

и деревья

Лекция 8 //

Бэггинг и случайные леса //

американский
математик
случайный лес.

Неустойчивость деревьев

Устойчивость моделей

- $X = (x_i, \underline{y_i})_{i=1}^\ell$ — обучающая выборка
- Обучаем модель $a(x)$
- Ожидаем, что модель устойчивая
- То есть не сильно меняется при небольших изменениях в X
- \tilde{X} — случайная подвыборка, примерно 90% исходной

в смысле какого-нибудь
метрического критерия

USE
S-де PR НАЕ

$$\frac{|y - \hat{y}|}{y}$$

2. PR, RECALL

$$F = \frac{PR \cdot RECALL}{PR + RECALL}$$

$$\sigma_v = 5$$

$$\chi \sim N(0, 1)$$

$$\text{Частота} = 80\%,$$

1 фолл

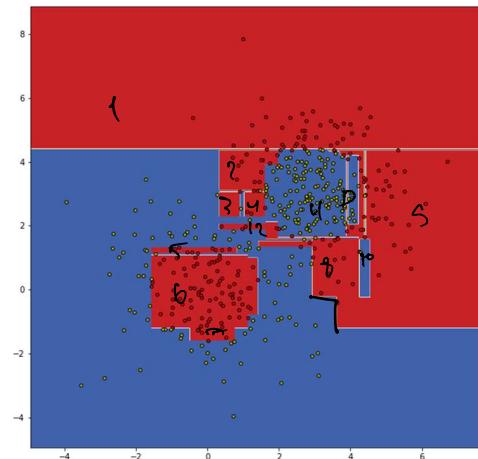
Устойчивость моделей

- \tilde{X} — случайная подвыборка, примерно 90% исходной
- Что будет происходить с деревьями на разных подвыборках?

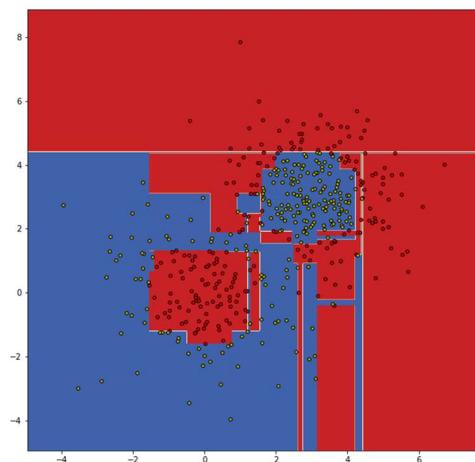
Обучение на подвыборках

• с большой глубиной

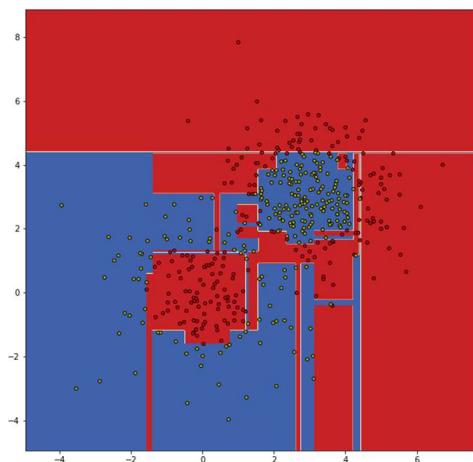
- max_depth
 - min_samples_leaf
 - min_imbalance
- отжимает
тесную



Обучение на подвыборках



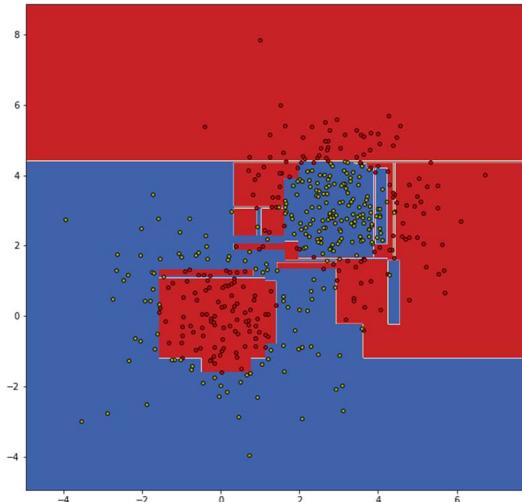
Обучение на подвыборках



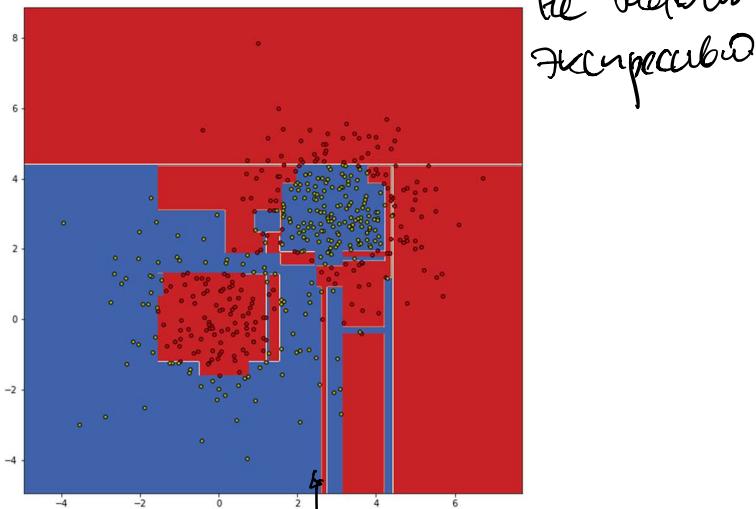
как сделать дерево
дерево устойчивым?

