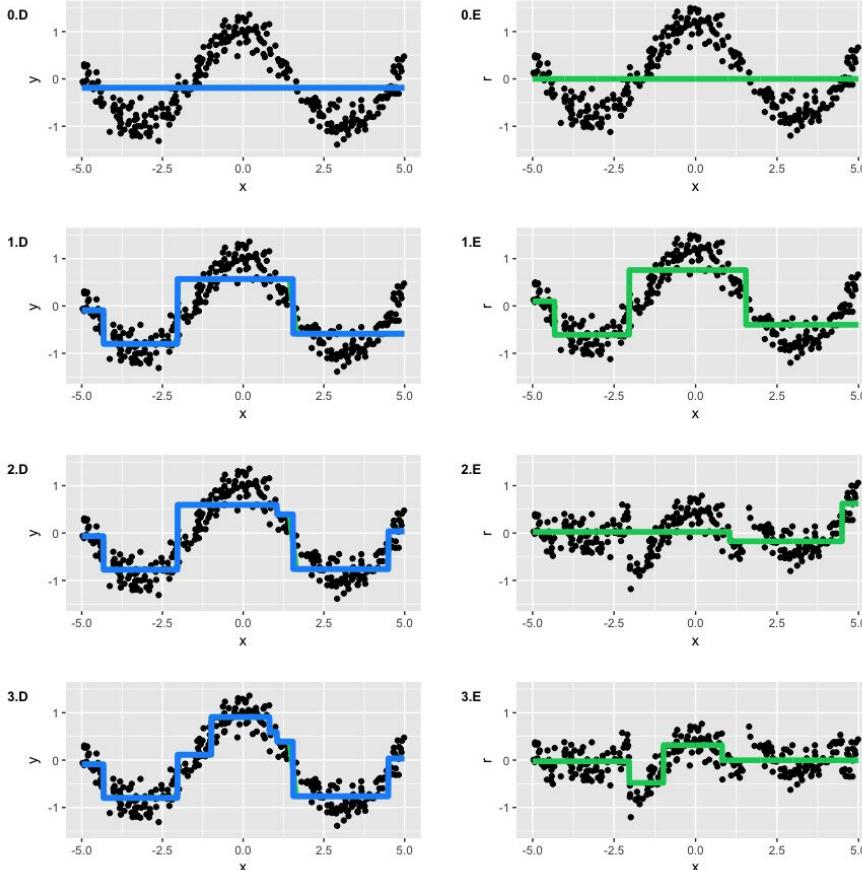
Left: full ensemble on each step.

Right: additional tree decisions.

