

# Машинное обучение

Лекция 04. Ансамбли решающих деревьев

Драль Алексей

<https://www.linkedin.com/in/alexey-dral>

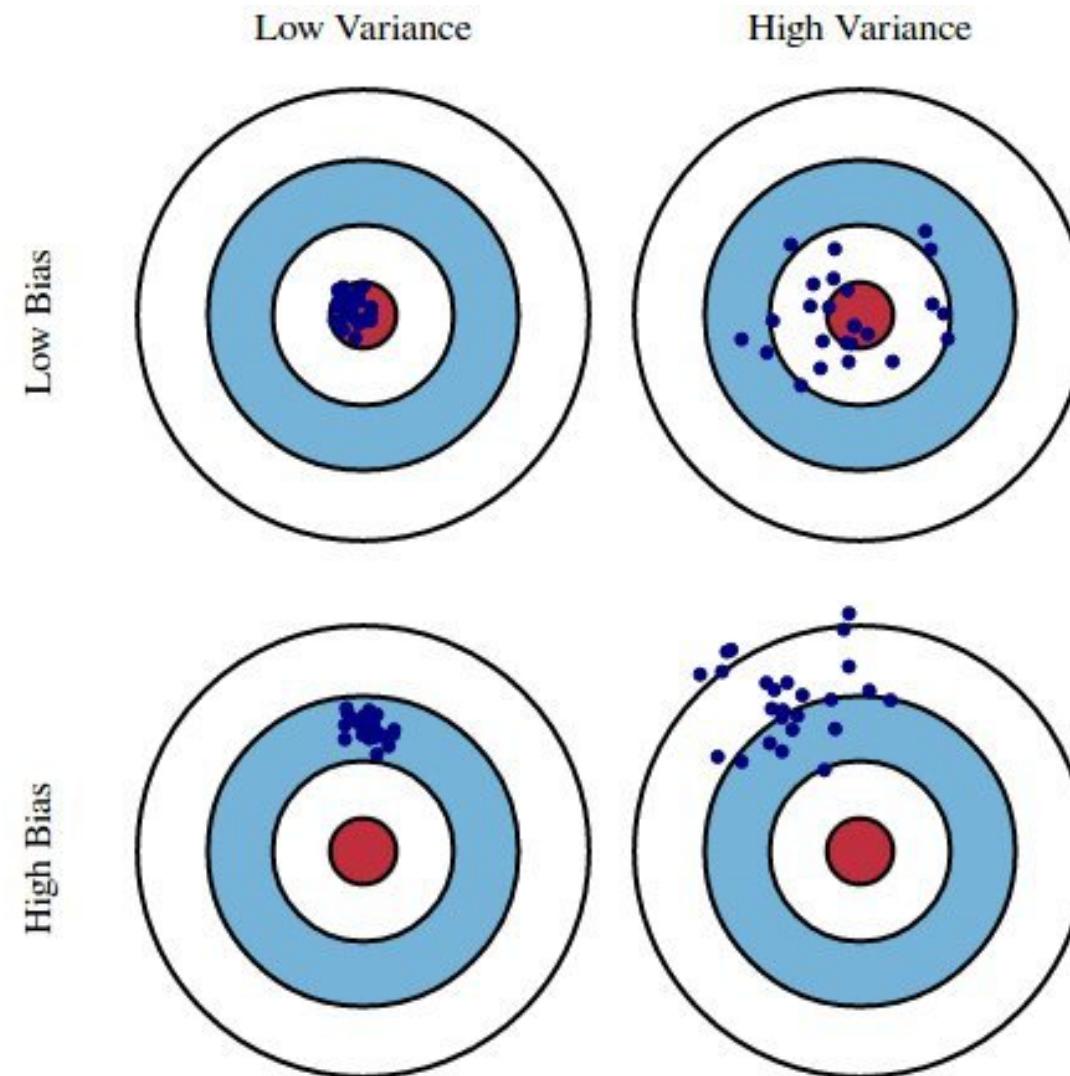
06.03.2018, Москва, ФИВТ МФТИ

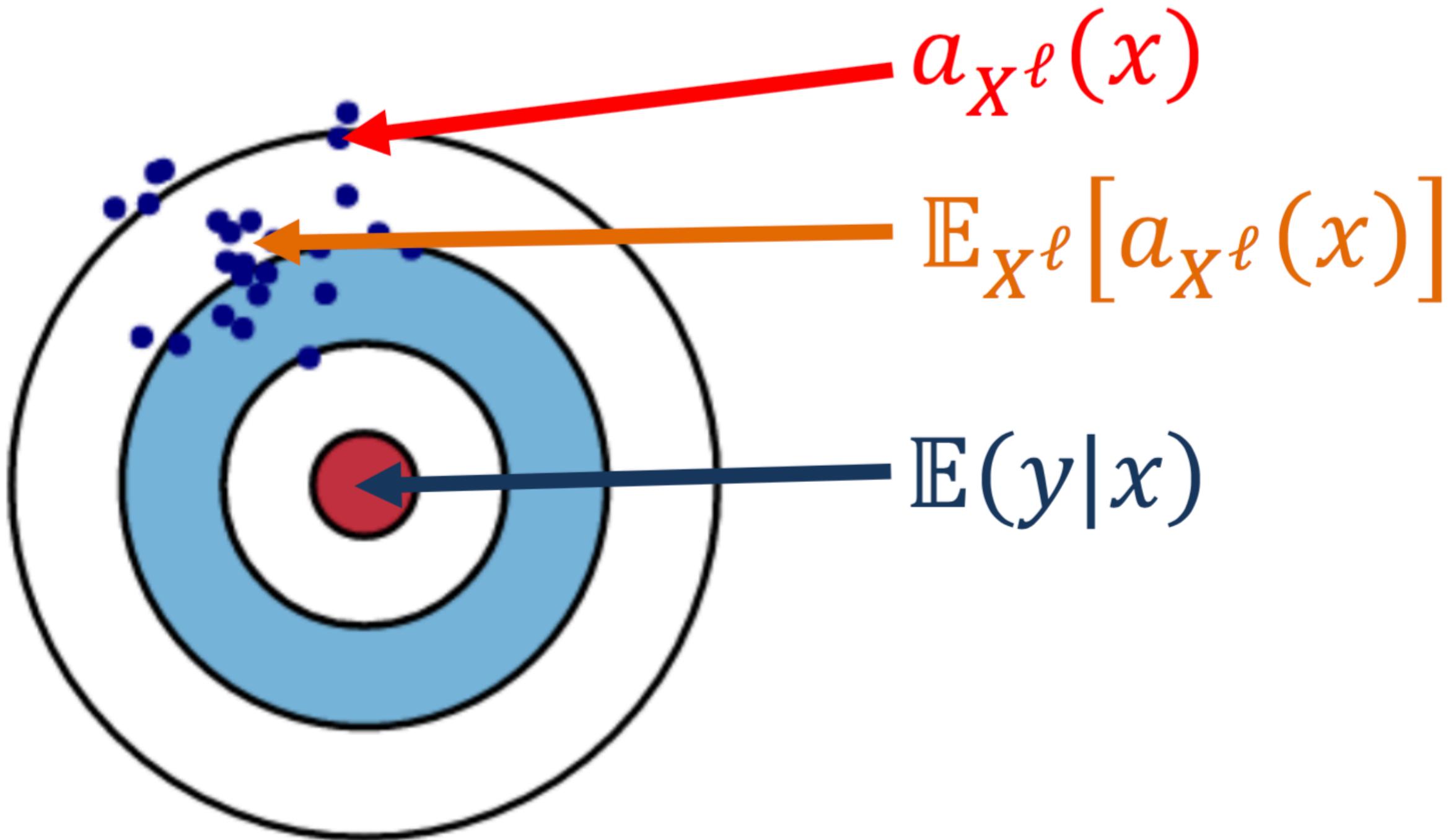
# План

Ансамбли решающих деревьев (Ensemble of Decision Trees):

- Bias-Variance Trade-off
- Бэггинг (Bagging = Bootstrap aggregating)
- Случайный лес (Random Forest)
- Бустинг (Boosting), AdaBoost и обобщения

# Bias-variance trade-off





# Bias-variance-noise decomposition

$$\begin{aligned}\mathbb{E}_{x,y} \mathbb{E}_{X^\ell} \left( y - a_{X^\ell}(x) \right)^2 = \\ & \underbrace{\mathbb{E}_{x,y} \left( y - \mathbb{E}(y|x) \right)^2}_{\text{Noise}^2} + \\ & + \underbrace{\mathbb{E}_{x,y} \left( \mathbb{E}(y|x) - \mathbb{E}_{X^\ell} [a_{X^\ell}(x)] \right)^2}_{\text{Bias}^2} + \\ & + \underbrace{\mathbb{E}_{x,y} \mathbb{E}_{X^\ell} \left( a_{X^\ell}(x) - \mathbb{E}_{X^\ell} [a_{X^\ell}(x)] \right)^2}_{\text{Variance}}\end{aligned}$$

# Q&A: проблемы обученного дерева

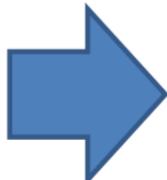


# Бэггинг (Bagging = Bootstrap aggregating)

Выборка из некоторого распределения:

№	значение
1	
2	
3	
N	

Хотим вычислить какую-то величину  $X$  по данным наблюдениями.

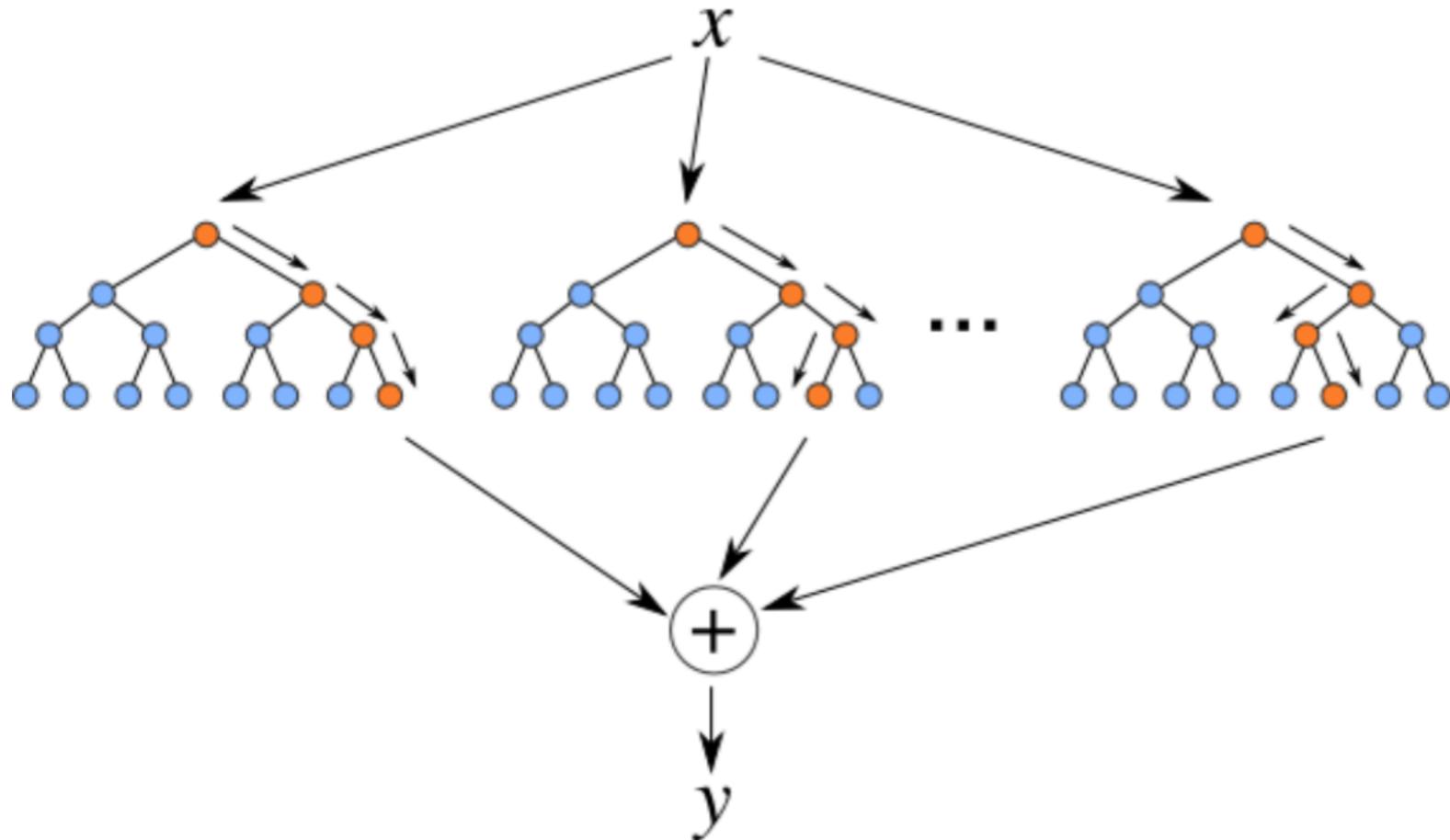


Было бы здорово вычислить  $X$  на многих выборках из распределения, а потом усреднить, но их у нас нет

