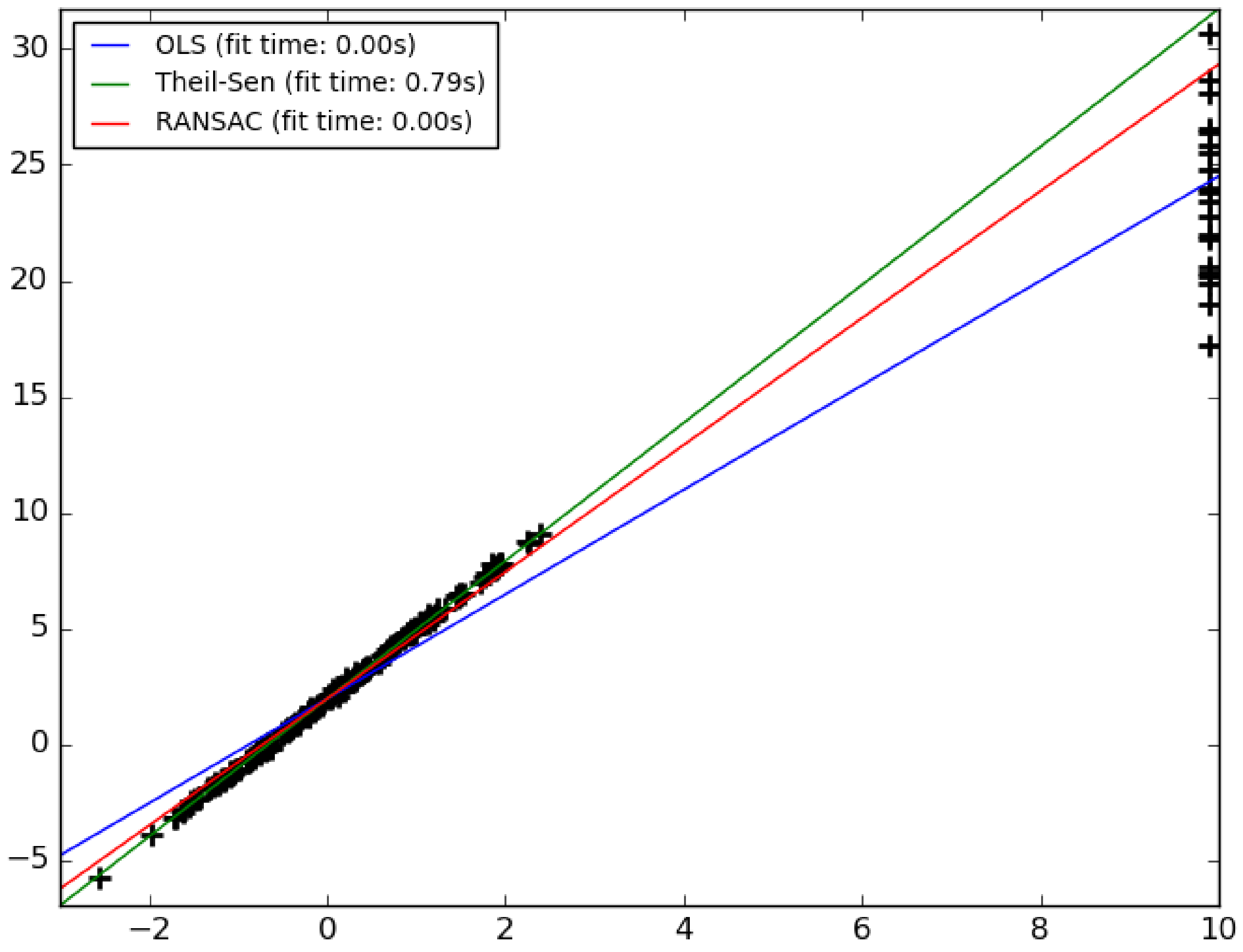
Ансамблевые методы

Theil-Sen Regressor

Считаем медианный вектор w по подмножествам X