1) одно дерево с большой
тегущей
не устойчиво

Обучение на подвыборках



2) одно дерево
с малой
шумом
не устойчиво
экспрессивно



$$\min_{\text{depth}} \text{loss} = 5$$

u65 u62:

Композиция моделей

vote = 0



- У нас получилось N деревьев: $b_1(x), \dots, b_N(x)$
- Объединим их через голосование большинством (majority vote):

$$a(x) = \arg \max_{y \in \mathbb{Y}} \sum_{n=1}^N [b_n(x) = y]$$

= \rightarrow итого y
 \geq если $b_n(x) = y$
 $= 0$ иначе

-	1
+	0
0	+
+	0
Σ	
<u>2</u>	

}
 считаем сколько
 было членов класса

Композиция моделей

- У нас получилось N деревьев: $b_1(x), \dots, b_N(x)$
- Объединим их через голосование большинством (majority vote):

$$a(x) = \arg \max_{y \in \mathbb{Y}} \sum_{n=1}^N [b_n(x) = y]$$



Количество деревьев,
выдавших класс y

Композиция моделей

- У нас получилось N деревьев: $b_1(x), \dots, b_N(x)$
- Объединим их через голосование большинством (majority vote):

$$a(x) = \arg \max_{y \in \mathbb{Y}} \sum_{n=1}^N [b_n(x) = y]$$

Выбираем класс,
который выбрало
большинство
деревьев

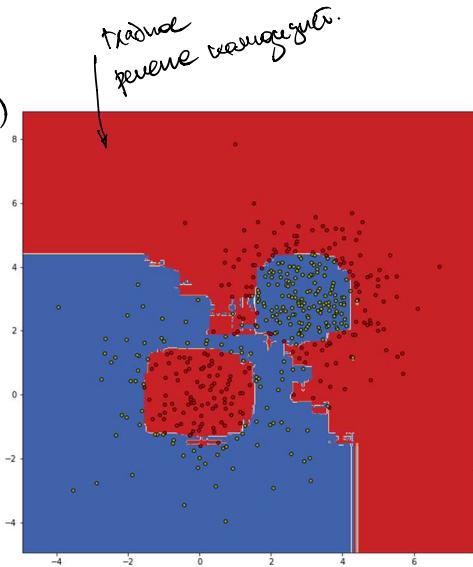
Количество деревьев,
выдавших класс y

1. избирал 80% CX
2. majority vot

Композиция моделей

Будем дерева!
 $[x_j < t]$

For j in range($\# \text{up}$)
 For t in $\mathbb{R}_{[t_0, t_1]}$:



~~НЕ~~

- такое разделение
недостижимо \Rightarrow усвоимый
алгоритм

точечное:

линейные модели

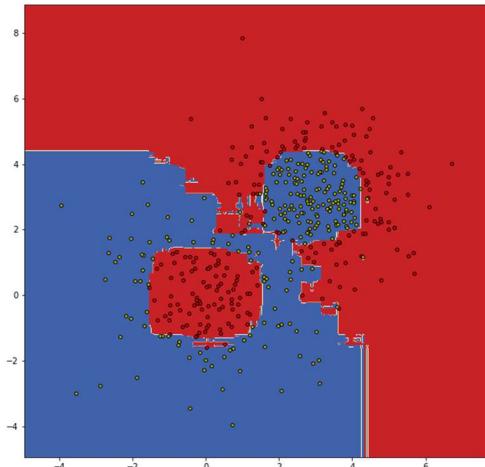
$$h(x, w) + \lambda \|w\|^2$$

OR: $\hat{w} = (X^T X + \lambda I)^{-1} X^T y$

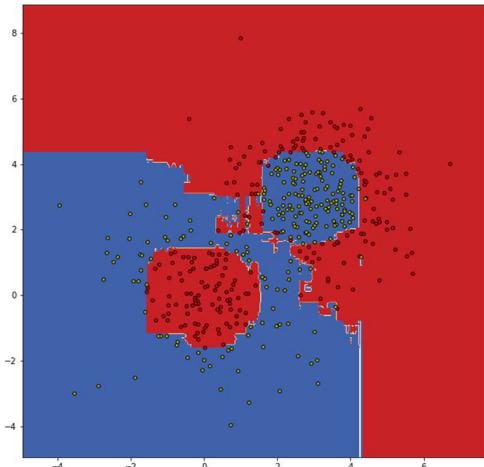
- для задач имеющие
такие модели дерева.

Композиция моделей

- = majority vote для задач классификации
- = усреднение



- как улучшить классификатор?
- разные ансамбли
- как формализовать?



Голосование по большинству и
усреднение

Majority vote



Majority vote

- Дано: N базовых алгоритмов $b_1(x), \dots, b_N(x)$
- Композиция: класс, за который проголосовало больше всего базовых алгоритмов

$$a(x) = \arg \max_{y \in \mathbb{Y}} \sum_{n=1}^N [b_n(x) = y]$$

||Усреднение|| наблюдений

- Наблюдение: усреднение результатов повышает их точность
- Измерение артериального давления
- Измерение скорости света
- Усреднение соседних пикселей изображения

$$\begin{aligned} y_1, \dots, y_N &\sim N(0, 1) \\ y_1 &| E_{y_1} = 0 \quad \therefore \bar{y} = \frac{1}{N} \sum_{n=1}^N y_n \\ &V_{y_1} = \textcircled{1} \quad \therefore E\bar{y} = \frac{1}{N} \sum_{n=1}^N (E y_n) = \frac{1}{N} \sum_{n=1}^N 0 = 0 \quad V \\ &\therefore V_{\bar{y}} = V\left(\frac{1}{N} \sum_{n=1}^N y_n\right) = \underbrace{\frac{1}{N^2} \sum_{n=1}^N V_{y_n}}_{N \cdot 1} = \textcircled{\frac{1}{N}} \end{aligned}$$