Решение:

1. Выбираем наугад одно наблюдение из имеющихся.
2. Повторяем пункт 1 столько раз, сколько у нас есть наблюдений. При этом некоторые из них мы можем выбрать повторно
3. Считаем интересующие нас величины по новой выборке. Запоминаем результат.
4. Повторяем пункты 1-3 много раз и усредняем

# Бэггинг (Bagging = Bootstrap aggregating)



# Бэггинг (Bagging = Bootstrap aggregating)

Bagging означает i.d trees, поэтому:

1.  $Bias(\text{bagged model}) = Bias(\text{tree})$
2.  $Var(\text{bagged model}) = \rho\sigma^2 + \frac{1-\rho}{B}\sigma^2, Var(\text{tree}) = \sigma^2,$   
 $cor(t_i, t_j) = \rho$

Задание:

1. Доказать утверждения (1) и (2)
2. Предложить улучшения модели.

Whiteboard Time!



# Улучшения / модификации бэггинга

- Random Subspace Method (RSM) – выбор признаков, а не объектов
- Pasting\* – выбор объектов без возвращения
- Random Forests
- Extremely Randomized Trees

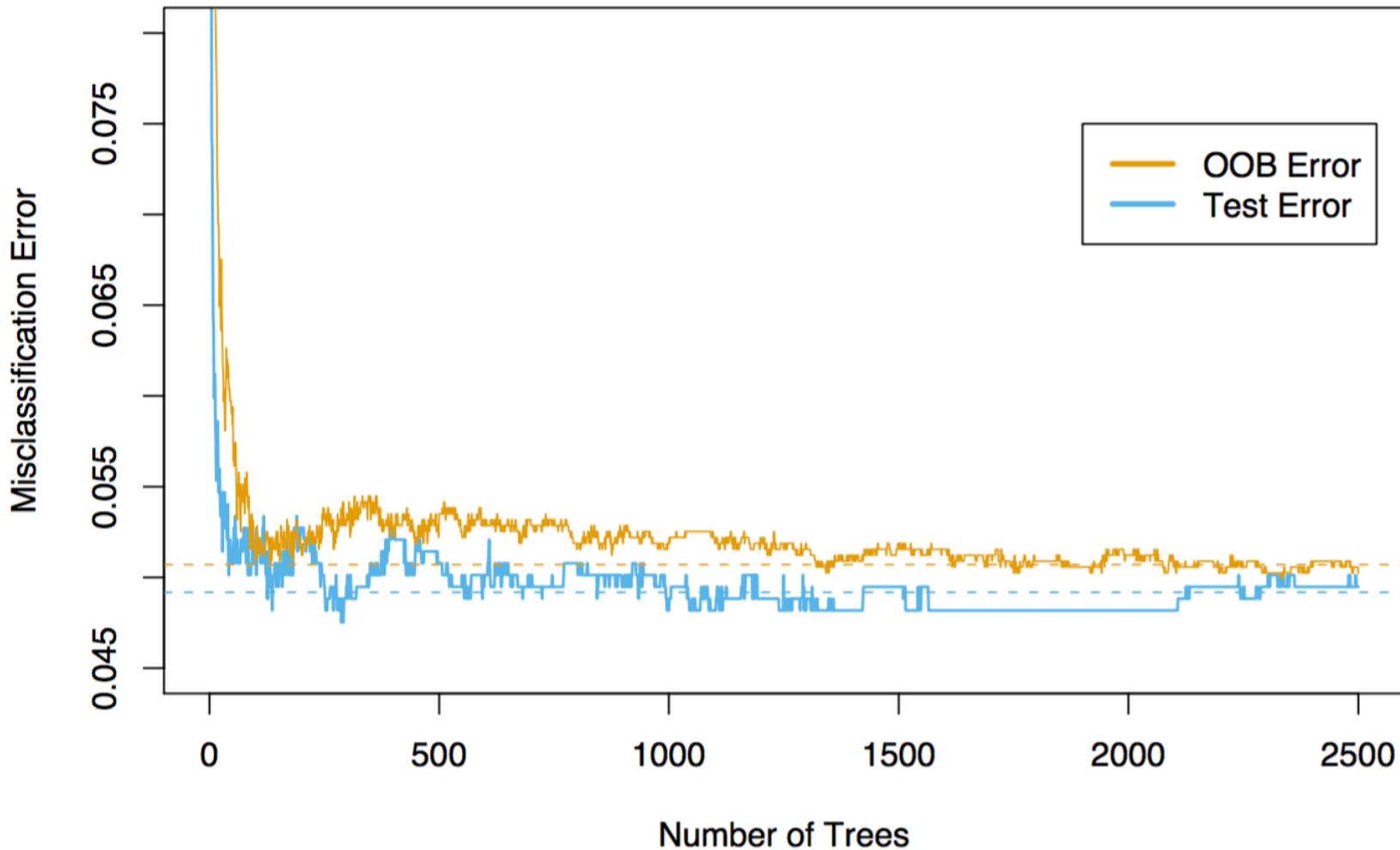
\* Friedman, Hall (2007) – показали, что sub-sampling (он же pasting) эффективная альтернатива bagging. Они доказали эквивалентность в рамках bias/variance tradeoff бэггингу, если использовать  $N/2$  элементов выборки, в то время как выбор меньшего числа объектов уменьшает variance еще больше.