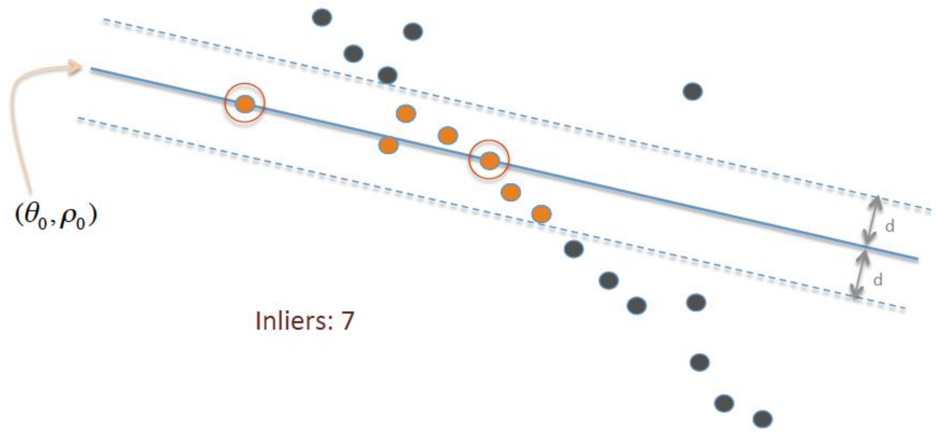


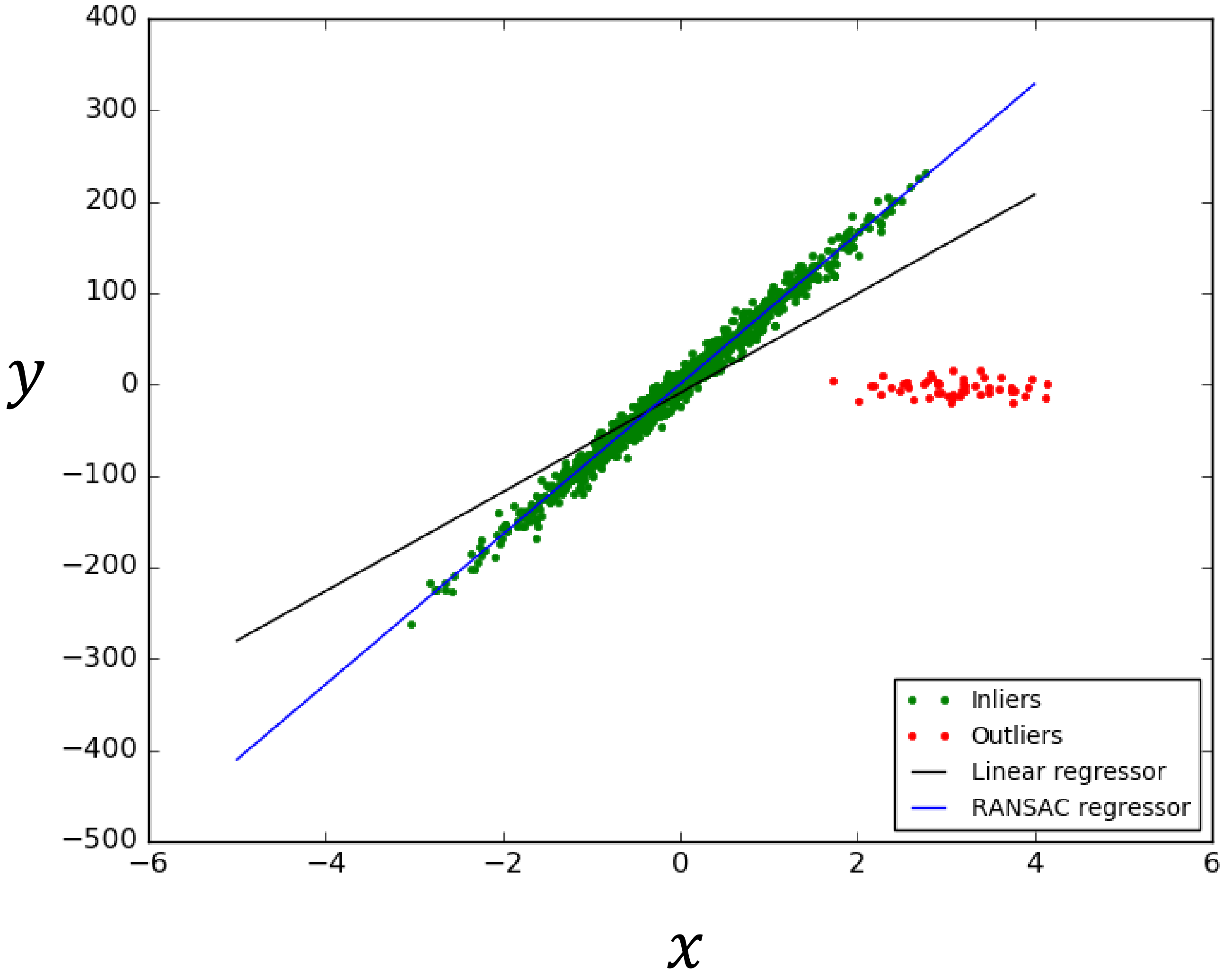
RANSAC: RANdom SAmple Consensus

1. Строим модель по случайной подвыборке.