Композиции моделей

Общий вид: классификация

- $b_1(x), \dots, b_N(x)$ – базовые модели
- Каждая хотя бы немного лучше случайного угадывания
- Композиция: голосование по большинству (majority vote)

$$a_N(x) = \arg \max_{y \in \mathbb{Y}} \sum_{n=1}^N [b_n(x) = y]$$

• $b_i(x) = \{P_1, P_2, P_3\}$ т.е. ведет к алгоритмам

$$a_N(x) = \prod_{i=1}^N b_i(x)$$

② Другой

исходя на разных излучениях



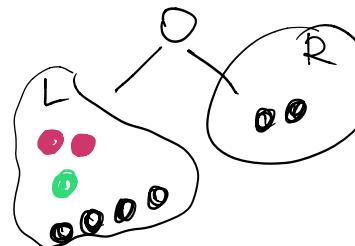
Общий вид: регрессия

Будет
Будет
такое

и т.д.

$a_N(x) = \sum_{n=1}^N b_n(x)$ — базовые модели

- Каждая хотя бы немного лучше случайного угадывания
- Композиция: усреднение

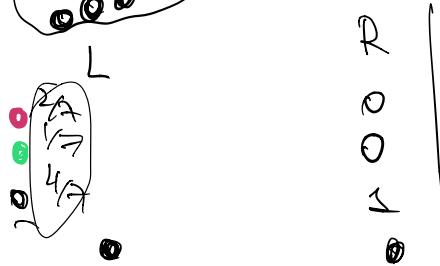


$$a_N(x) = \frac{1}{N} \sum_{n=1}^N b_n(x)$$

т.к. $a_N(x) = \frac{1}{N} \left[\prod_{n=1}^N b_n(x) \right]^{\frac{1}{N}}$

mean
mode
t.e.

$a_N(x) = \frac{1}{N} \sum_{n=1}^N a_n(x)$.



$$1. \underset{\text{Data}}{\underline{\underline{F_{\text{true}}}}} = \underset{\text{GB}}{\underline{\underline{G_B}}} \quad | \quad 2. \text{модели с помощью трансформации}$$

Базовые модели

- $b_1(x), \dots, b_N(x)$ — базовые модели
- Как на одной выборке построить \tilde{N} различных моделей?
- Вариант 1: обучить их независимо на разных подвыборках
- Вариант 2: обучать последовательно для корректировки ошибок

\uparrow
GB

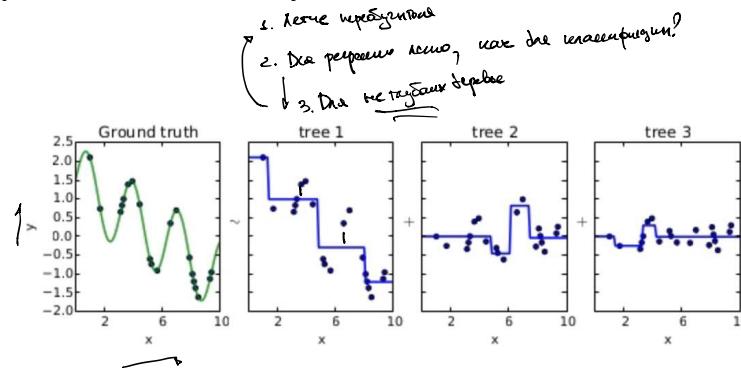
1. первая модель \rightarrow ошибки
2. следующая модель (X_{tr} , ошибок (1)),
- :

Boosting

||

БУСТИНГ

- Каждая следующая модель исправляет ошибки предыдущих
- Например, градиентный бустинг



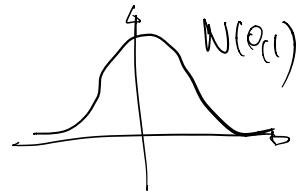
~~Подбор~~.
↓

БЭГИНГ

- Bagging (bootstrap aggregating)
- Базовые модели обучаются независимо ✓
- Каждый обучается на подмножестве обучающей выборки ✓
- Подмножество выбирается с помощью бутстрата

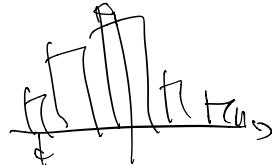
Бутстррап

$B \approx 200$



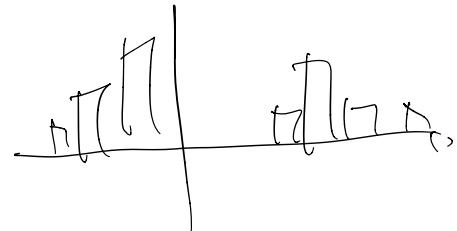
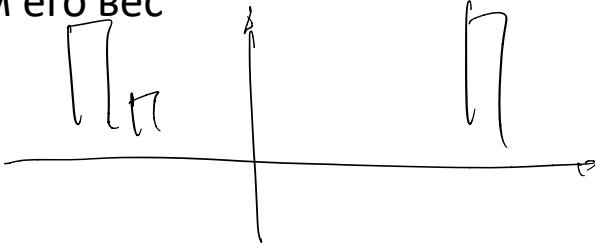
- Выборка с возвращением
- Берём ℓ элементов из X
- + Пример: $\{x_1, x_2, x_3, x_4\} \rightarrow \{x_1, x_2, x_2, x_4\}$