# Случайный лес (Random Forest)

- Bagging над деревьями
- Случайные разбиение в узлах деревьев: в **каждом узле** дерева выбирается  $m$  случайных признаков и из них выбирается лучший предикат для разбиения выборки
- Cross-Validation из коробки: Out-of-Bag (OOB) оценка

Random Forests, Brieman (2001)

# Случайный лес (Random Forest)



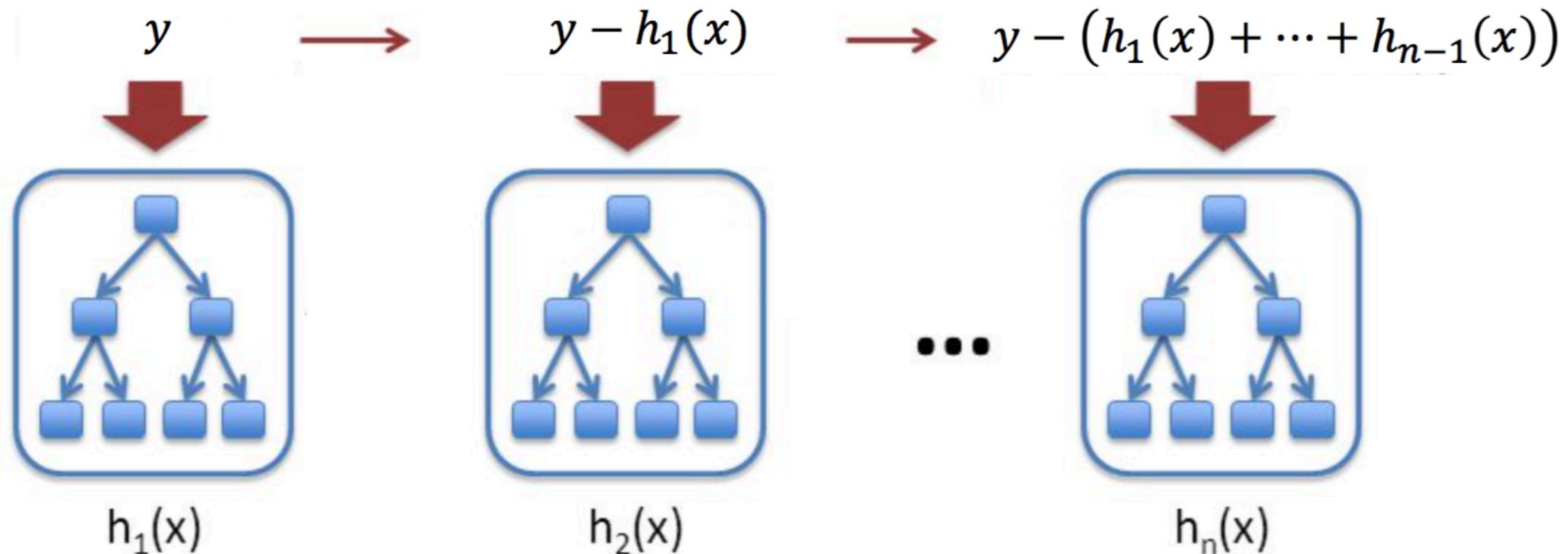
# Extremely randomized trees\*

- Усиление случайности выбора предикатов в Random Forest – выбирается  $m$  случайных признаков, а затем выбираются **случайный набор порогов** для каждого выбранного признака (уменьшение дисперсии за счет небольшого увеличения отклонения)
- - доступно в sklearn

P. Geurts, D. Ernst., and L. Wehenkel, “Extremely randomized trees”, Machine Learning, 63(1), 3-42, 2006

# Бустинг (Boosting), AdaBoost и обобщения

Идея Gradient Boosted Decision Trees (GBDT):



# Q&A: какие должны быть деревья?



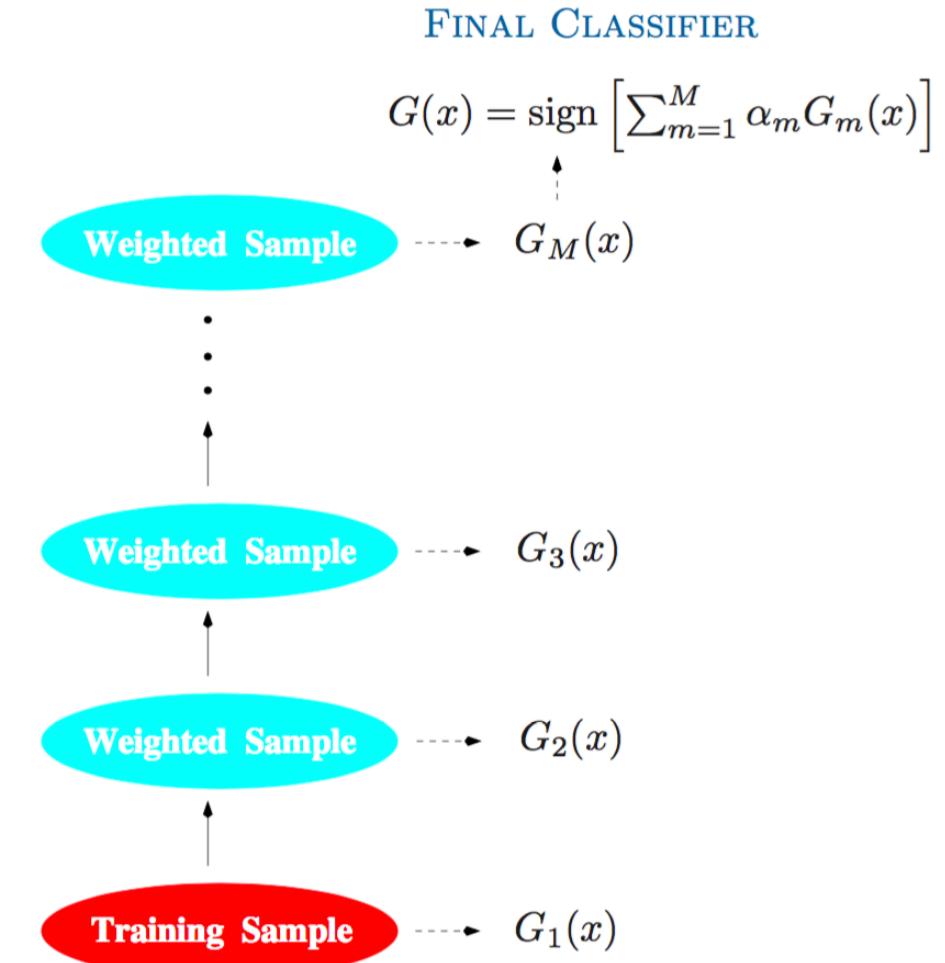
# Q&A: какие должны быть деревья?