2. Принимаем модель в набор, если достаточное количество точек лежит внутри определенной полосы (inliers).

3. Повторяем 1-2 заданное количество раз, после чего выбираем лучшую модель из набора и дообучаем на всех inliers.





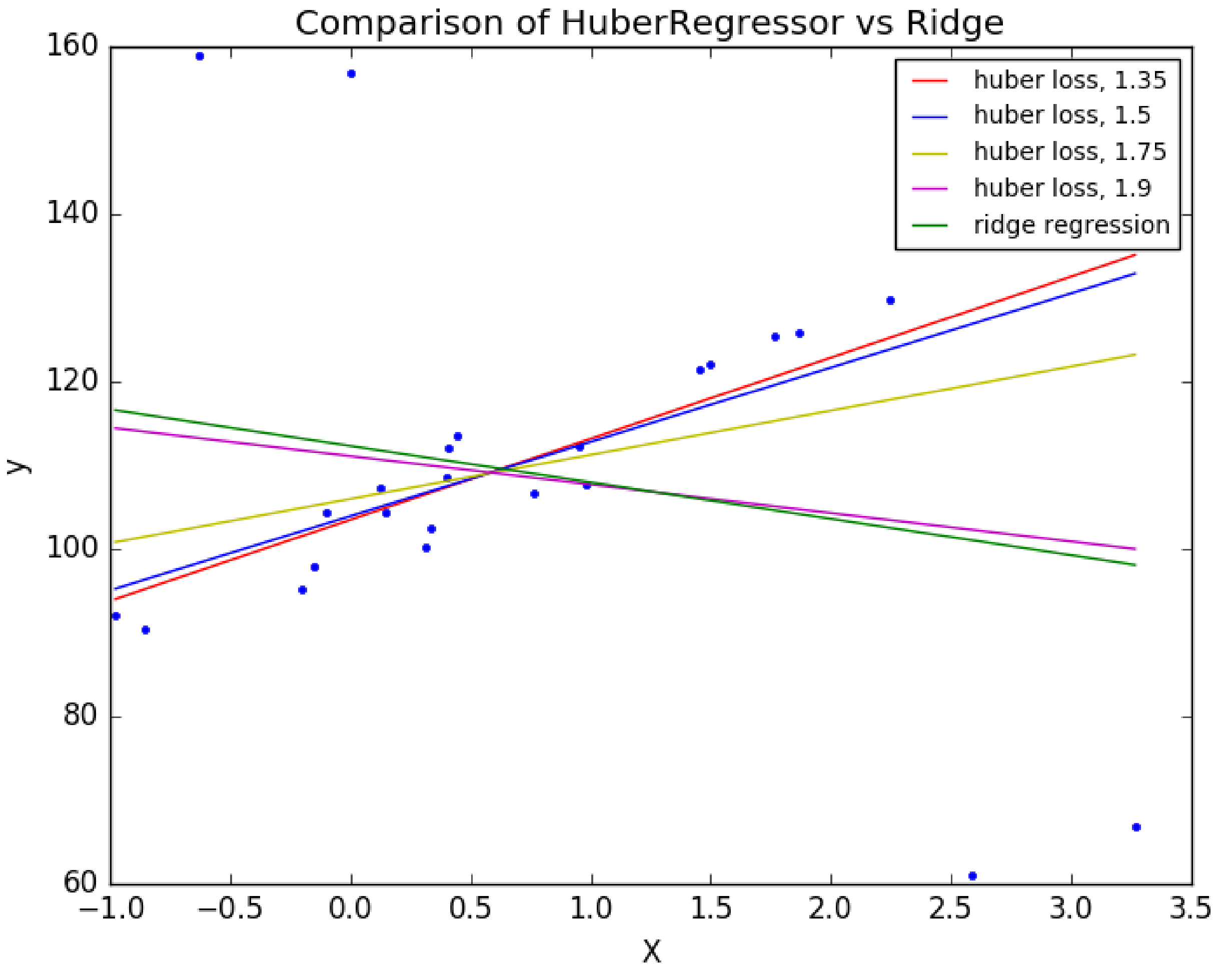
Huber Regressor

Квадратная ошибка для близких точек (inliers), линейная для дальних (outliers).

