Machine Learning
Lecture 6: Gradient boosting

Young && Yandex

girafe ai

Radoslav Neychev

Outline

01 Boosting intuitions

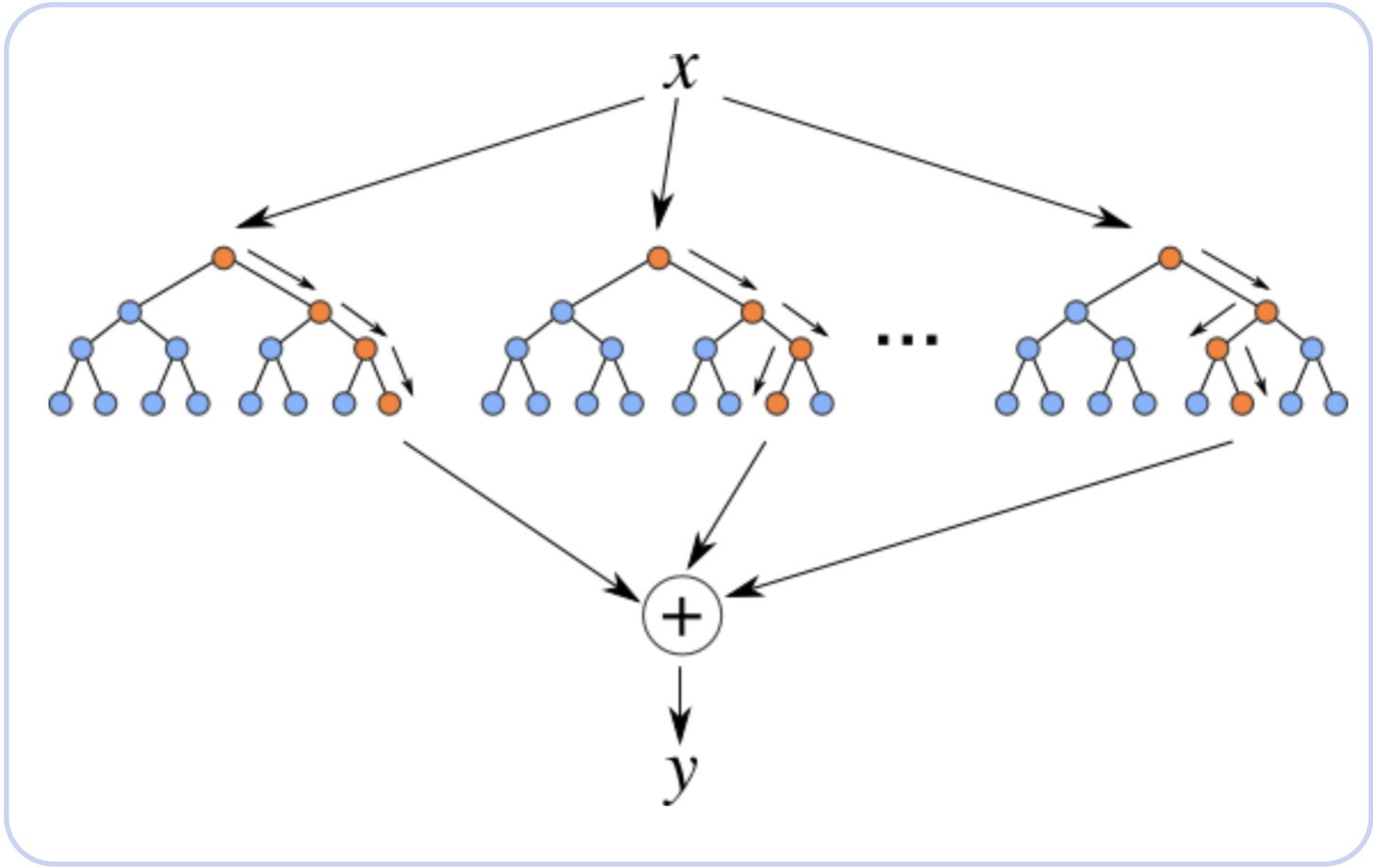
02 Gradient boosting

03 Blending

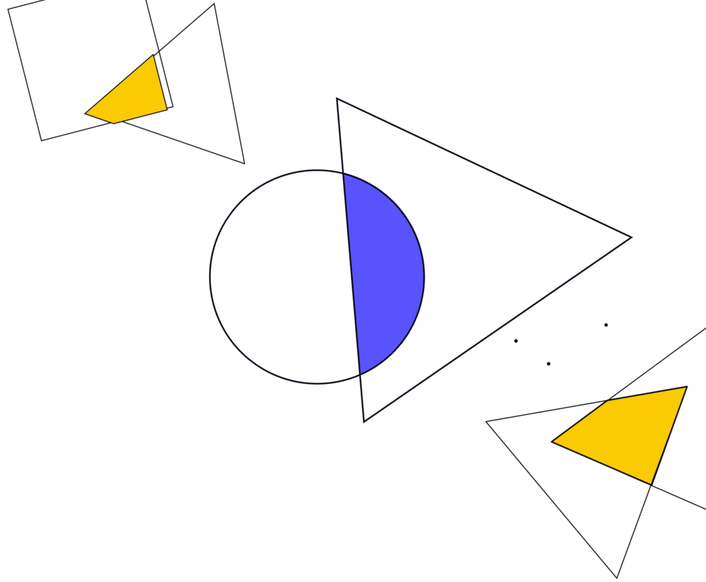
04 Stacking

Random Forest

Bagging + RSM = Random Forest



Random Forest



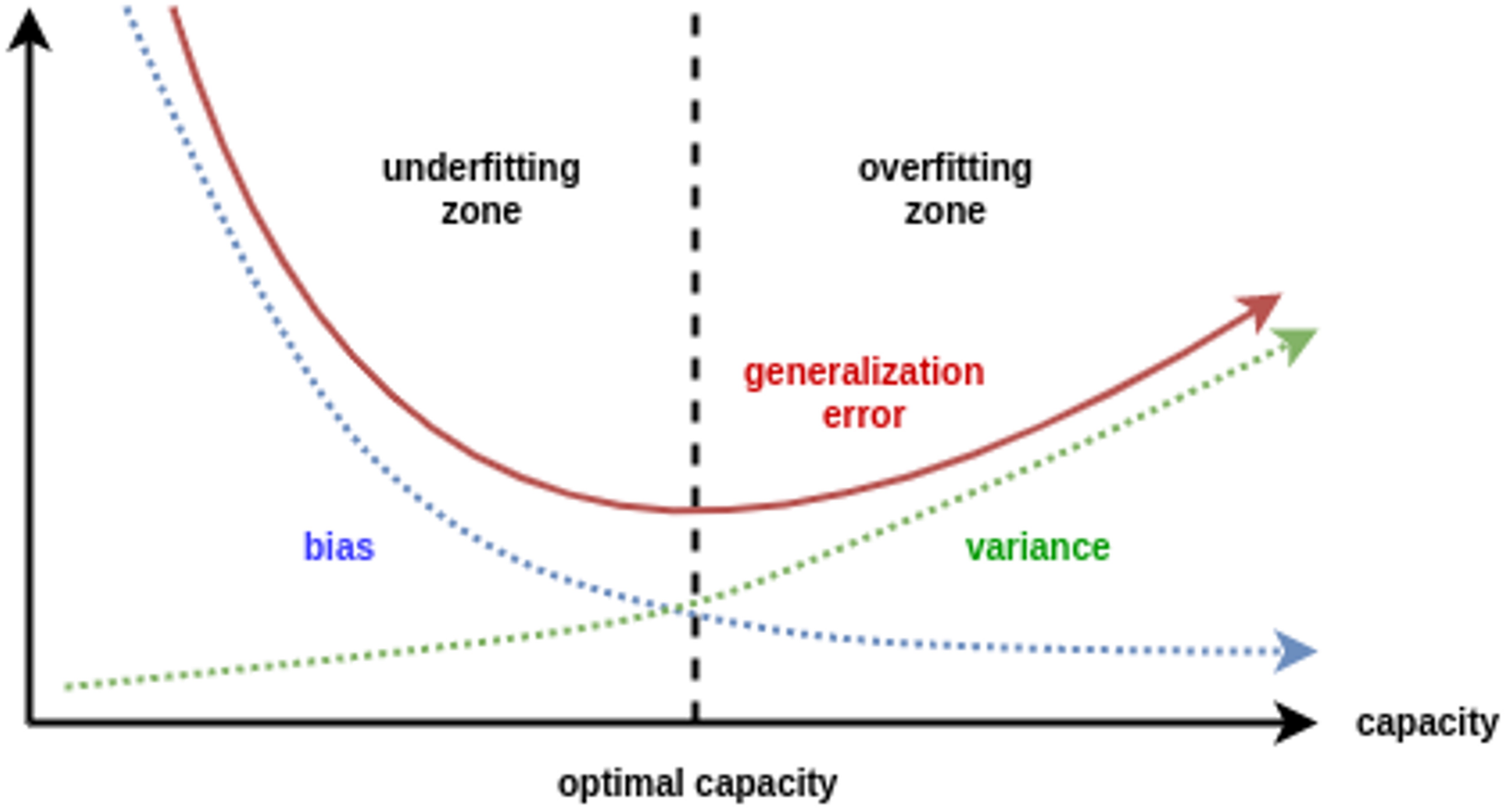
01 One of the greatest “universal” models