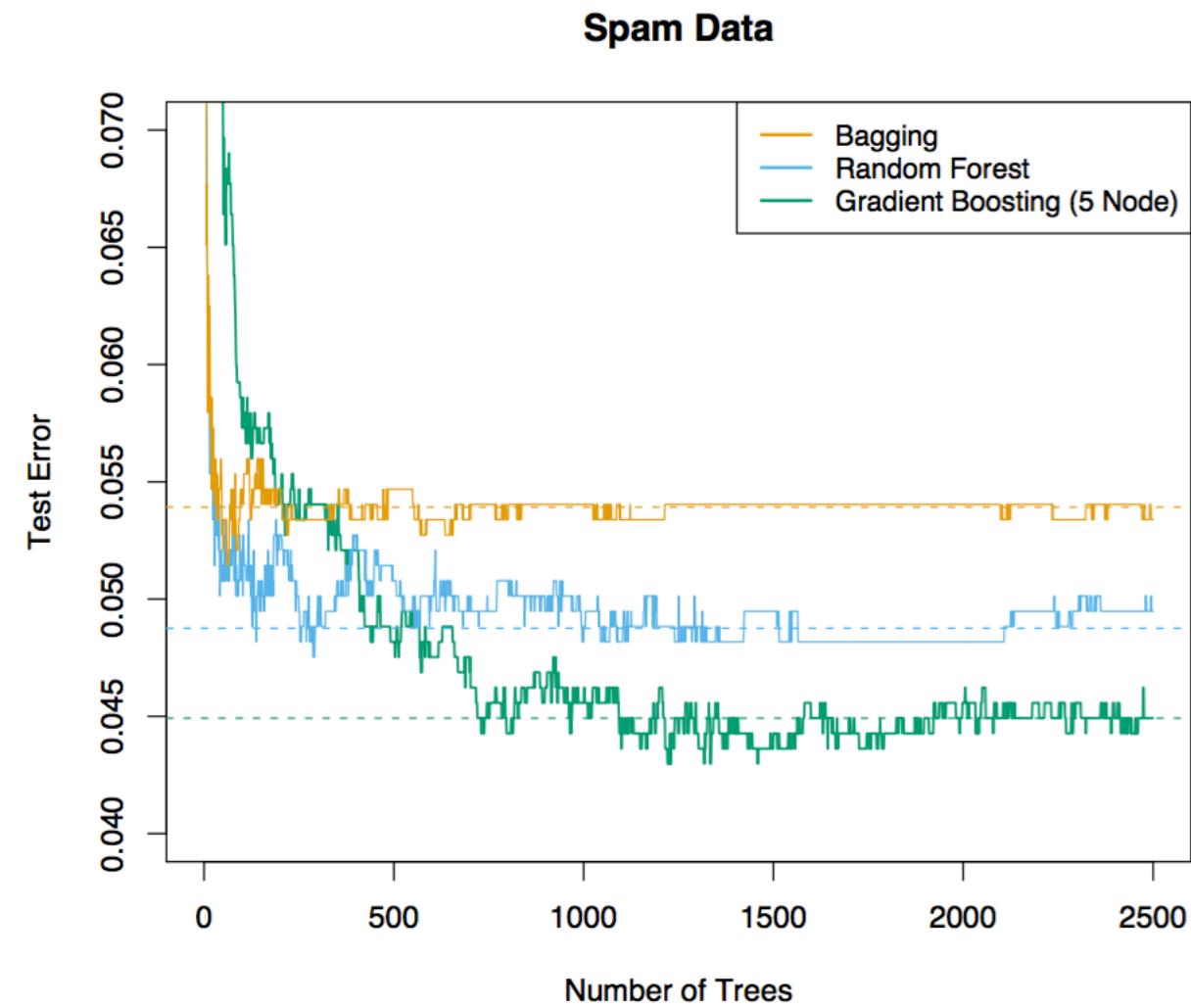
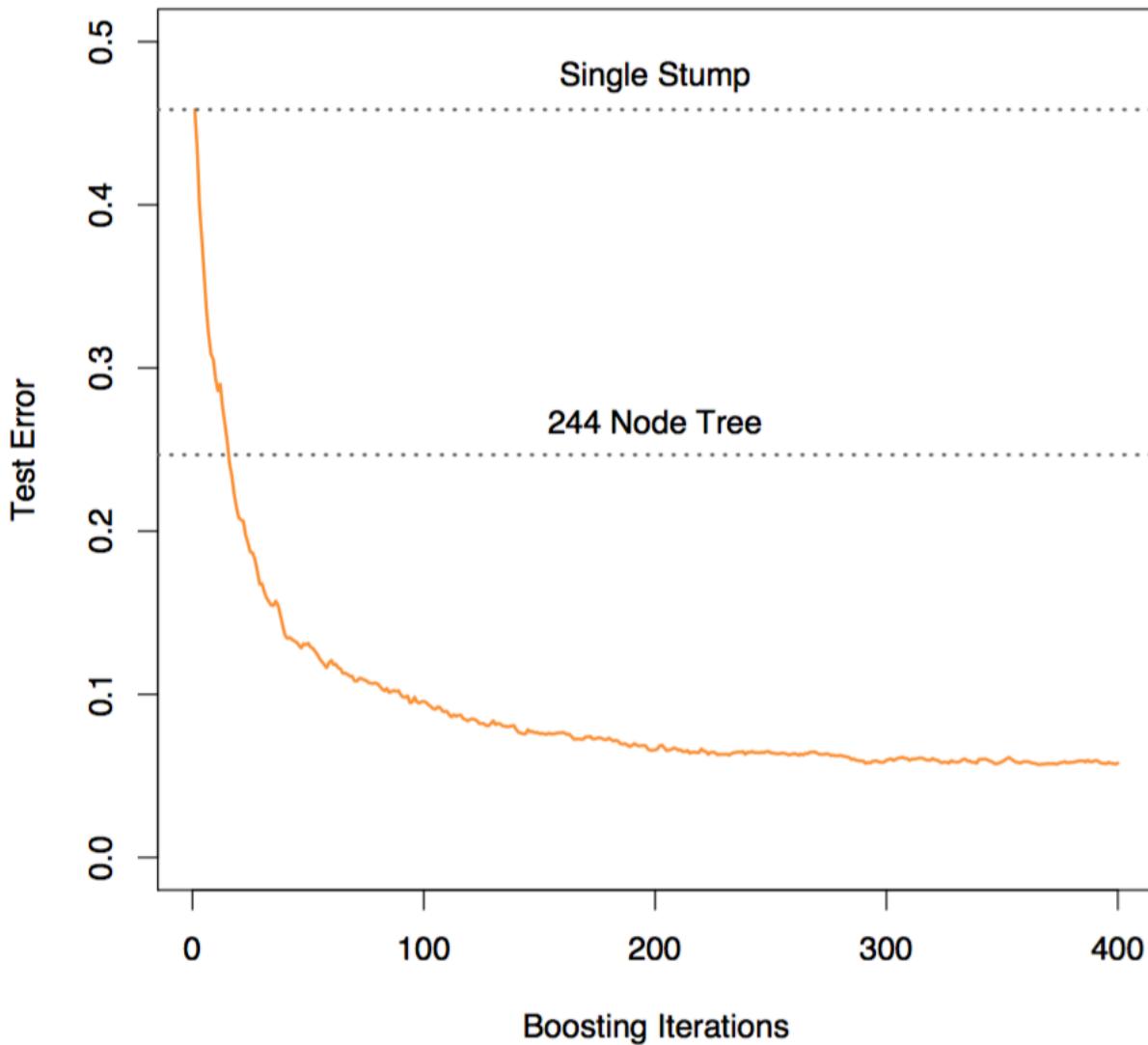
Decision Stump (пень)