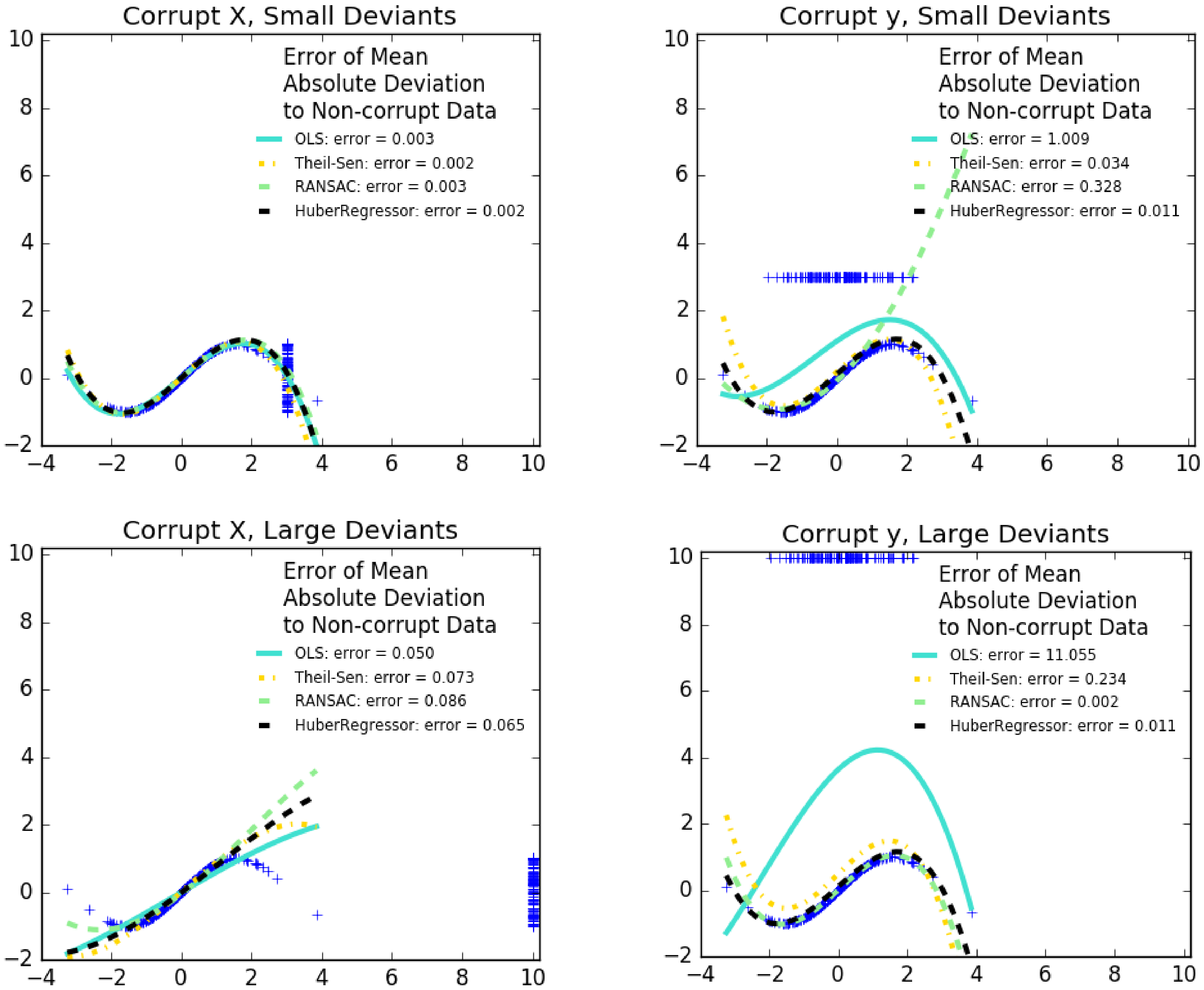
$$\min_{w,\sigma} \sum_{i=1}^{N} H_m\left(\frac{\mathrm{x}_i \mathrm{w} - y_i}{\sigma}\right) + \alpha \|\mathrm{w}\|_2^2$$

$$H_m(z) = \begin{cases} z^2, & \text{если } |z| < \epsilon \\ 2\epsilon|z| - \epsilon^2, & \text{если } |z| \geq \epsilon \end{cases}$$

σ - константа масштабирования.