02 There are some modifications: Extremely Randomized Trees, Isolation Forest, etc.

03 Allows to use train data for validation: OOB

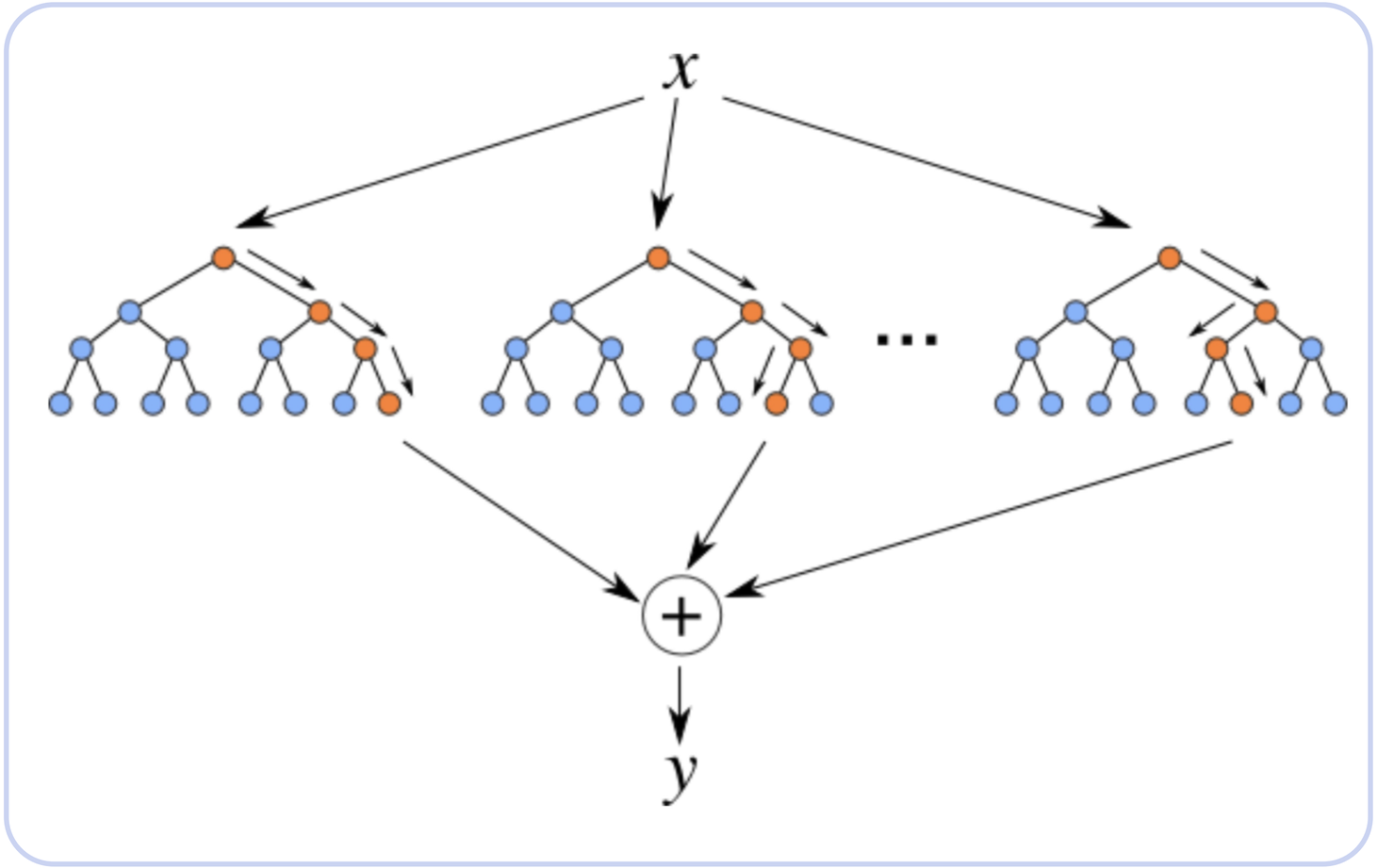
$$\text{OOB} = \sum_{i=1}^{\ell} L\left(y^{(i)}, \frac{1}{\sum_{n=1}^{N}[\boldsymbol{x}^{(i)} \notin \boldsymbol{X}_n]} \sum_{n=1}^{N}[\boldsymbol{x}^{(i)} \notin \boldsymbol{X}_n] b_n(\boldsymbol{x}^{(i)})\right)$$

Bias-variance tradeoff



Random Forest

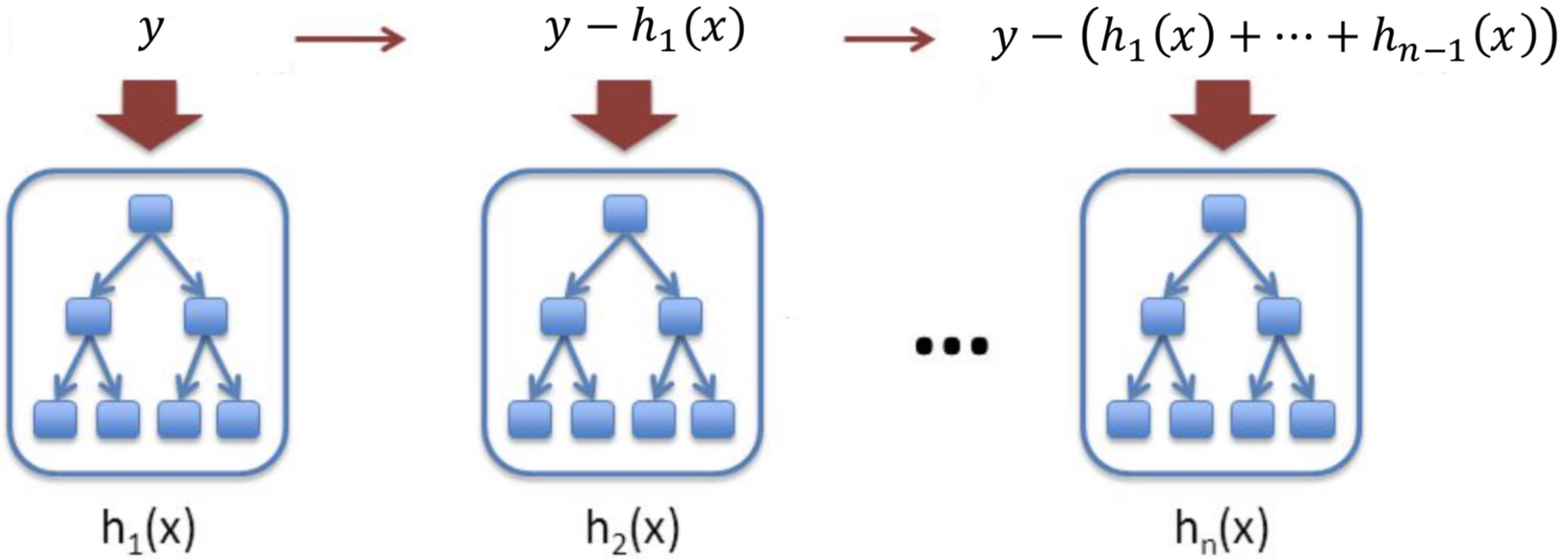
Is Random Forest decreasing bias or variance by building the trees ensemble?



Boosting

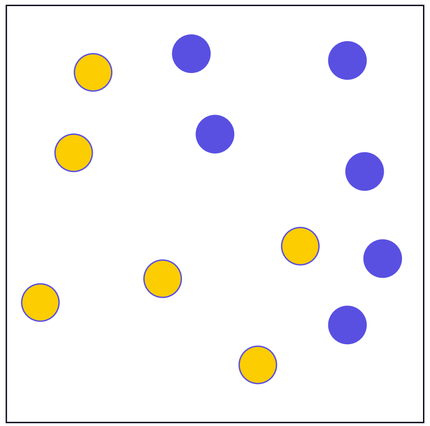
Boosting

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



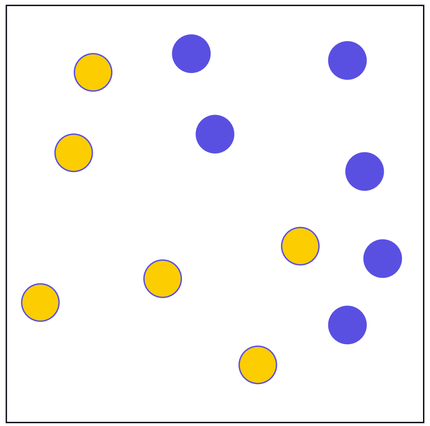
Boosting: intuition

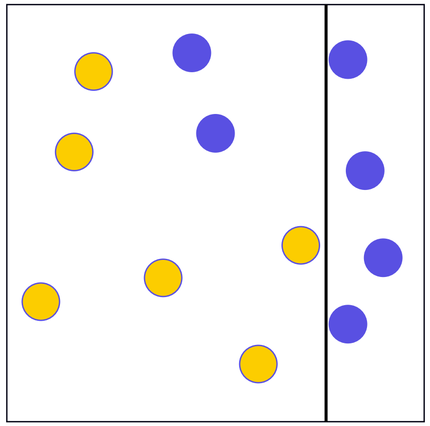
Binary classification
Use decision stumps.