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



# Gradient boosting: example

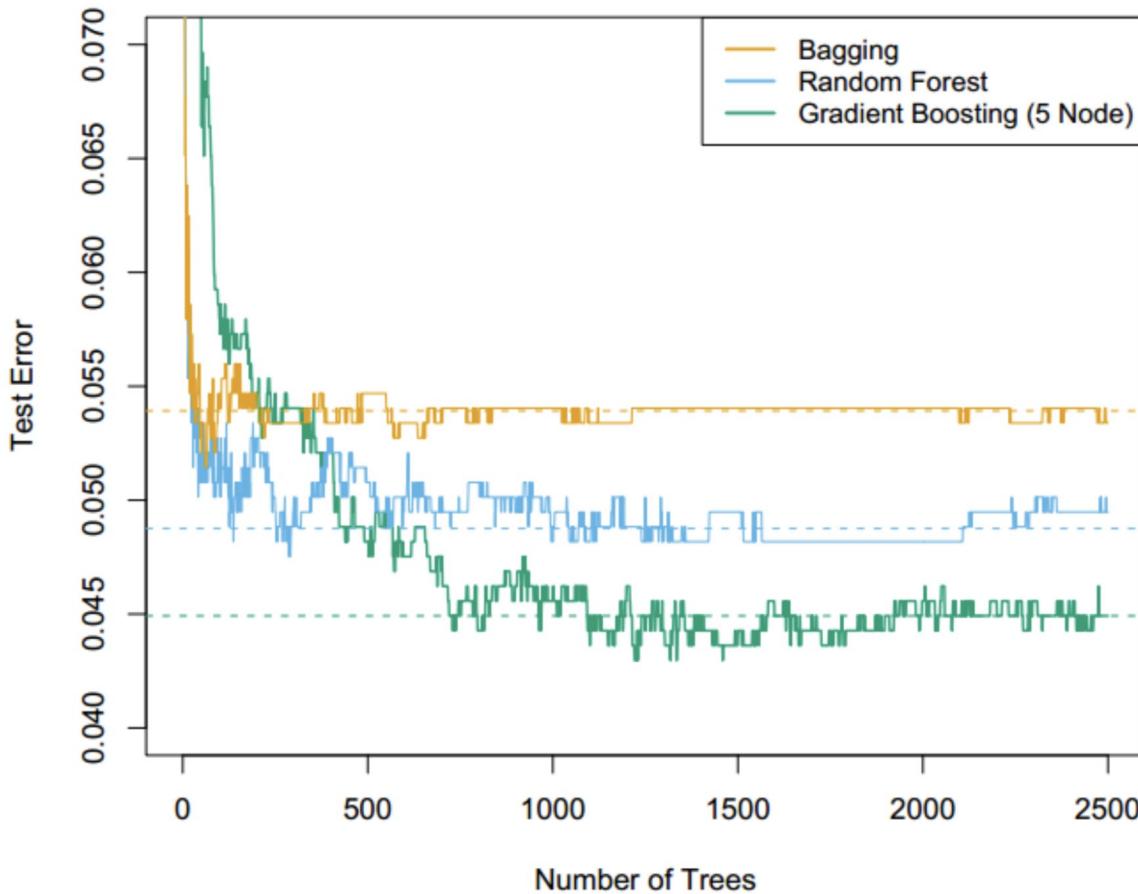


Left: full ensemble on each step.

Right: additional tree decisions.

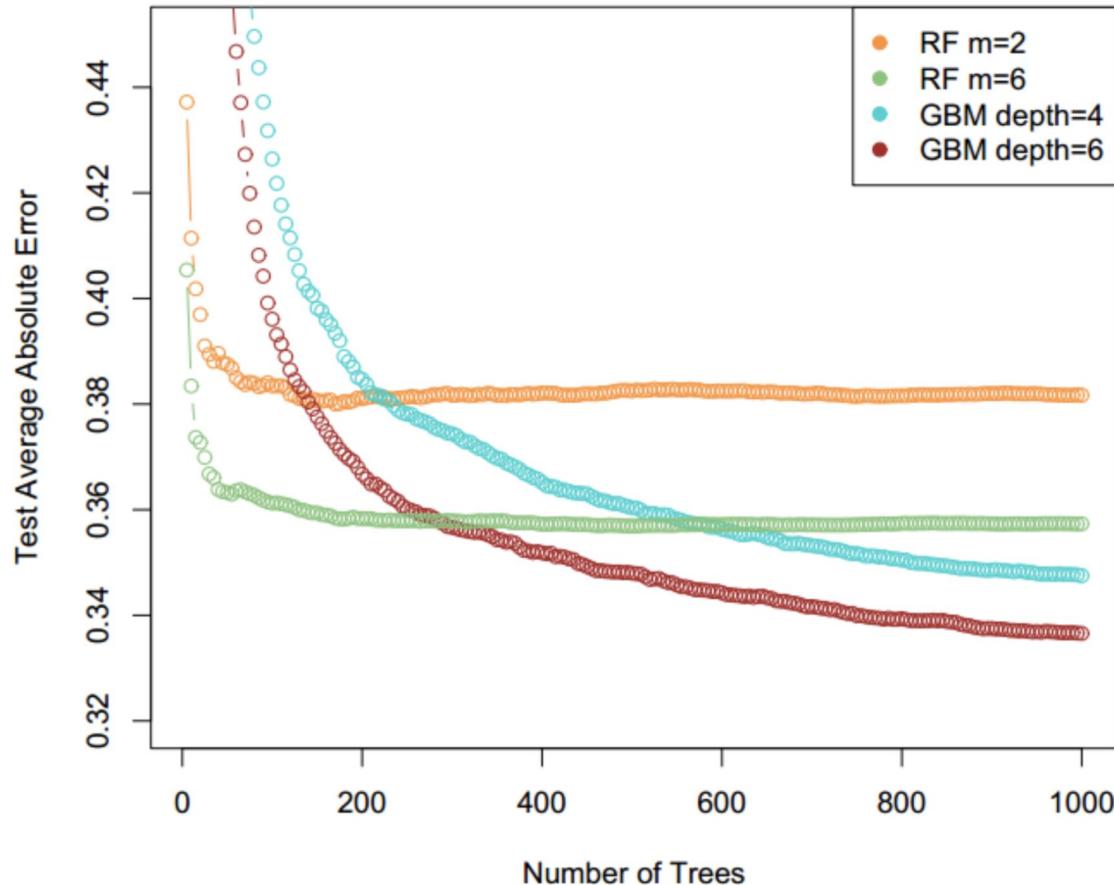
Example by ODS; source:  
<https://habr.com/ru/company/ods/blog/327250/>