RANSAC vs. Theil-Sen vs. Huber



Voting

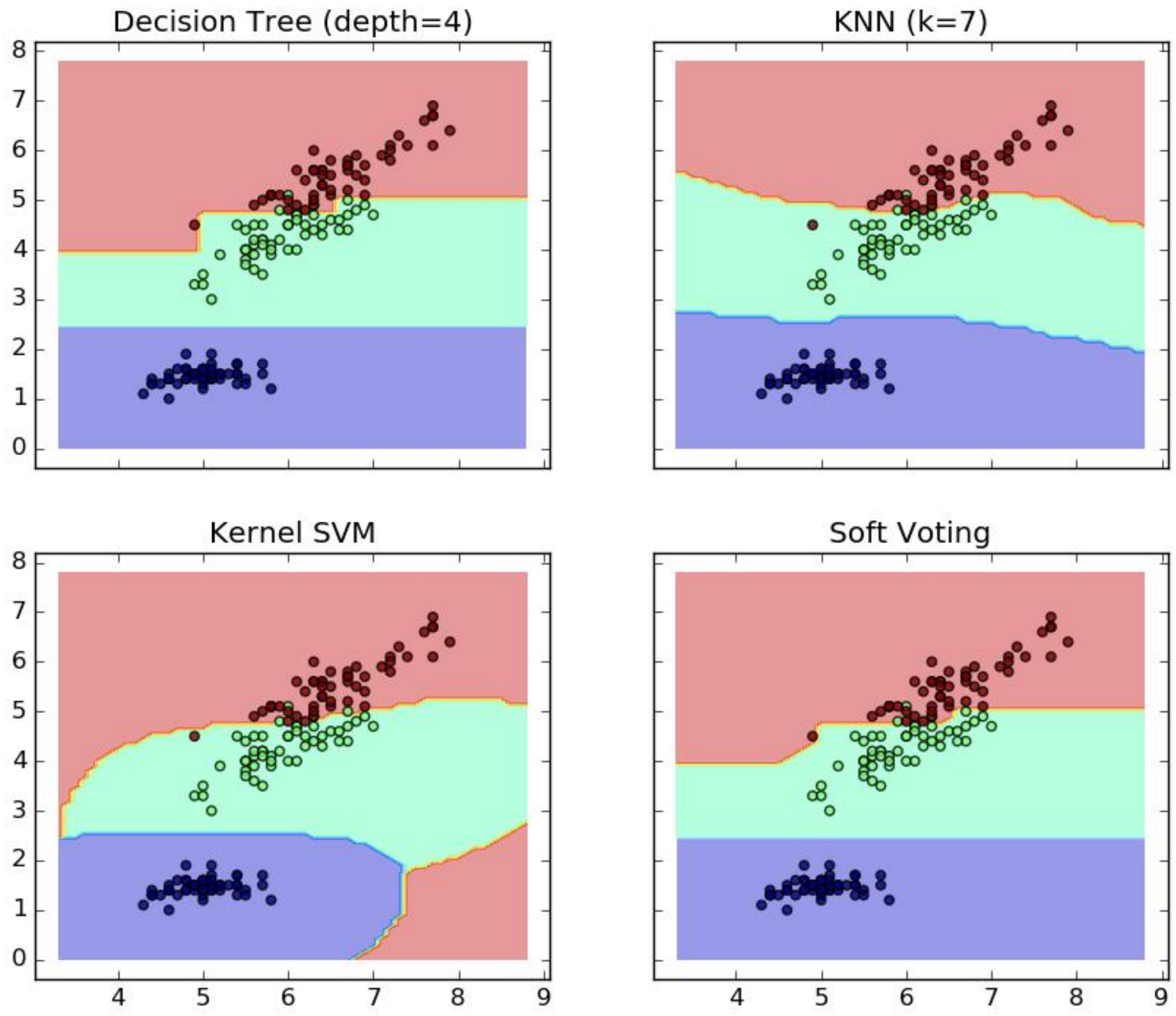
Hard voting:

Побеждает класс, за который голосует большее число голосов.

(При ничьей в sklearn победа отдается первому по алфавиту классу).

Soft voting:

Считается средняя вероятность.