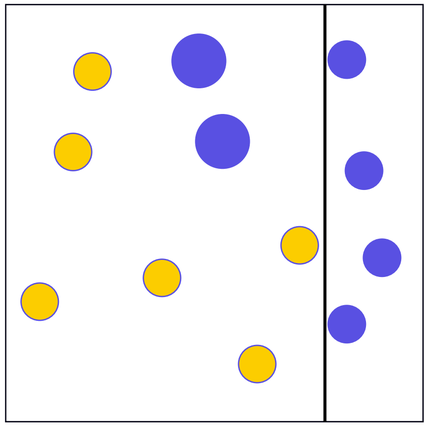


Boosting: intuition

Binary classification
Use decision stumps.

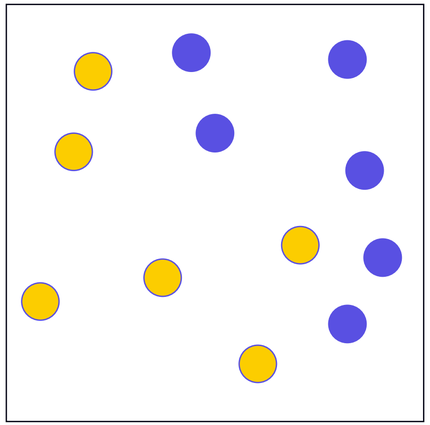


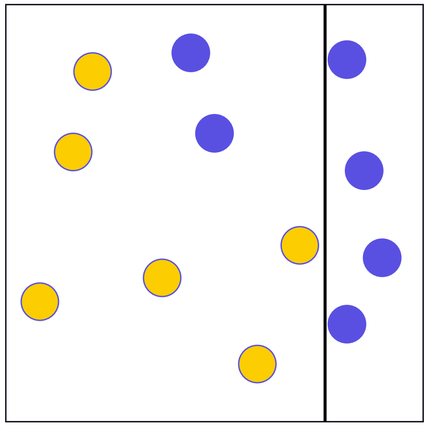


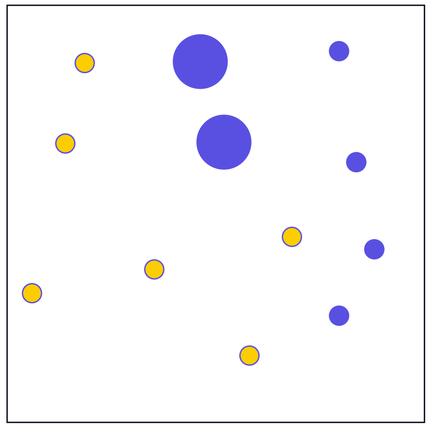


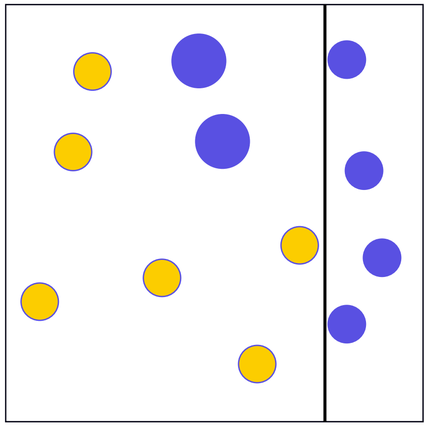
Boosting: intuition

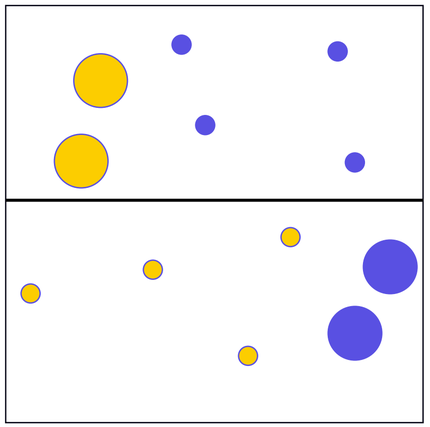
Binary classification
Use decision stumps.





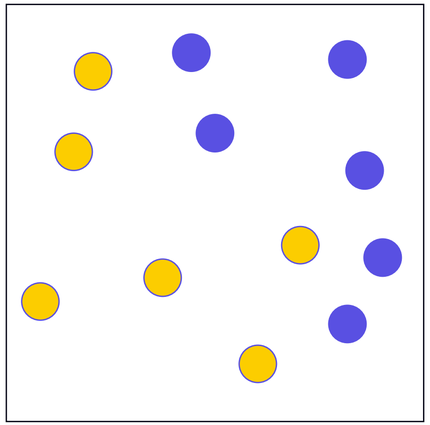


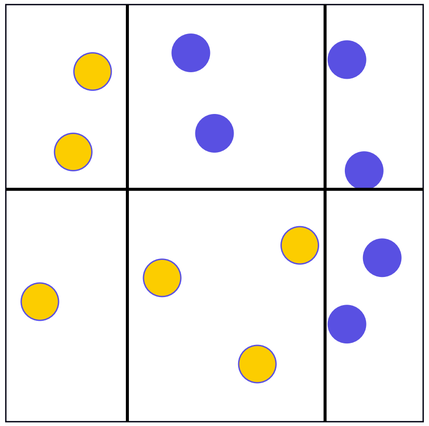


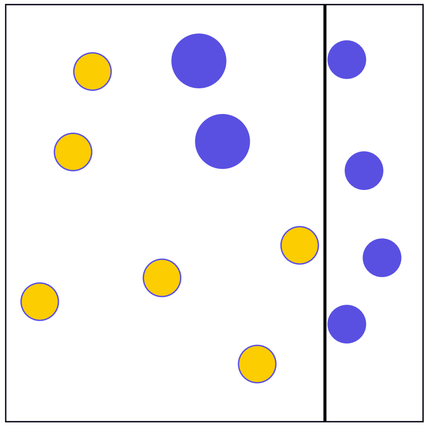


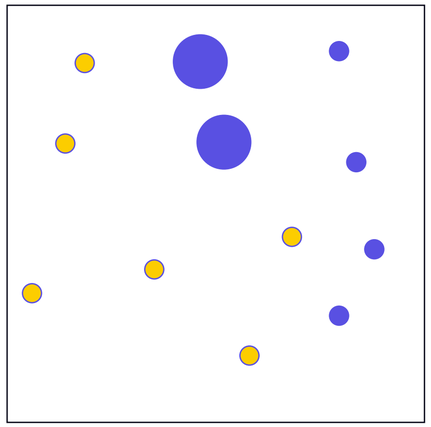
Boosting: intuition

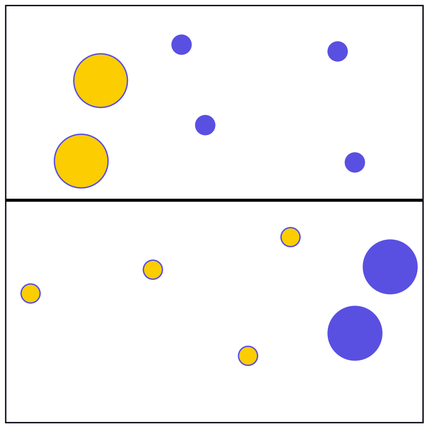
Binary classification
Use decision stumps.