Семплирование
из той же совокупности

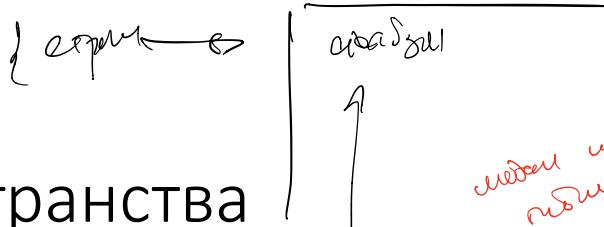


✓ каждое подвыборка
репрезентативна

- Будет:
- В подвыборке будет ℓ объектов, из них около 63.2% уникальных
 - Если объект входит в выборку несколько раз, то мы как бы повышаем его вес



Случайные подпространства



множество
пункт.



учебная
часть

- Выбираем случайное подмножество признаков

- Обучаем модель только на них

$$E \underbrace{\left(\sum_{n=1}^N b_n(x) \right)}_{\text{множество}} = \frac{1}{N} \sum_{n=1}^N E b_n(x) = \begin{cases} \text{одинаковые априори} \\ \text{на разных подмножествах} \end{cases} \stackrel{y}{=} \underline{\underline{E b_5(x)}}$$

$$V \left(\frac{1}{N} \sum_{n=1}^N b_n(x) \right) = \frac{1}{N^2} V \left(\sum_{n=1}^N b_n(x) \right) = \frac{1}{N^2} \sum_{n=1}^N V(b_n(x)) + \underbrace{2 \sum_{\substack{i < j \\ N^2}} \text{cov}(b_i(x), b_j(x))}_{\text{учебная часть}}$$

одинаково это же

Случайные подпространства

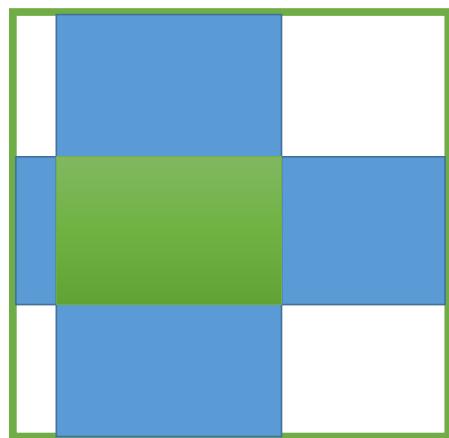
- Выбираем случайное подмножество признаков
 - Обучаем модель только на них
- !
- Может быть плохо, если имеются важные признаки, без которых невозможно построить разумную модель

Виды рандомизации

- 1) • Бэггинг: случайная подвыборка
- +
2) • Случайные подпространства:
случайное подмножество
признаков

можно брать все

Random Forest.



Резюме

- Будем объединять модели в композиции через усреднение или голосование большинством
- Бэггинг — композиция моделей, обученных независимо на случайных подмножествах объектов
- Можно ещё рандомизировать по признакам
- Как лучше всего?

Смещение и разброс моделей

Форема
Байеса

$$p(y|x) = \frac{p(y|x)p(x)}{\int y p(y|x) dy}$$

Факторизован
пред.

Разложение ошибки на смещение и разброс

x признак
 y ответ

$p(x,y) =$

$\underbrace{E_{(x,y)}[(y - E[y|x])^2]}_{\text{шум}}$
 $L(\mu) = \underbrace{\mathbb{E}_{x,y}[(y - \mathbb{E}[y|x])^2]}_{\text{шум}} +$
 $+ \underbrace{\mathbb{E}_x \left[(\mathbb{E}_X[\mu(X)] - \mathbb{E}[y|x])^2 \right]}_{\text{смещение}} + \underbrace{\mathbb{E}_x \left[\mathbb{E}_X[(\mu(X) - \mathbb{E}_X[\mu(X)])^2] \right]}_{\text{разброс}}$

(2)

- Разберём на уровне идеи

$$\mathbb{E}_x \left(\mathbb{E}_X[\mu(X)] - \mathbb{E}[y|x] \right)^2 + \mathbb{E}_x \left[\mathbb{E}_X[(\mu(X) - \mathbb{E}_X[\mu(X)])^2] \right]$$

Разложение ошибки на смещение и разброс

- Ошибка модели складывается из трёх компонент
- Шум (noise) — характеристика сложности и противоречивости данных

Разложение ошибки на смещение и разброс

- Ошибка модели складывается из трёх компонент
- Шум (noise) — характеристика сложности и противоречивости данных
- Смещение (bias) — способность модели приблизить лучшую среди всех возможных моделей