# AdaBoost.M1 (1997, Freund, Schapire)

- Задача бинарной классификации:  $Y \in \{-1, 1\}$
- Def. слабый классификатор (weak classifier)
  - ошибка предсказания чуть лучше, чем у случайного угадывания.
- Цель бустинга – последовательное применение слабых классификаторов на на измененных версиях данных для построения (сильной) композиции алгоритмов.



# AdaBoost.M1 / Gradient Boosting



---

**Algorithm 10.1** AdaBoost.M1.

---

1. Initialize the observation weights  $w_i = 1/N$ ,  $i = 1, 2, \dots, N$ .
  2. For  $m = 1$  to  $M$ :
    - (a) Fit a classifier  $G_m(x)$  to the training data using weights  $w_i$ .
    - (b) Compute
$$\text{err}_m = \frac{\sum_{i=1}^N w_i I(y_i \neq G_m(x_i))}{\sum_{i=1}^N w_i}.$$
    - (c) Compute  $\alpha_m = \log((1 - \text{err}_m)/\text{err}_m)$ .
    - (d) Set  $w_i \leftarrow w_i \cdot \exp[\alpha_m \cdot I(y_i \neq G_m(x_i))]$ ,  $i = 1, 2, \dots, N$ .
  3. Output  $G(x) = \text{sign} \left[ \sum_{m=1}^M \alpha_m G_m(x) \right]$ .
- 

\* Friedman et al. (2000) - Discrete AdaBoost ( $Y \in \{-1, 1\}$ ),  
Real AdaBoost ( $Y \in [-1, 1]$ )

---

**Algorithm 10.2** Forward Stagewise Additive Modeling.

1. Initialize  $f_0(x) = 0$ .
2. For  $m = 1$  to  $M$ :
  - (a) Compute

$$(\beta_m, \gamma_m) = \arg \min_{\beta, \gamma} \sum_{i=1}^N L(y_i, f_{m-1}(x_i) + \beta b(x_i; \gamma)).$$

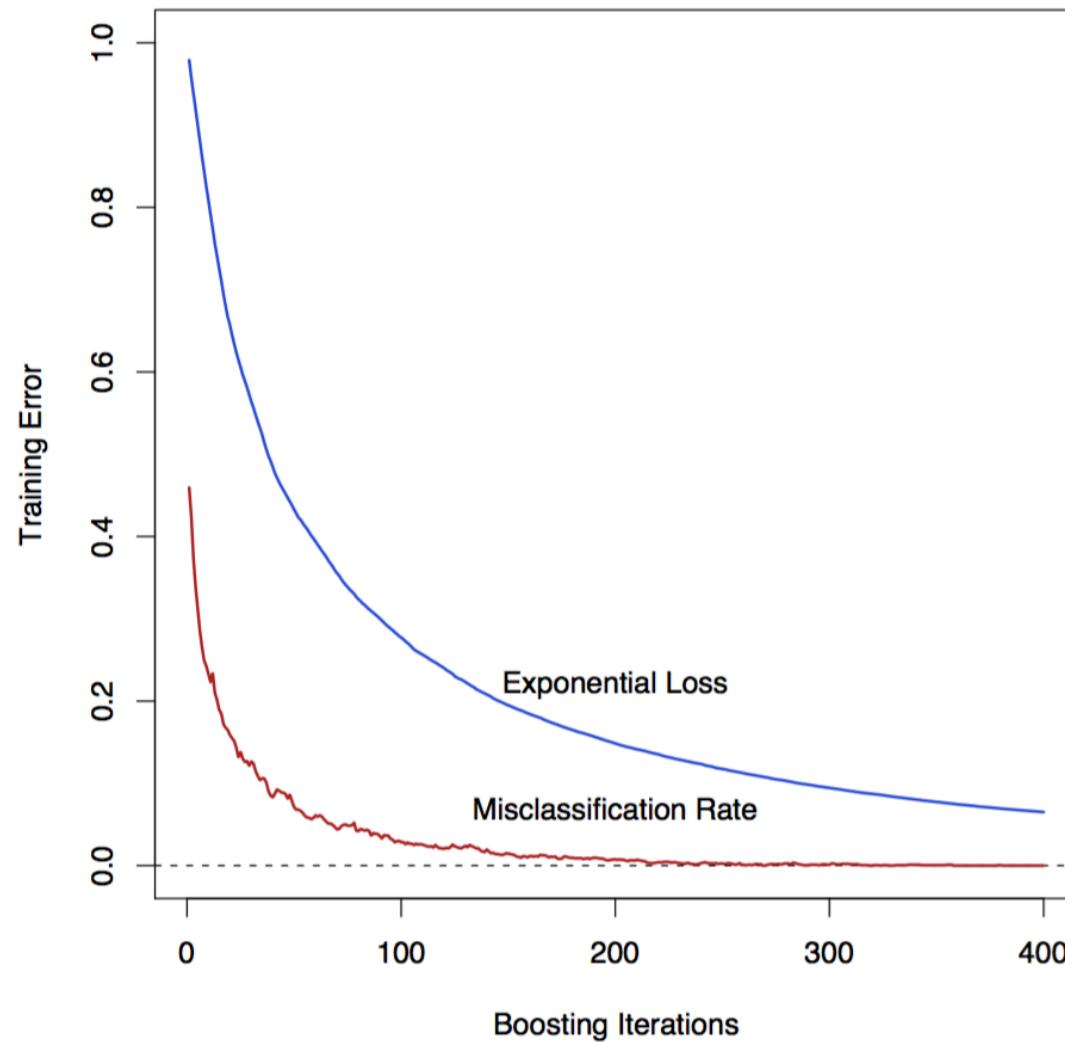
- (b) Set  $f_m(x) = f_{m-1}(x) + \beta_m b(x; \gamma_m)$ .
- 