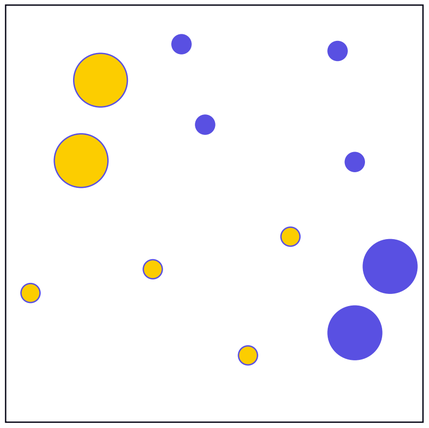


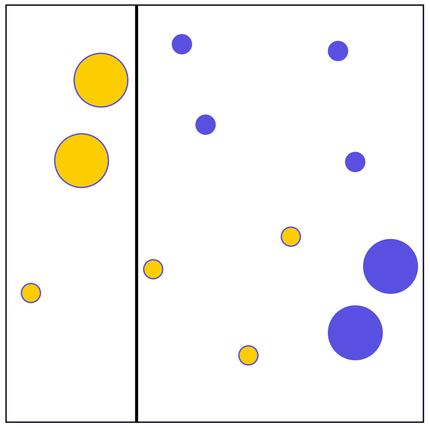






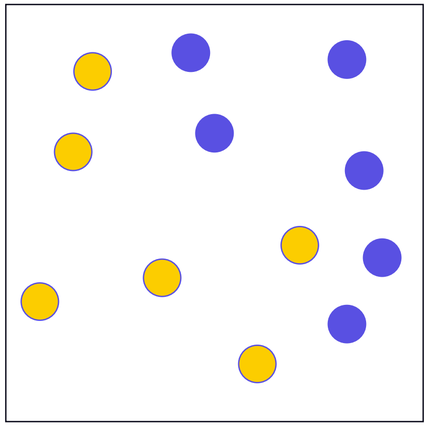


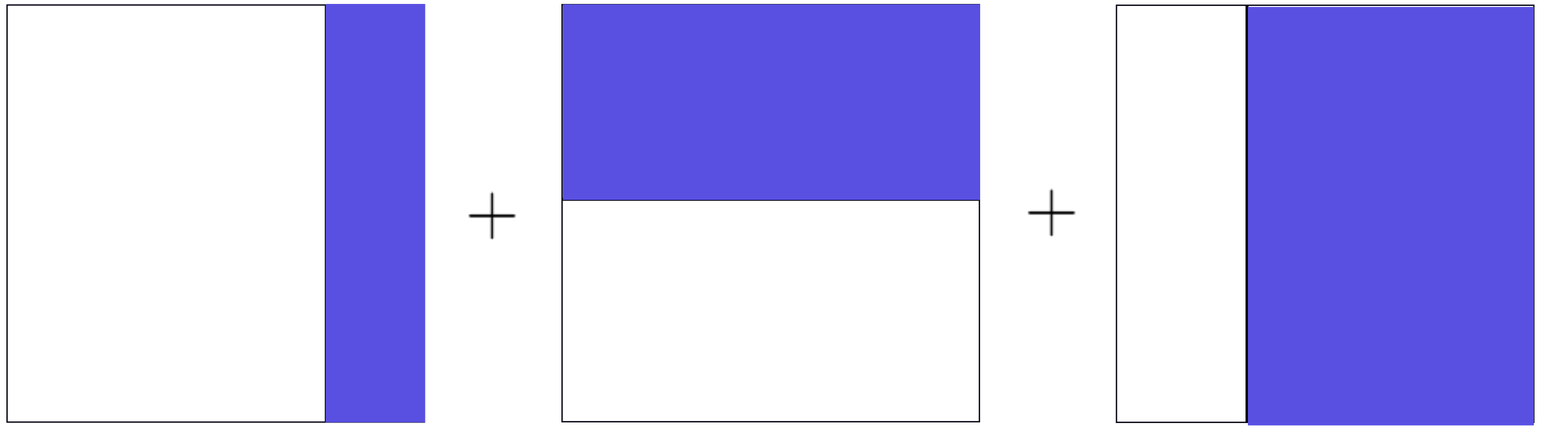




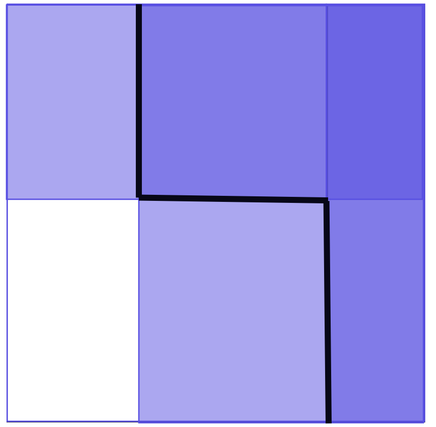
Boosting: intuition

Binary classification
Use decision stumps.

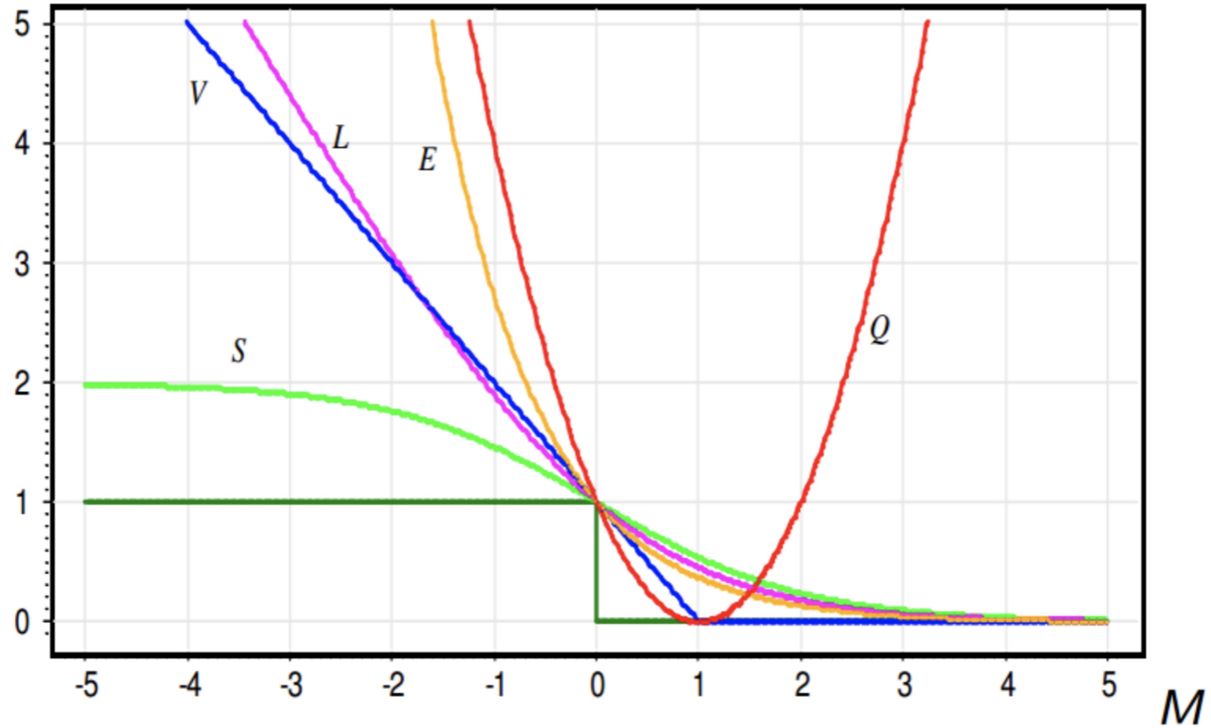




$$\hat{f}_T(x) = \sum_{t=1}^{T} \rho_t h_t(x) =$$



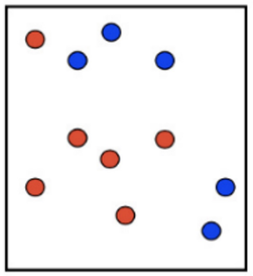
Recap: loss functions for classification



square loss	$Q(M) = (1 - M)^2$
hinge loss	$V(M) = (1 - M)_+$
savage loss	$S(M) = 2(1 + e^M)^{-1}$
logistic loss	$L(M) = \log(1 + e^{-M})$
exponential loss	$E(M) = e^{-M}$

Loss functions for classification

Boosting: AdaBoost



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

$$L(y^{(i)}, \hat{f}_T(\boldsymbol{x}^{(i)})) = \exp\left(-y^{(i)} \hat{f}_T(\boldsymbol{x}^{(i)})\right) = \exp\left(-y^{(i)} \sum_{t=1}^{T} \rho_t h_t(\boldsymbol{x}^{(i)})\right.$$

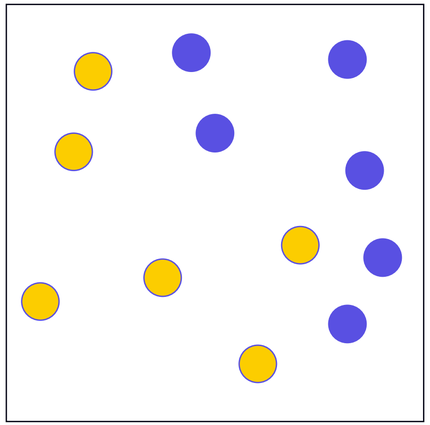
const on step T

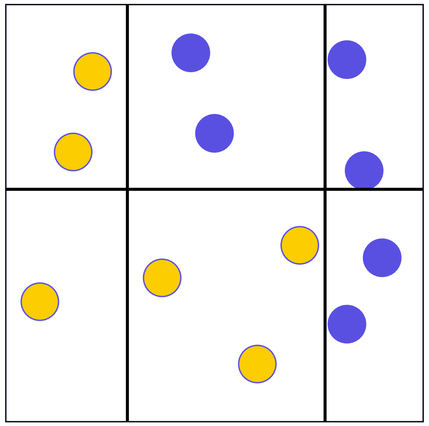
$$= \exp\left(-y^{(i)} \sum_{t=1}^{T-1} \rho_t h_t(\boldsymbol{x}^{(i)})\right) \cdot \exp\left(-y^{(i)} \rho_T h_T(\boldsymbol{x}^{(i)}\right.$$