Разложение ошибки на смещение и разброс

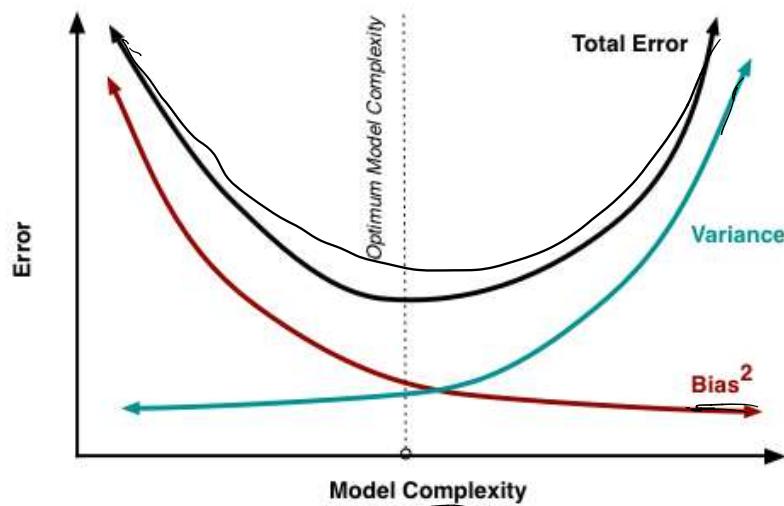
- Ошибка модели складывается из трёх компонент
- Шум (noise) — характеристика сложности и противоречивости данных
- Смещение (bias) — способность модели приблизить лучшую среди всех возможных моделей
- Разброс (variance) — устойчивость модели к изменениям в обучающей выборке

Смещение и разброс

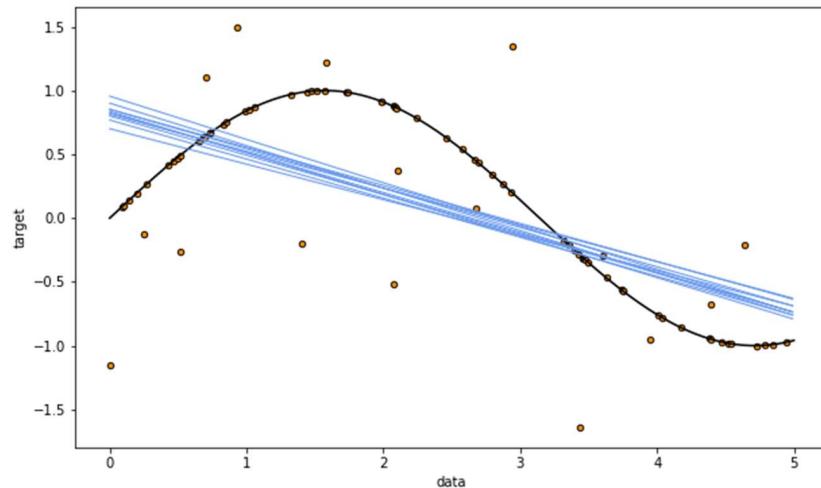
- Высокое смещение может говорить о недообучении (слишком большая ошибка)
- Высокий разброс может говорить о переобучении (слишком сложная модель)