Bagging

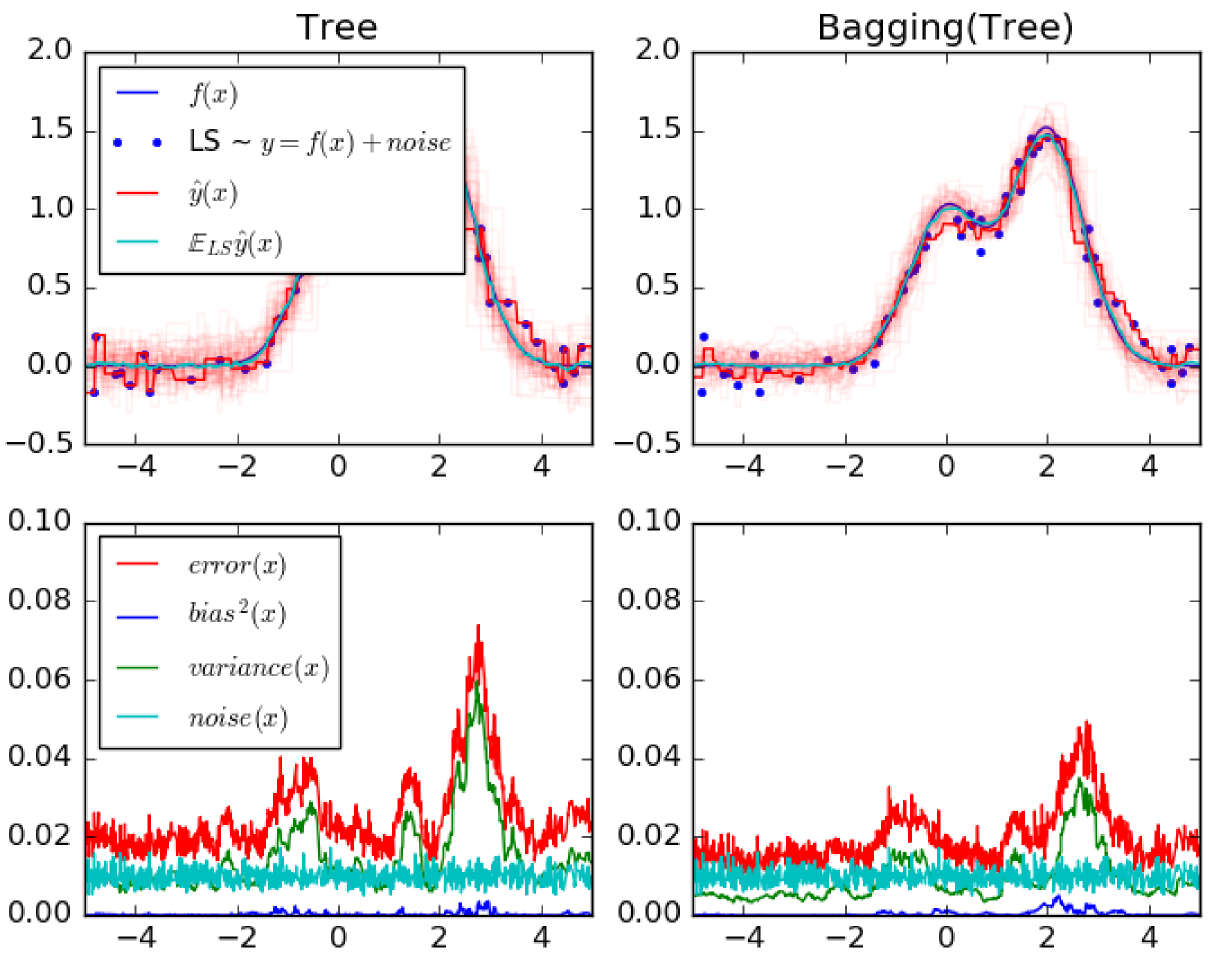
Типы набора выборок из изначальной:

Pasting – просто случайная подвыборка, без повторений.

Bagging – случайная подвыборка (может быть того же размера, что и вся выборка) с повторениями.

Random Subspaces – случайный набор признаков.

Random Patches – случайная подвыборка со случайным набором признаков.



Random Forests (Случайные леса)

Обучим много деревьев на «случайных данных»