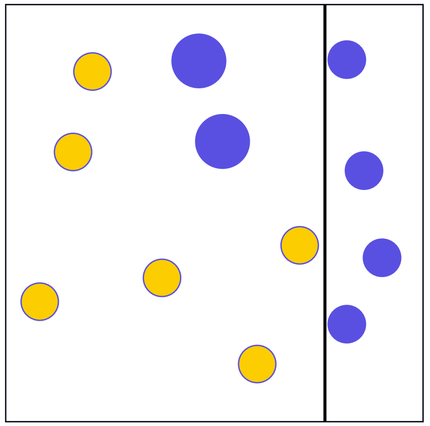
$$= w^{(i)} \cdot \exp\left(-y^{(i)} \rho_T h_T(\boldsymbol{x}^{(i)})\right)$$

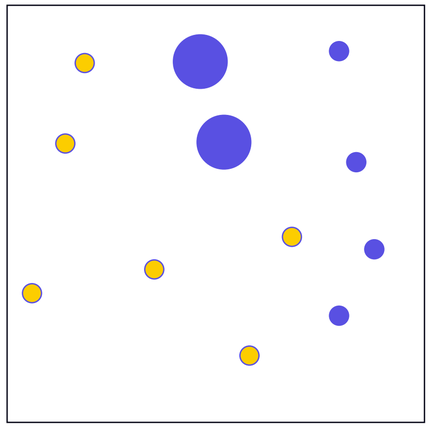
Boosting: intuition

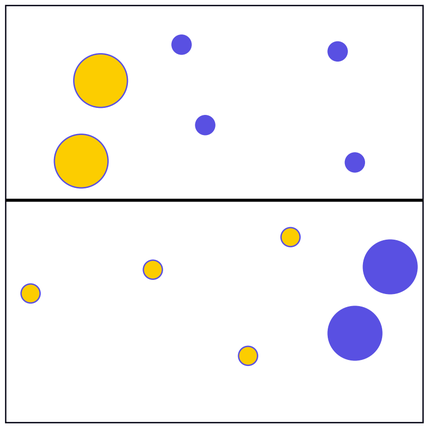
Binary classification
Use decision stumps.

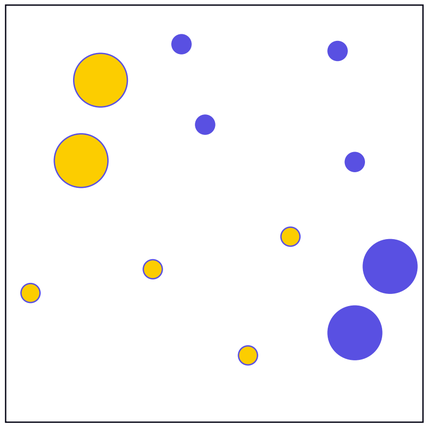


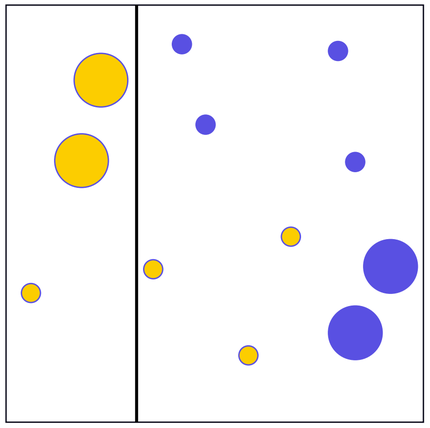




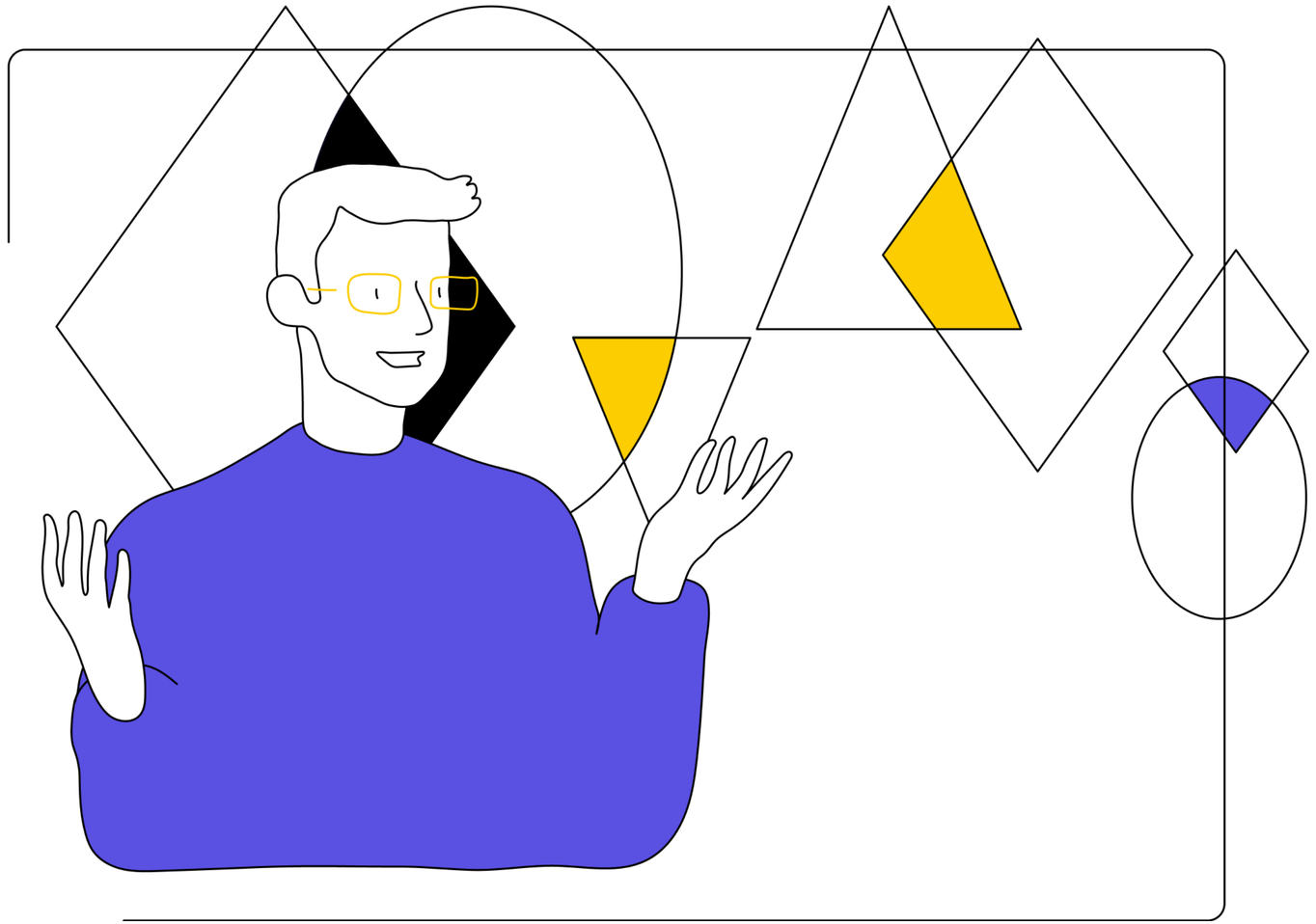




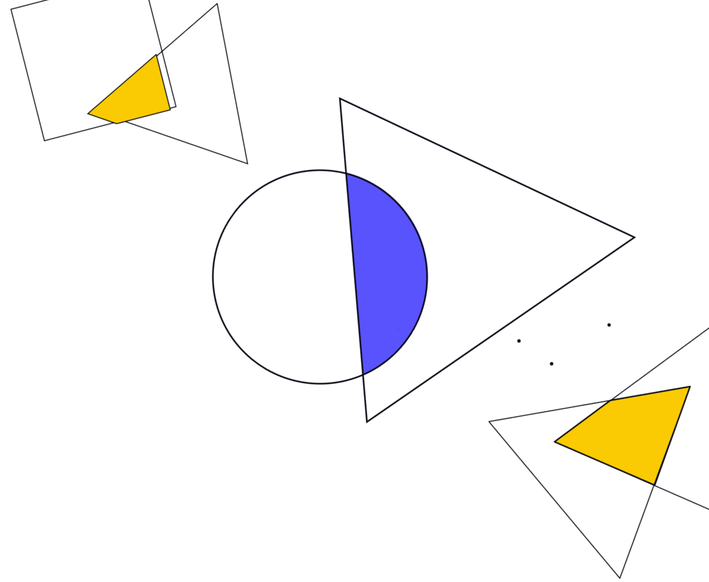




Gradient boosting



Gradient boosting: theory



Denote dataset
and Loss function

$$\{(\boldsymbol{x}^{(i)}, y^{(i)})\}_{i=1}^{n}$$

Optimal model:

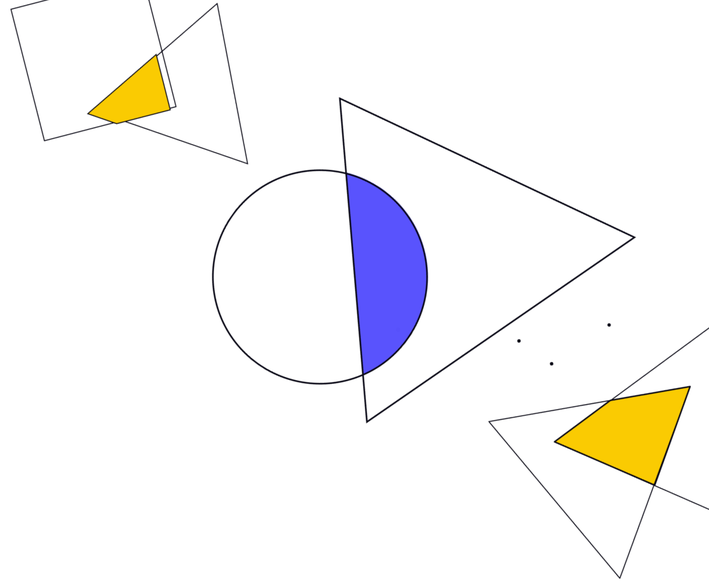
$$\hat{f}(\boldsymbol{x}) = \arg\min_{f(\boldsymbol{x})} L(y, f(\boldsymbol{x})) = \arg\min_{f(\boldsymbol{x})} \mathbb{E}_{\boldsymbol{x},y}[L(y, f(\boldsymbol{x}))]$$

Let it be from parametric family

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

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

Gradient boosting: theory



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

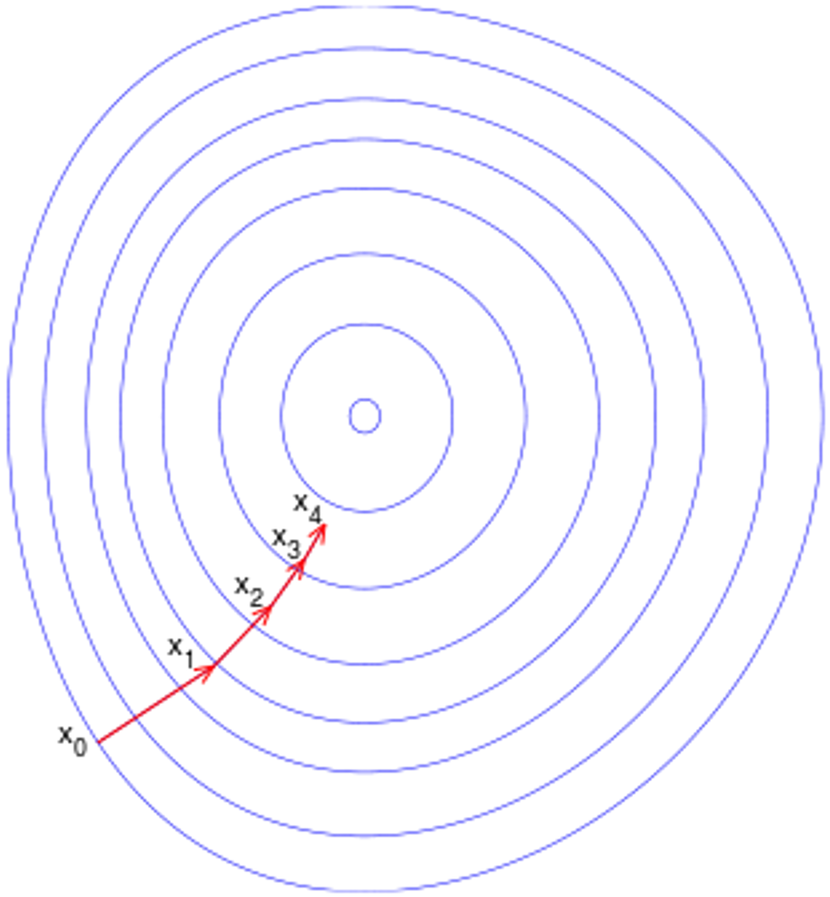
$$\hat{f}_{T-1}(\boldsymbol{x}) = \sum_{t=0}^{T-1} g_t(\boldsymbol{x}),$$

$$(\rho_T, \boldsymbol{\theta}_T) = \arg\min_{\rho,\boldsymbol{\theta}} \mathbb{E}_{\boldsymbol{x},y} \left[L\left(y, \hat{f}_{T-1}(\boldsymbol{x}) + \rho_T \cdot h(\boldsymbol{x}, \boldsymbol{\theta}_T)\right)\right],$$

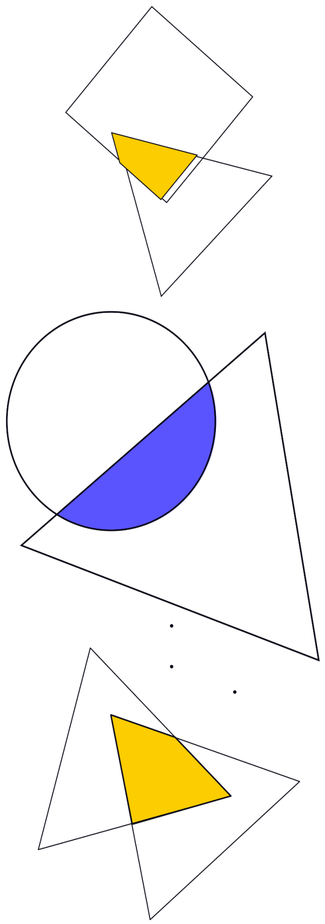
$$g_t(\boldsymbol{x}) = \rho_t \cdot h(\boldsymbol{x}, \boldsymbol{\theta}_t)$$

Gradient boosting: theory

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



Gradient boosting: theory



$$\hat{f}_{T-1}(\boldsymbol{x}) = \sum_{t=0}^{T-1} g_t(\boldsymbol{x}),$$

$$r_t^{(i)} = - \left[\frac{\partial L(y^{(i)}, f(\boldsymbol{x}^{(i)}))}{\partial f(\boldsymbol{x}^{(i)})} \right]_{f(\boldsymbol{x})=\hat{f}_T(\boldsymbol{x})}, \quad \text{for } i = 1, \ldots, n,$$

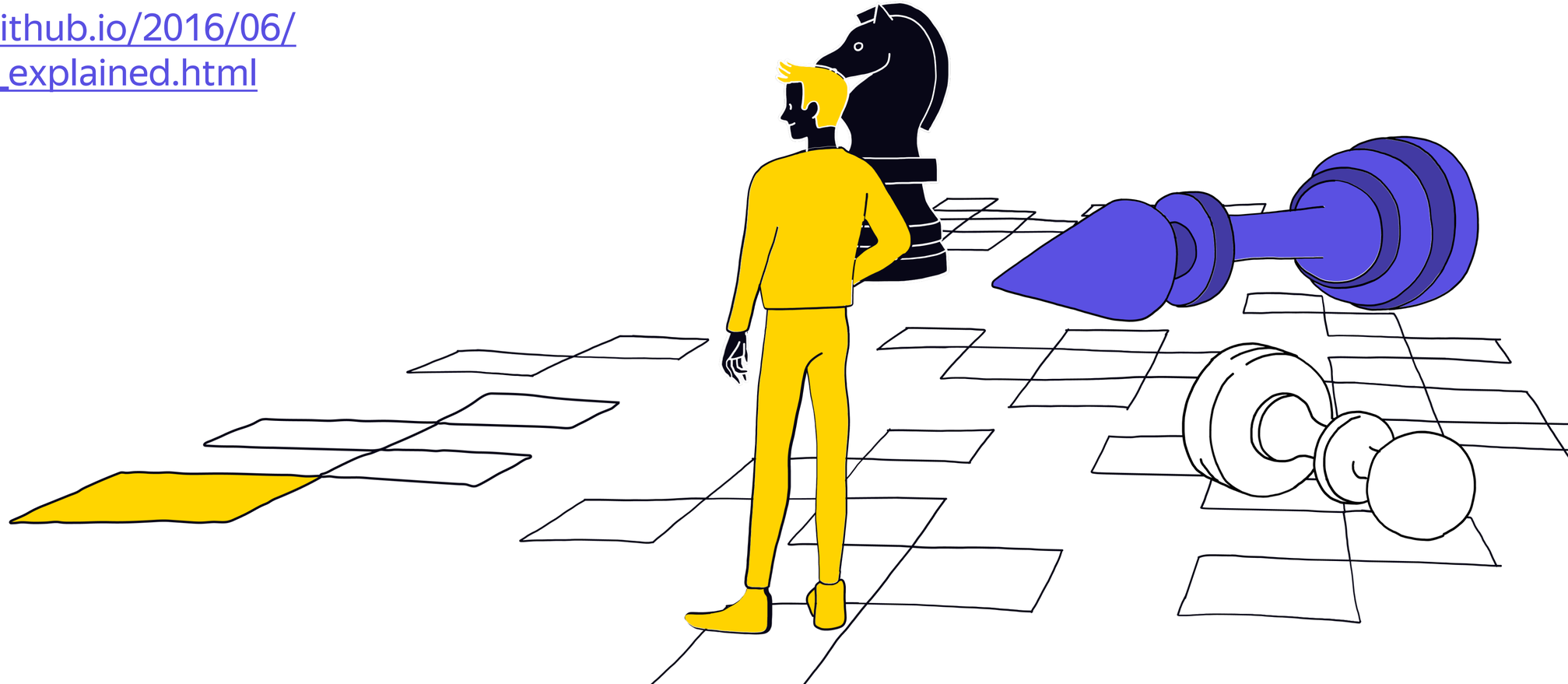
$$\boldsymbol{\theta}_T = \arg\min_{\boldsymbol{\theta}} \sum_{i=1}^{n} \left(r_t^{(i)} - h(\boldsymbol{x}^{(i)}, \boldsymbol{\theta}) \right)^2,$$

$$\rho_t = \arg\min_{\rho} \sum_{i=1}^{n} L \left(y^{(i)}, \hat{f}_{t-1}(\boldsymbol{x}^{(i)}) + \rho \cdot h(\boldsymbol{x}^{(i)}, \boldsymbol{\theta}_T) \right)$$