Bias-variance tradeoff

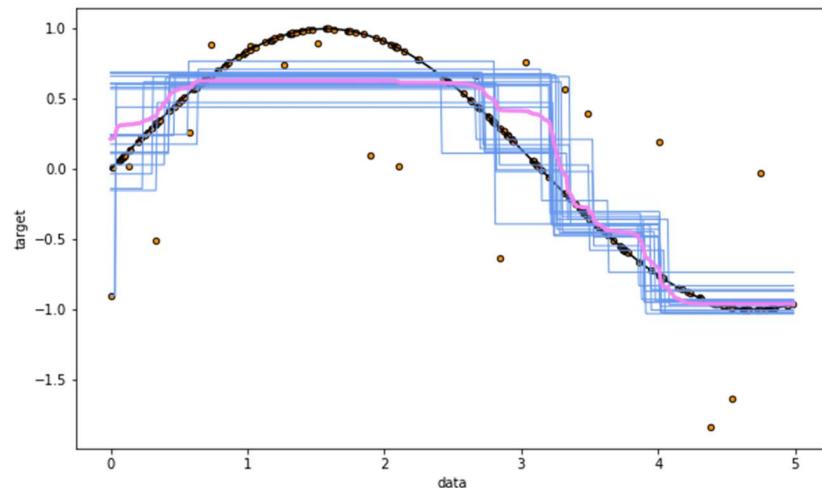
↓ bias ↓ Variance
↑ bias ↓ variance



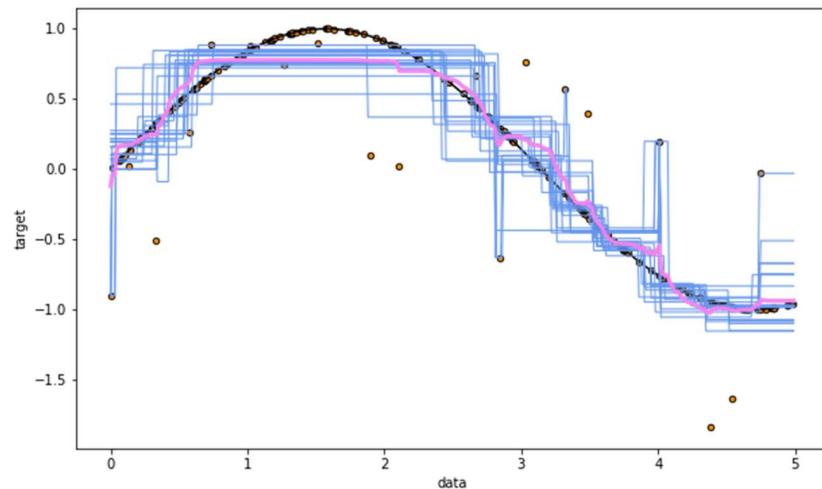
Смещение и разброс: линейная модель



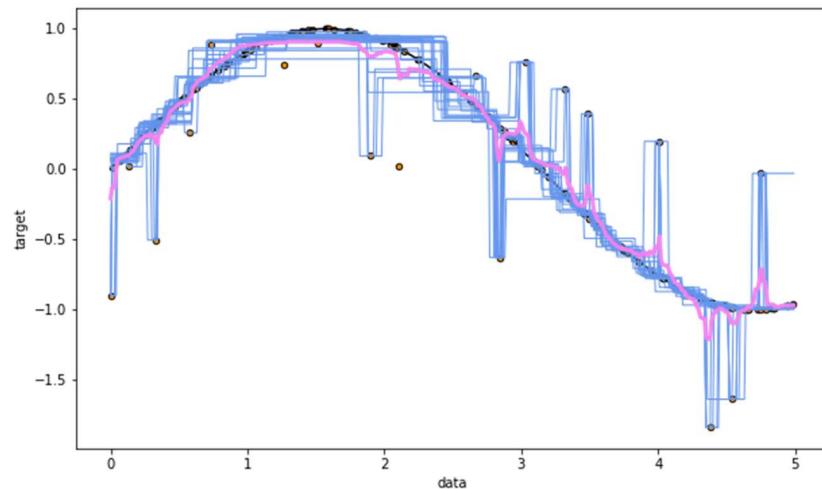
Смещение и разброс: деревья



Смещение и разброс: деревья



Смещение и разброс: деревья



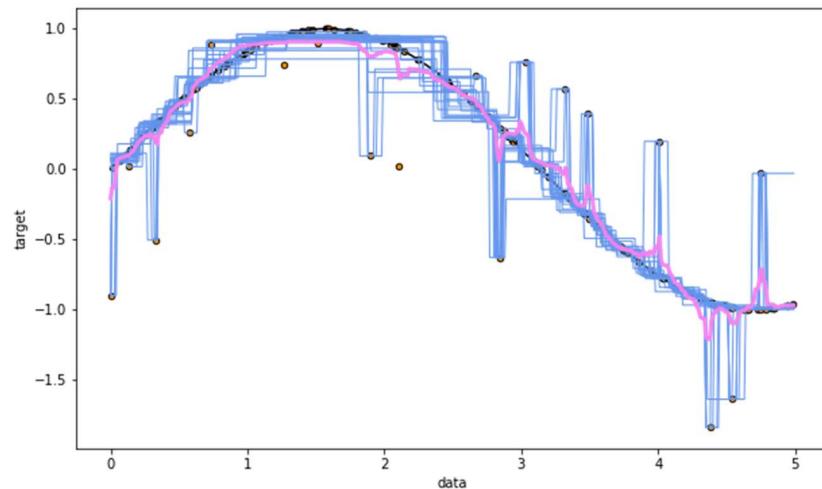
БЭГИНГ

- Смещение $a_N(x)$ такое же, как у $b_n(x)$
- Разброс $a_N(x)$:

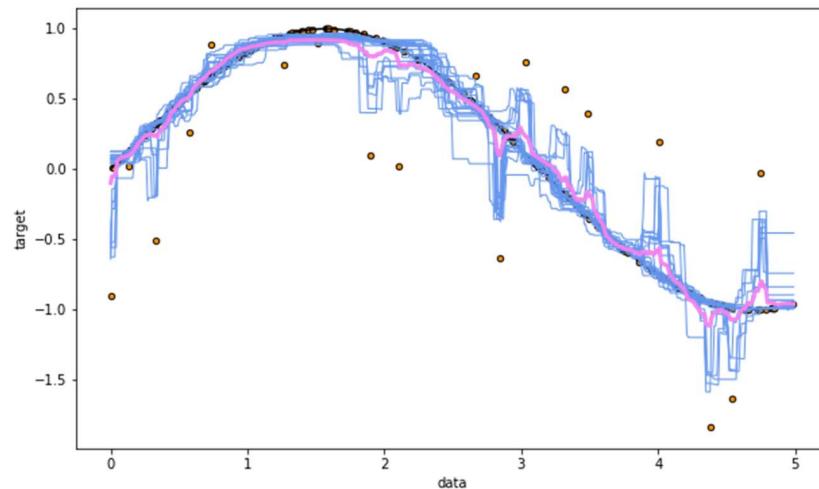
$$\frac{1}{N}(\text{разброс } b_n(x)) + \text{ковариация}(b_n(x), b_m(x))$$

- Если базовые модели независимы, то разброс уменьшается в N раз!
- Чем более похожи выходы базовых моделей, тем меньше эффект от построения композиции

Смещение и разброс: деревья



Смещение и разброс: бэггинг



Random Forest

Случайное поддерево

[Случайный лес]

Жадный алгоритм

$\text{SplitNode}(m, R_m)$

1. Если выполнен критерий останова, то выход
2. Ищем лучший предикат: $j, t = \arg \min_{j,t} Q(R_m, j, t)$
3. Разбиваем с его помощью объекты: $R_\ell = \left\{ (x, y) \in R_m \mid [x_j < t] \right\},$
 $R_r = \left\{ (x, y) \in R_m \mid [x_j \geq t] \right\}$
4. Повторяем для дочерних вершин: $\text{SplitNode}(\ell, R_\ell)$ и $\text{SplitNode}(r, R_r)$

Жадный алгоритм

$\text{SplitNode}(m, R_m)$

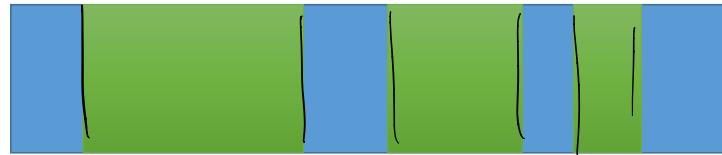
1. Если выполнен критерий останова, то выход
2. Ищем лучший предикат: $j, t = \arg \min_{j,t} Q(R_m, j, t)$
3. Разбиваем с его помощью объекты: $R_\ell = \left\{ (x, y) \in R_m \mid [x_j < t] \right\},$
 $R_r = \left\{ (x, y) \in R_m \mid [x_j \geq t] \right\}$
4. Повторяем для дочерних вершин: $\text{SplitNode}(\ell, R_\ell)$ и $\text{SplitNode}(r, R_r)$