Задание:

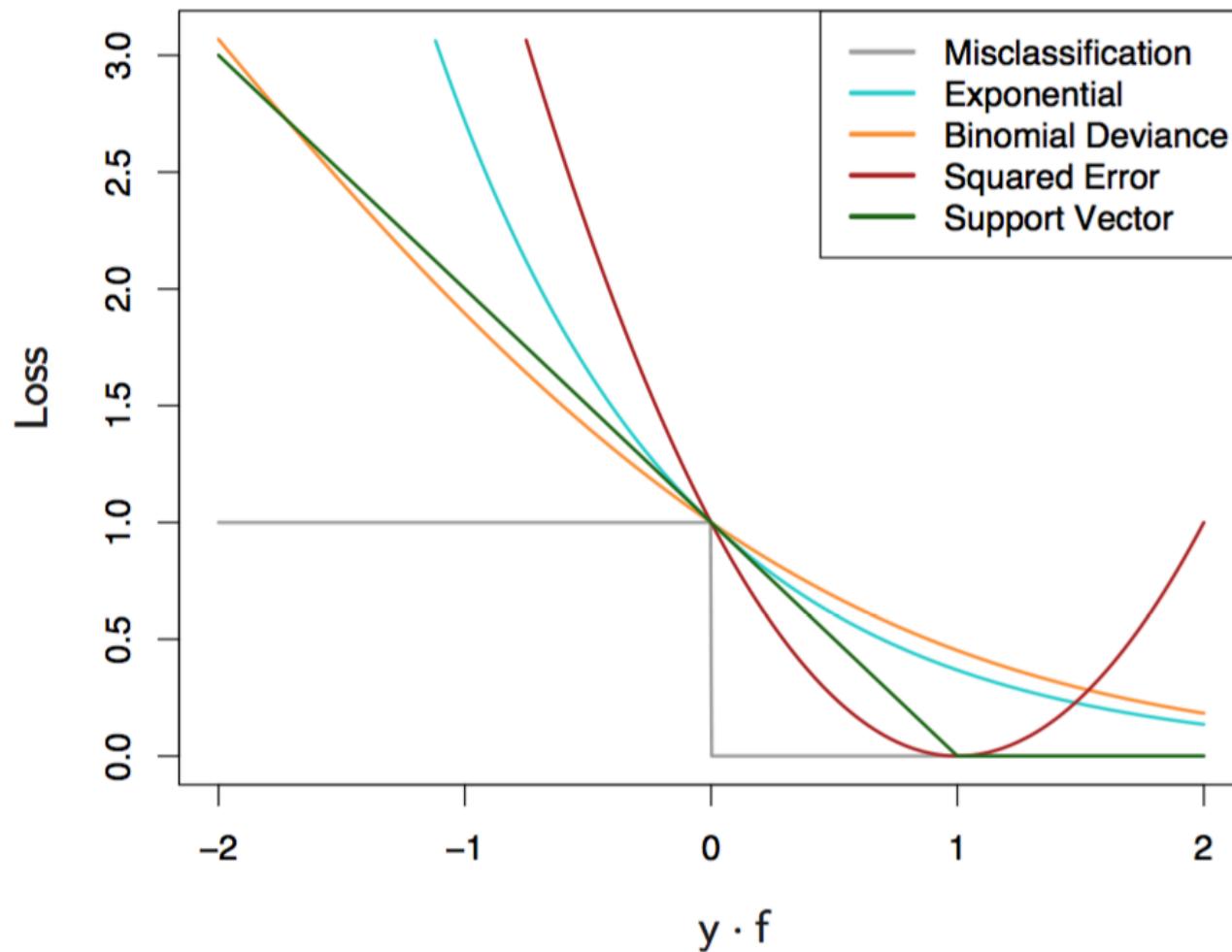
AdaBoost ~ Forward Stagewise Additive Modeling  
при  $L(y, f(x)) = \exp(-y f(x))$



# AdaBoost и Exponential Loss

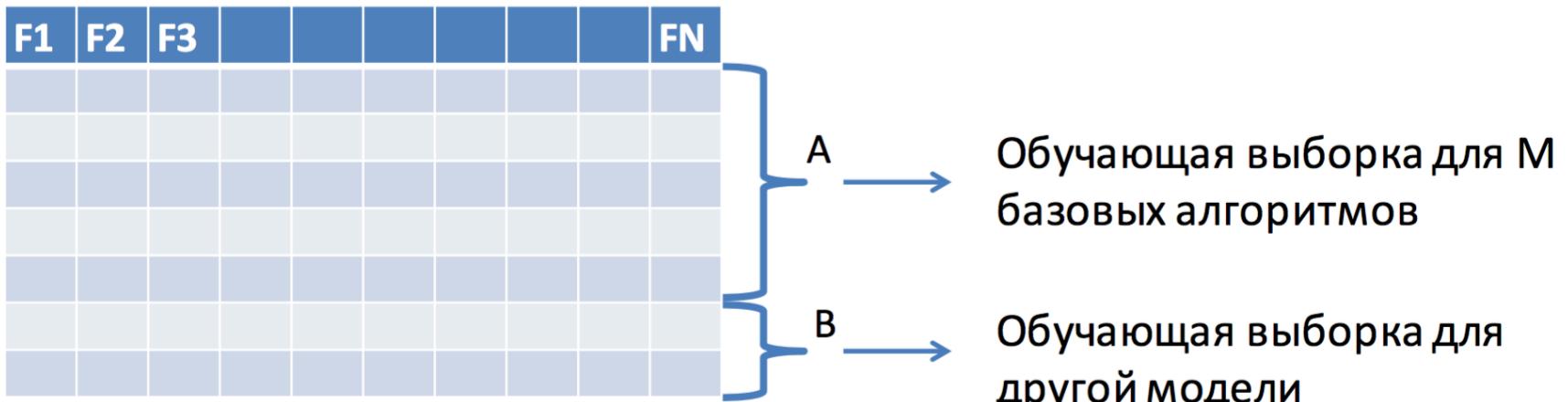


# ФУНКЦИИ ПОТЕРЬ



# Stacking

Обучающая выборка:



# Blending

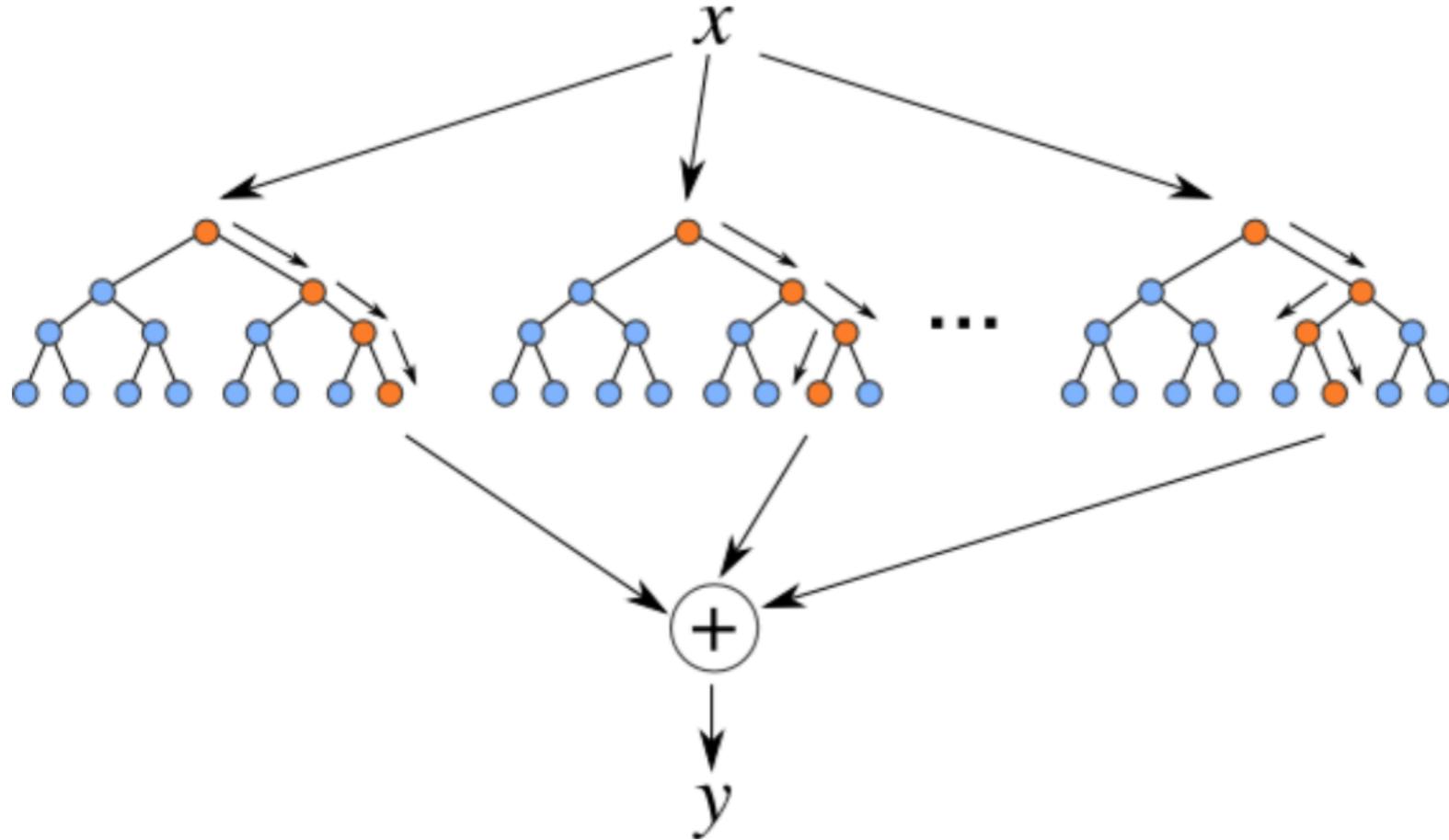
Частный случай Stacking:

$$a(x) = \sum_{t=1}^B \alpha_t b_t(x), \sum_{t=1}^B \alpha_t = 1$$

# Q&A: как ускорить вычисление алгоритмов?



# Q&A «проверка связи»: bagging, boosting или RF ?



# Резюме

- bias/variance - хороший интуитивный (и статистический) инструмент для анализа алгоритмов машинного обучения
- уменьшение variance - построение композиций деревьев на основе bootstrap:
  - Bagging / RSM / Pasting / Random Forest / Extreemle Randomized Trees
- уменьшение bias - построение композиций деревьев на основе boosting:
  - AdaBoost / Forward Stagewise Additive Modeling
- Как правильно делать Stacking и что такое Blending

# Обратная связь

Отзывы о прошедших лекциях и семинарах просьба оставлять здесь:

[https://ml-mipt.github.io/2018part1\\_Schedule/](https://ml-mipt.github.io/2018part1_Schedule/)

# Полезные материалы

- материалы курса ФИВТ МФТИ (включая архивы прошлых лет) -  
<https://github.com/ml-mipt/ml-mipt-part1>
- Hastie T., Tibshirani R., Friedman J. The Elements of Statistical Learning. (Chapter 10. Boosting, Chapter 15. Random Forests)
- James, G., Witten, D., Hastie, T., Tibshirani, R. An Introduction to Statistical Learning with Applications in R (Chapter 8. Tree-Based Methods)
- материалы курса ФКН ВШЭ -  
[http://wiki.cs.hse.ru/Машинное\\_обучение\\_1](http://wiki.cs.hse.ru/Машинное_обучение_1)

# Задача со звездочкой #1: split strategy

Цель: реализовать Random Forest правильным способом и немного «неправильным» – при построении дерева выбирать  $m$  признаков не в каждом узле дерева, а один раз на все дерево. Сравнить два подхода экспериментально (за теоретическое сравнение +1 балл) и написать отчет об исследовании.

Бонус: 2 балла (первым Зм приславшим)

Оформление: электронный документ (не скан / фото) +  
программный код ([github](#) / [gitlab](#) / [gist](#) / [nbviewer](#) / ...)

Тема письма: “ML-2018, ensemble, split strategy, ФИО (группа)”

Высыпать по адресу: [aadral@gmail.com](mailto:aadral@gmail.com)