Gradient boosting: beautiful demo

Great demo:

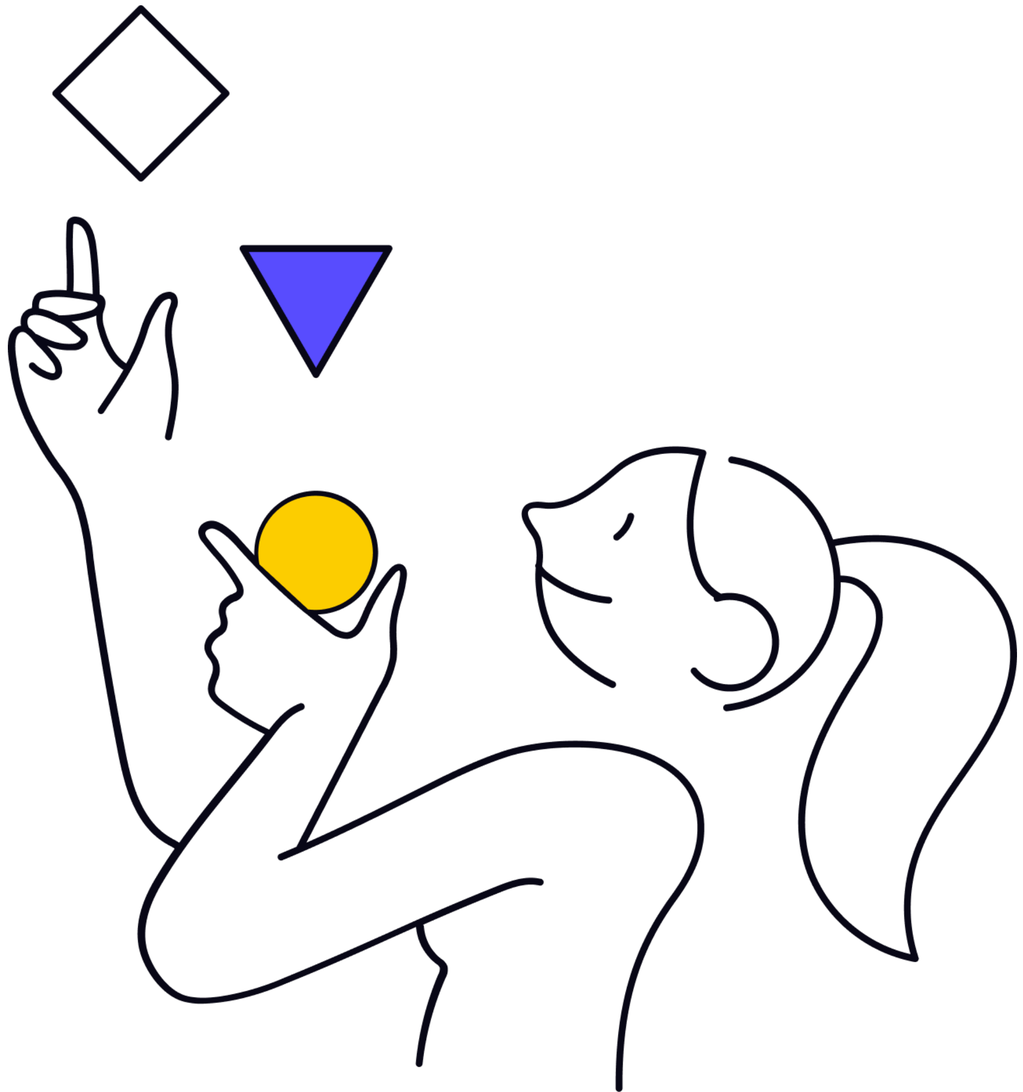
http://arogozhnikov.github.io/2016/06/24/gradient_boosting_explained.html



Gradient boosting

What we need:

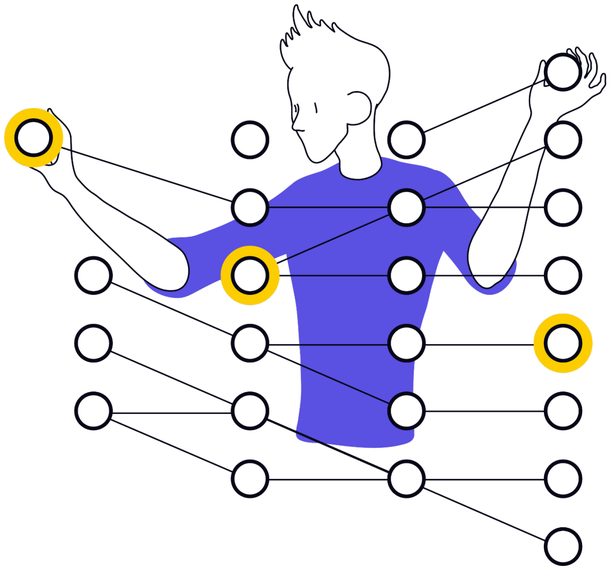
- Data.
- Loss function and its gradient.
- Family of algorithms (with constraints if necessary).
- Number of iterations M.
- Initial value (GBM by Friedman): constant.



Gradient boosting: example

What we need:

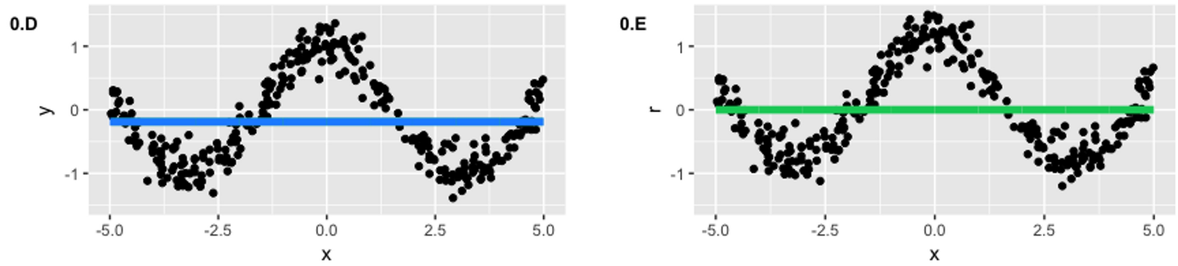
- Data: toy dataset —— $y = cos(x) + \epsilon, \epsilon \sim \mathcal{N}(0, \frac{1}{5}), x \in [-5, 5]$
- Loss function: MSE
- Family of algorithms: decision trees with depth 2
- Number of iterations M = 3
- Initial value: just mean valu



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

Gradient boosting: example

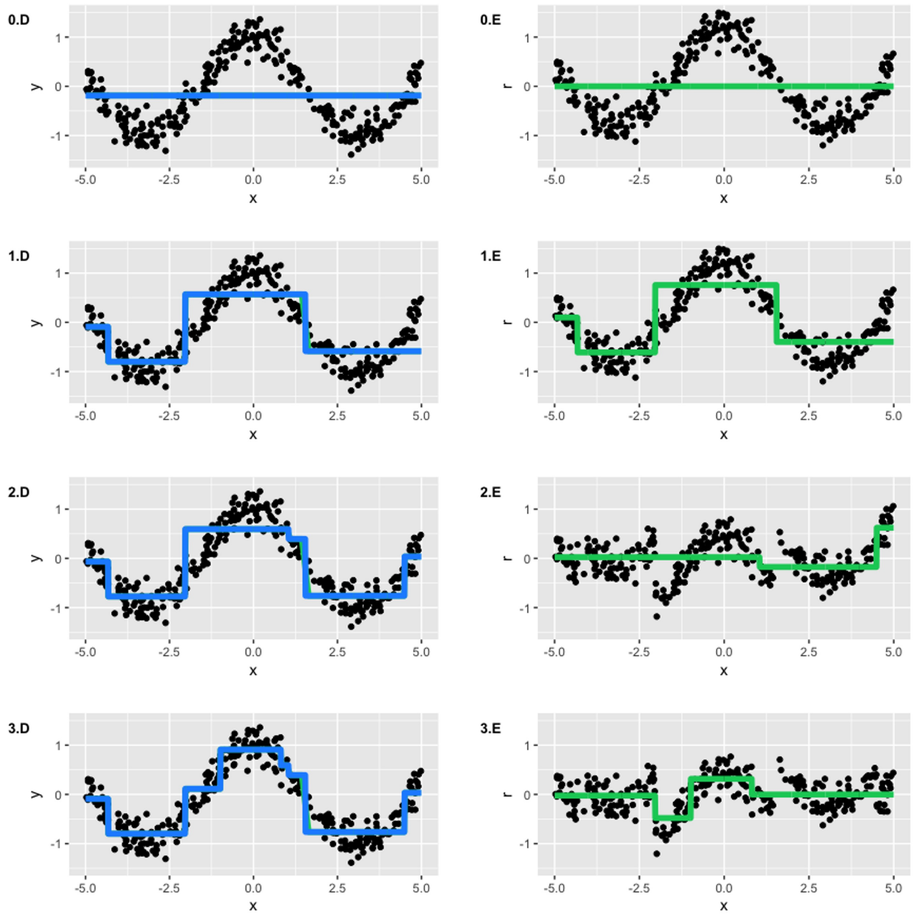
Left: full ensemble on each step.
Right: additional tree decisions.