Выбор предиката

$$\{x_j < t\}$$

$$j, t = \arg \min_{j,t} Q(R_m, j, t)$$

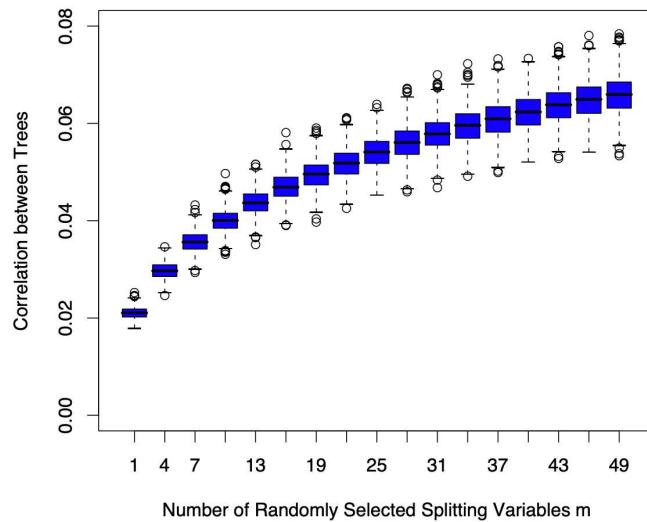
- Будем искать лучший предикат среди случайного подмножества признаков размера q



① *выбор*

② *делим метод*

Корреляция между деревьями



Hastie, Tibshirani, Friedman. The Elements of Statistical Learning.

Корреляция между деревьями

(2) Рекомендации для q :

- Регрессия: $q = \frac{d}{\sqrt{3}} \rightarrow \frac{1}{\sqrt{2}} = \frac{1}{\sqrt{4}}$?
- Классификация: $q = \sqrt{d}$

① $\frac{= \text{auto}}{\sqrt{d}}$

“тесно”
и не “тесно”.

Случайный лес (Random Forest)

Для $n = 1, \dots, N$:

1. Сгенерировать выборку \tilde{X} с помощью бутстрапа
 2. Построить решающее дерево $b_n(x)$ по выборке \tilde{X}
 3. Дерево строится, пока в каждом листе не окажется
не более n_{min} объектов
 4. Оптимальное разбиение ищется среди \underline{q} случайных признаков
- 

Случайный лес (Random Forest)

Для $n = 1, \dots, N$:

1. Сгенерировать выборку \tilde{X} с помощью бутстрата
2. Построить решающее дерево $b_n(x)$ по выборке \tilde{X}
3. Дерево строится, пока в каждом листе не окажется не более n_{min} объектов
4. Оптимальное разбиение ищется **среди q случайных признаков**

Выбираются заново при каждом разбиении!

Случайный лес (Random Forest)

- Регрессия:

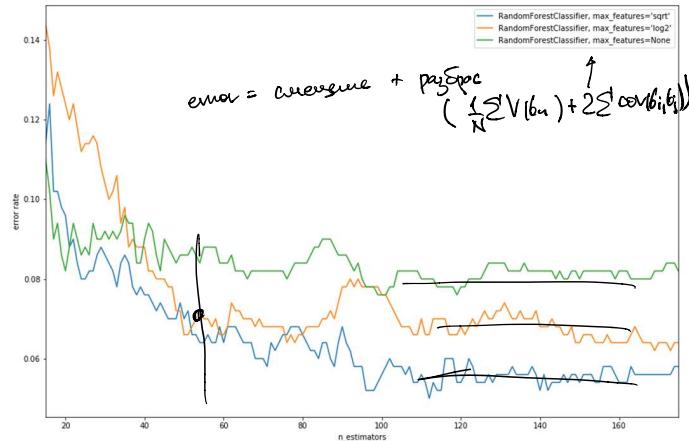
$$a(x) = \frac{1}{N} \sum_{n=1}^N b_n(x)$$

- Классификация:

$$a(x) = \arg \max_{y \in \mathbb{Y}} \sum_{n=1}^N [b_n(x) = y]$$

Универсальный метод

- Ошибка сначала убывает, а затем выходит на один уровень
- Случайный лес не переобучается при росте N



↗
следующее изображение
объекта
→ Вектор
— вектор с информацией.

Out-of-bag

↗
4

- Каждое дерево обучается примерно на 63% данных
- Остальные объекты — как бы тестовая выборка для дерева
- X_n — обучающая выборка для $b_n(x)$
- Можно оценить ошибку на новых данных:

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

↗
 (X_2, X_2, X_4, X_4)
 b_2
 (X_1, X_5)

Out-of-bag

- Каждое дерево обучается примерно на 63% данных
- Остальные объекты — как бы тестовая выборка для дерева
- X_n — обучающая выборка для $b_n(x)$
- Можно оценить ошибку на новых данных:

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



Для каждого
объекта
выборки

Out-of-bag

- Каждое дерево обучается примерно на 63% данных
- Остальные объекты — как бы тестовая выборка для дерева
- X_n — обучающая выборка для $b_n(x)$
- Можно оценить ошибку на новых данных:

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

Для каждого
объекта
выборки

Суммируем ответы для всех
деревьев, в которые не
попал объект

Out-of-bag

- Каждое дерево обучается примерно на 63% данных
- Остальные объекты — как бы тестовая выборка для дерева
- X_n — обучающая выборка для $b_n(x)$
- Можно оценить ошибку на новых данных:

$$Q_{test} = \frac{1}{\ell} \sum_{i=1}^{\ell} L \left(y_i, \underbrace{\frac{1}{\sum_{n=1}^N [x_i \notin X_n]} \sum_{n=1}^N [x_i \notin X_n] b_n(x_i)}_{\text{Среднее ответов} = \text{предсказание на объекте}} \right)$$

Для каждого объекта выборки Среднее ответов = предсказание на объекте Суммируем ответы для всех деревьев, в которые не попал объект

Out-of-bag (пример)

- 4 объекта: (x_1, y_1) , (x_2, y_2) , (x_3, y_3) , (x_4, y_4)

[the Bootstrap sample
составляется вектором = баге несет]

Out-of-bag (пример)

- 4 объекта: $(x_1, y_1), (x_2, y_2), (x_3, y_3), (x_4, y_4)$
- 3 дерева:
 - Дерево b_1 : train on $(x_1, y_1), (x_2, y_2)$
 - Дерево b_2 : train on $(x_3, y_3), (x_4, y_4)$
 - Дерево b_3 : train on $(x_1, y_1), (x_2, y_2), (x_3, y_3)$

Out-of-bag (пример)

- 4 объекта: $(x_1, y_1), (x_2, y_2), (x_3, y_3), (x_4, y_4)$
- 3 дерева:
 - Дерево b_1 : train on $(x_1, y_1), (x_2, y_2) \rightarrow x_3, x_4$
 - Дерево b_2 : train on $(x_3, y_3), (x_4, y_4) \rightarrow x_1, x_2$
 - Дерево b_3 : train on $(x_1, y_1), (x_2, y_2), (x_3, y_3) \rightarrow x_4$.

$$Q_1 = L(y_1, b_2(x_1))$$

Out-of-bag (пример)

- 4 объекта: $(x_1, y_1), (x_2, y_2), (x_3, y_3), (x_4, y_4)$
- 3 дерева:
 - Дерево b_1 : train on $(x_1, y_1), (x_2, y_2)$
 - Дерево b_2 : train on $(x_3, y_3), (x_4, y_4)$
 - Дерево b_3 : train on $(x_1, y_1), (x_2, y_2), (x_3, y_3)$

$$Q_1 = L(y_1, b_2(x_1))$$
$$Q_2 = L(y_2, b_2(x_2))$$

Out-of-bag (пример)

- 4 объекта: $(x_1, y_1), (x_2, y_2), (x_3, y_3), (x_4, y_4)$
- 3 дерева:
 - Дерево b_1 : train on $(x_1, y_1), (x_2, y_2)$
 - Дерево b_2 : train on $(x_3, y_3), (x_4, y_4)$
 - Дерево b_3 : train on $(x_1, y_1), (x_2, y_2), (x_3, y_3)$

$$Q_1 = L(y_1, b_2(x_1))$$

$$Q_2 = L(y_2, b_2(x_2))$$

$$Q_3 = L(y_3, b_1(x_3))$$

Out-of-bag (пример)

- 4 объекта: $(x_1, y_1), (x_2, y_2), (x_3, y_3), (x_4, y_4)$
- 3 дерева:
 - Дерево b_1 : train on $(x_1, y_1), (x_2, y_2)$
 - Дерево b_2 : train on $(x_3, y_3), (x_4, y_4)$
 - Дерево b_3 : train on $(x_1, y_1), (x_2, y_2), (x_3, y_3)$

$$\begin{aligned}Q_1 &= L(y_1, b_2(x_1)) \\Q_2 &= L(y_2, b_2(x_2)) \\Q_3 &= L(y_3, b_1(x_3)) \\Q_4 &= L\left(y_4, \underbrace{(b_1(x_4) + b_3(x_4))}_{\text{---}}\right)\end{aligned}$$

Out-of-bag (пример)

$$\begin{aligned}Q_1 &= L(y_1, b_2(x_1)) \\Q_2 &= L(y_2, b_2(x_2)) \\Q_3 &= L(y_3, b_1(x_3)) \\Q_4 &= L\left(y_4, \frac{1}{2}(b_1(x_4) + b_3(x_4))\right)\end{aligned}$$

$$Q_{test} = \frac{1}{4}(Q_1 + Q_2 + Q_3 + Q_4)$$

Важность признаков

- Перестановочный метод для проверки важности j -го признака
- Перемешиваем соответствующий столбец в матрице «объекты-признаки» для тестовой выборки
- Измеряем качество модели
- Чем сильнее оно упало, тем важнее признак

Резюме

+ agreement to be open
in public

- Случайный лес — метод на основе бэггинга, в котором делается попытка повысить разнообразие деревьев, не зависеть от них

-  Метод практически без гиперпараметров N, = 100, 200

- Можно оценить обобщающую способность без тестовой выборки

00 B.

Sklearn

- RandomForest

$$\text{oob} = \frac{\text{True}}{\text{False}}$$

◦ oob-score -

DecisionTree / SVM

- Bagging Regressor (median)

- Bagging Classifier (voting)

◦ oob-score .

$$x_s, \dots, x_t \rightarrow RF(x_i) \rightarrow y_i \quad (V)$$

$$\rightarrow RF(x_t, \underbrace{y_{t-s}}_{\text{auxiliary}}) \rightarrow$$