## Spam Data





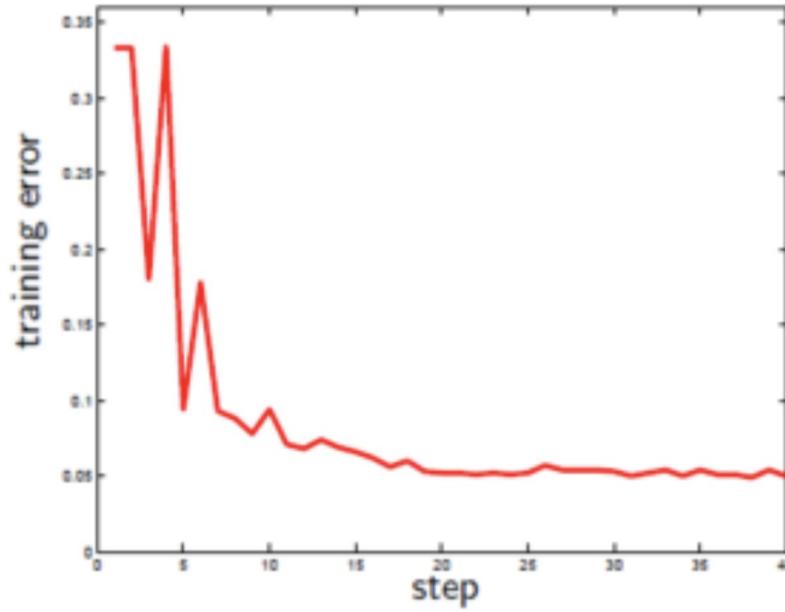
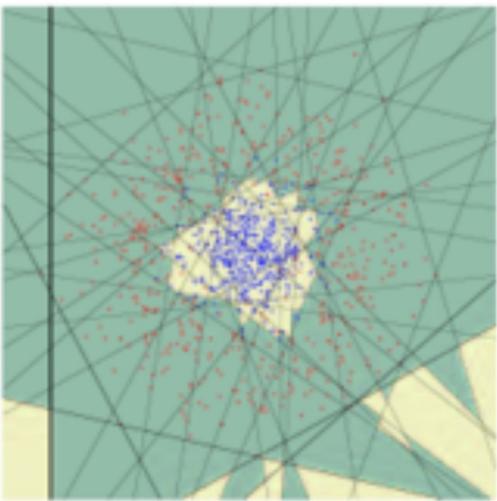
## California Housing Data



# 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
3. Blending
4. Stacking

# Thanks for attention!

Questions?



girafe  
ai