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.

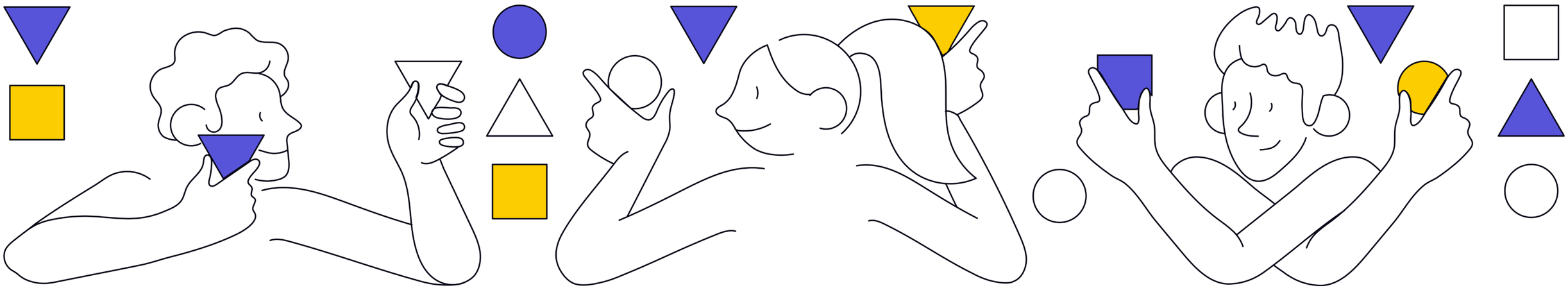


Example by ODS; source: https://habr.com/ru/company/ods/blog/327250/

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



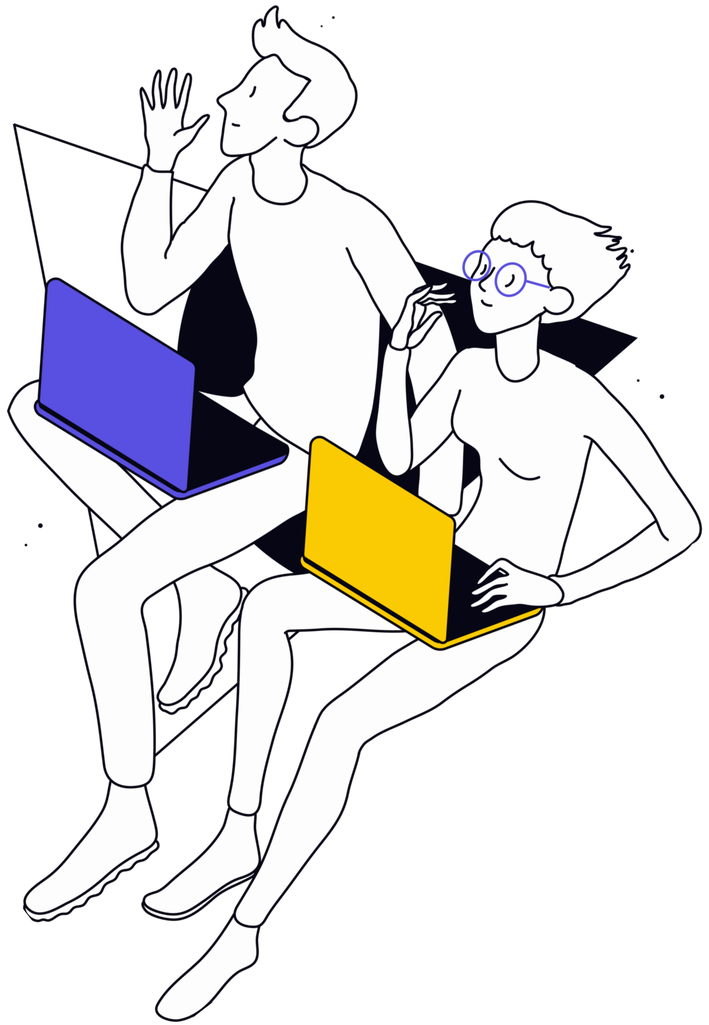
Revise

01 Boosting intuitions

02 Gradient boosting

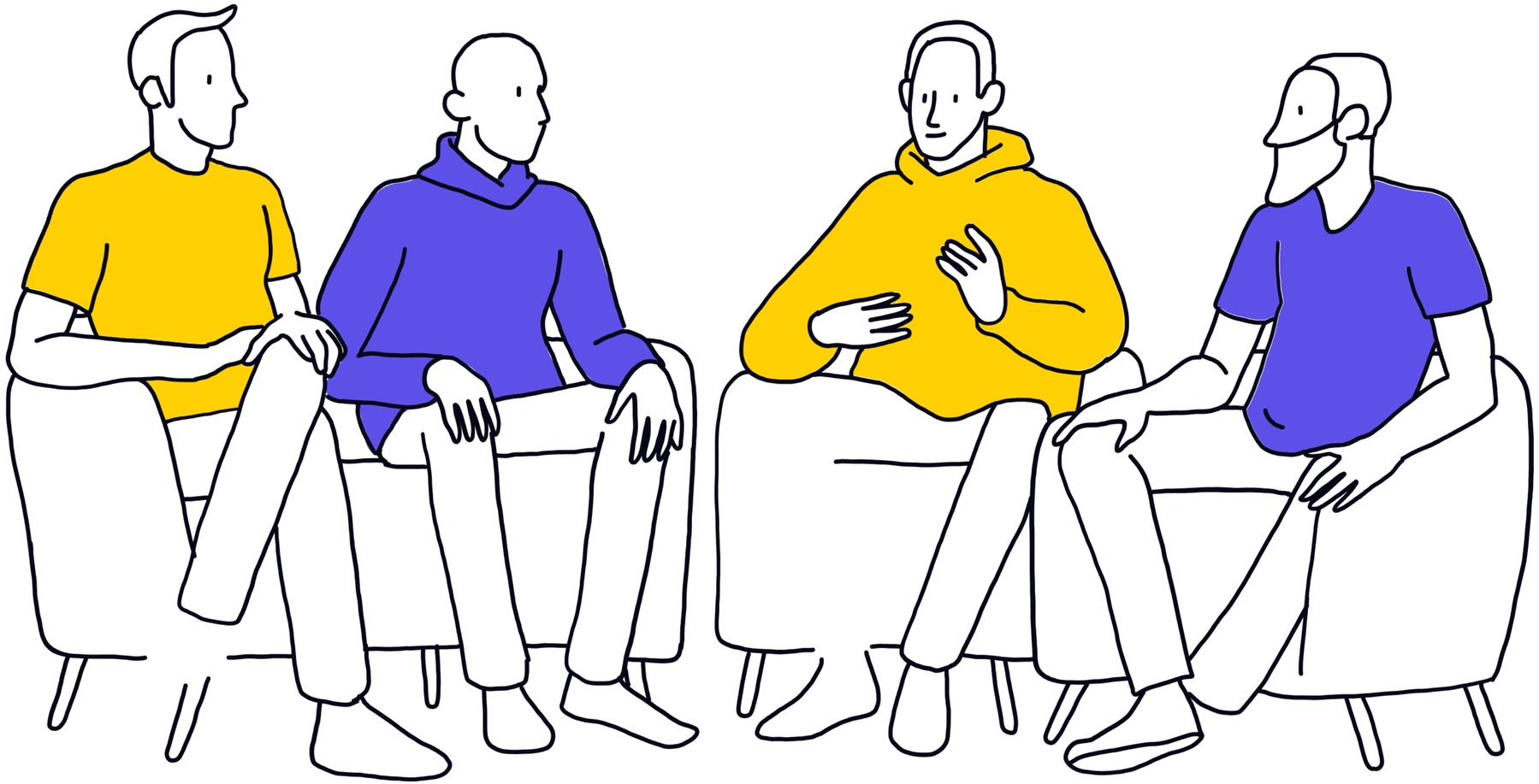
03 Blending

04 Stacking



Thanks for attention!

Questions?



Radoslav Neychev