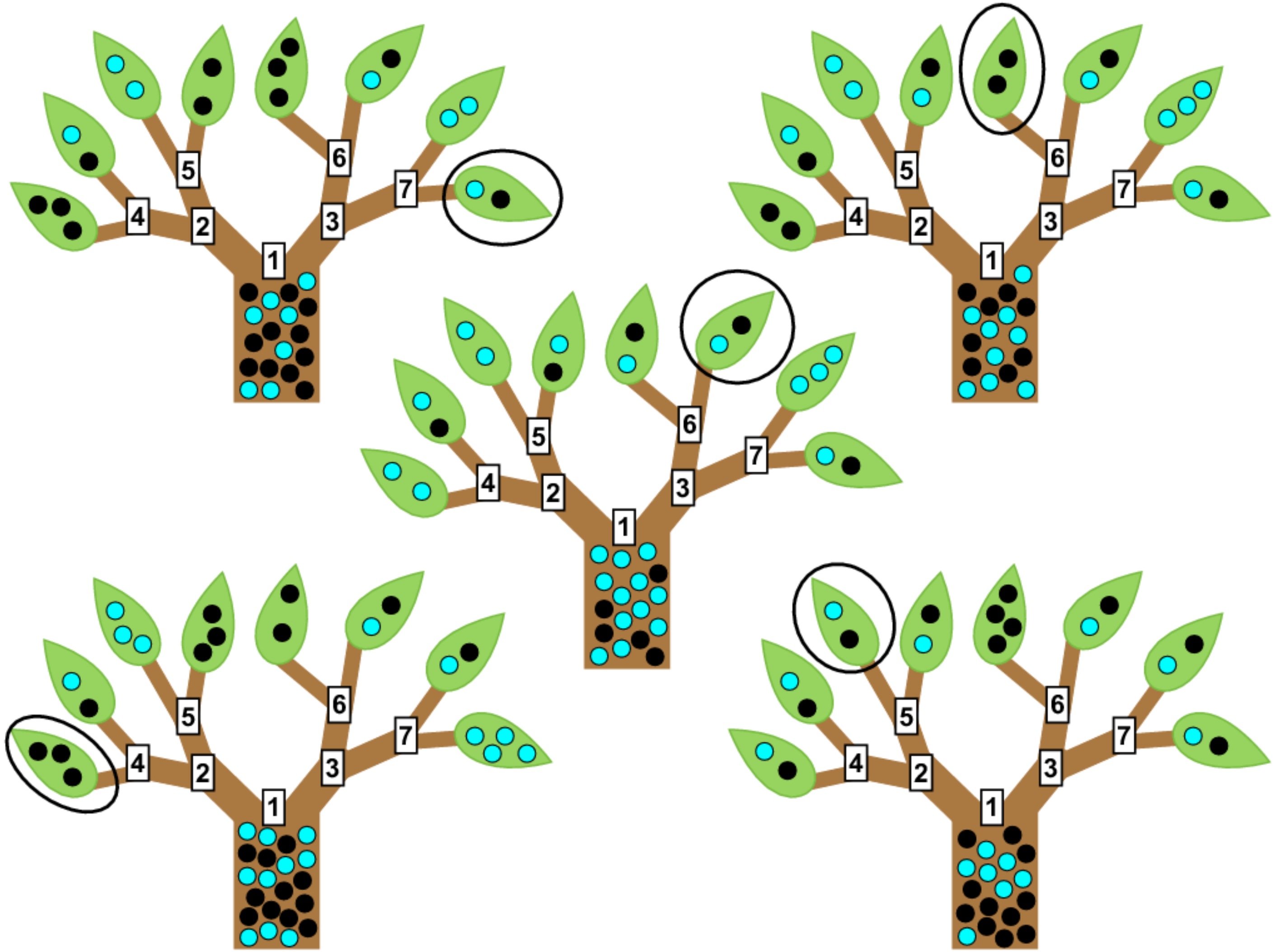
Случайные данные:

1) Датасеты с помощью **bagging (b**ootstrap **agg**regat**ing)** – вытаскиваем из исходного датасета столько же данных, но с повторениями.

2) Берем случайное подмножество признаков.

Результат – голосование, вероятность, логарифм вероятностей.

Параметры – количество используемых примеров и признаков для построения каждого дерева.



Adaptive Boosting (AdaBoost)

Начнем с равномерного распределения: $D_1(i) = \frac{1}{N}$, для $(\mathrm{x}_1, y_1), \dots, (\mathrm{x}_N, y_N)$.

На каждом шаге:

Обучим слабую гипотезу $h_t\colon \mathrm{X} \to \{-1, +1\}$, такую что взвешенная ошибка минимальна:

$$E_t = \sum_{i=1}^{N} D_t(i)\ \left[h_t(\mathrm{x}_i) \neq y_i\right], \qquad \text{вес гипотезы: } \alpha_t = \frac{1}{2}\ln\left(\frac{1 - E_t}{E_t}\right)$$

Меняем веса данных в зависимости текущих ошибок на них:

$$D_{t+1}(i) = D_t(i) e^{-\alpha_t y_i h_t(\mathrm{x}_i)}$$

Итоговая гипотеза – сумма (с коэффициентами) слабых гипотез:

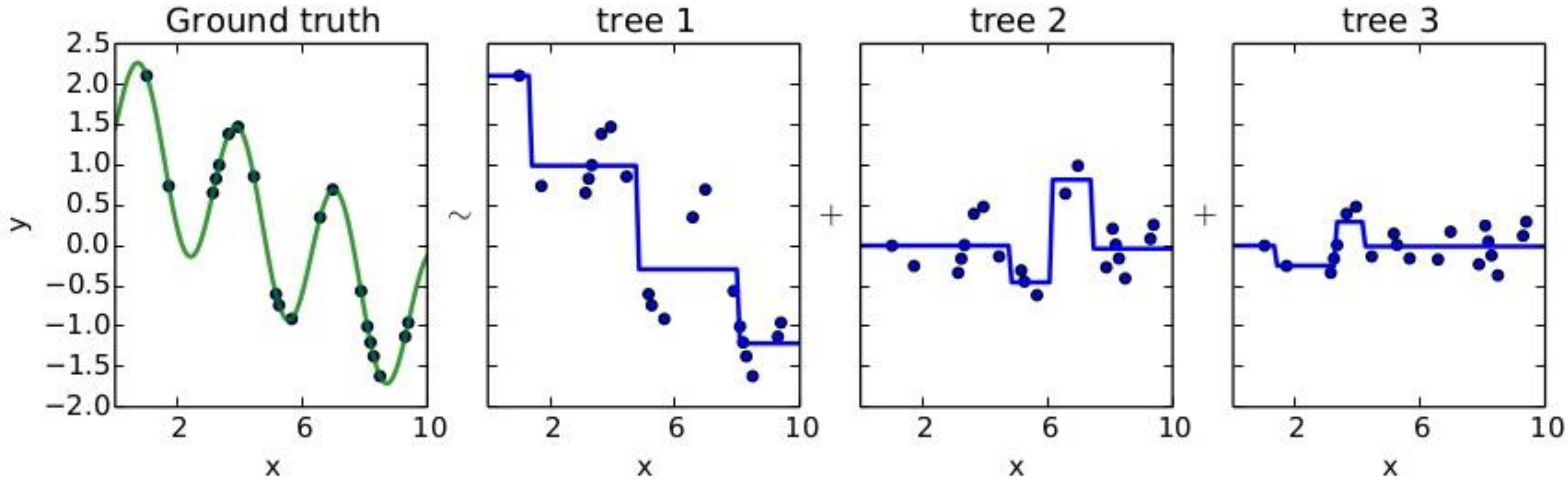
$$H(x) = sign\left(\sum_{t=1}^{T} \alpha_t h_t(\mathrm{x})\right)$$

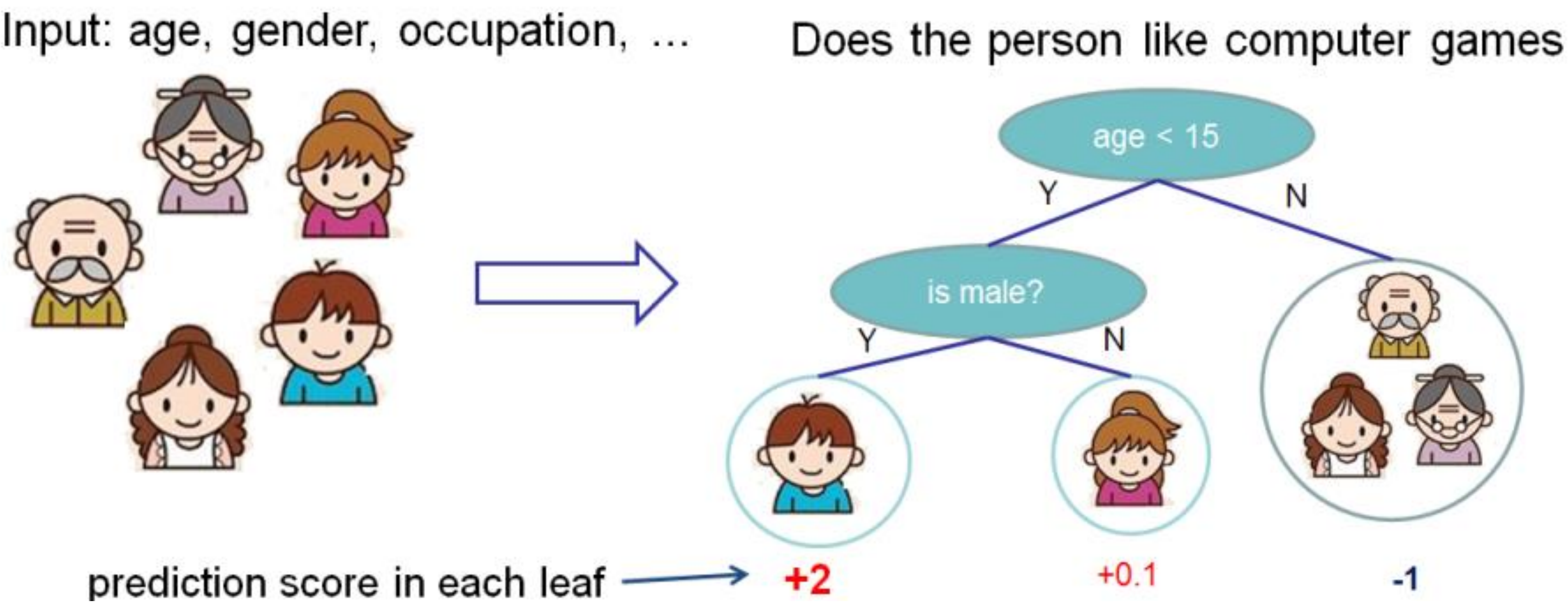
Gradient Boosting

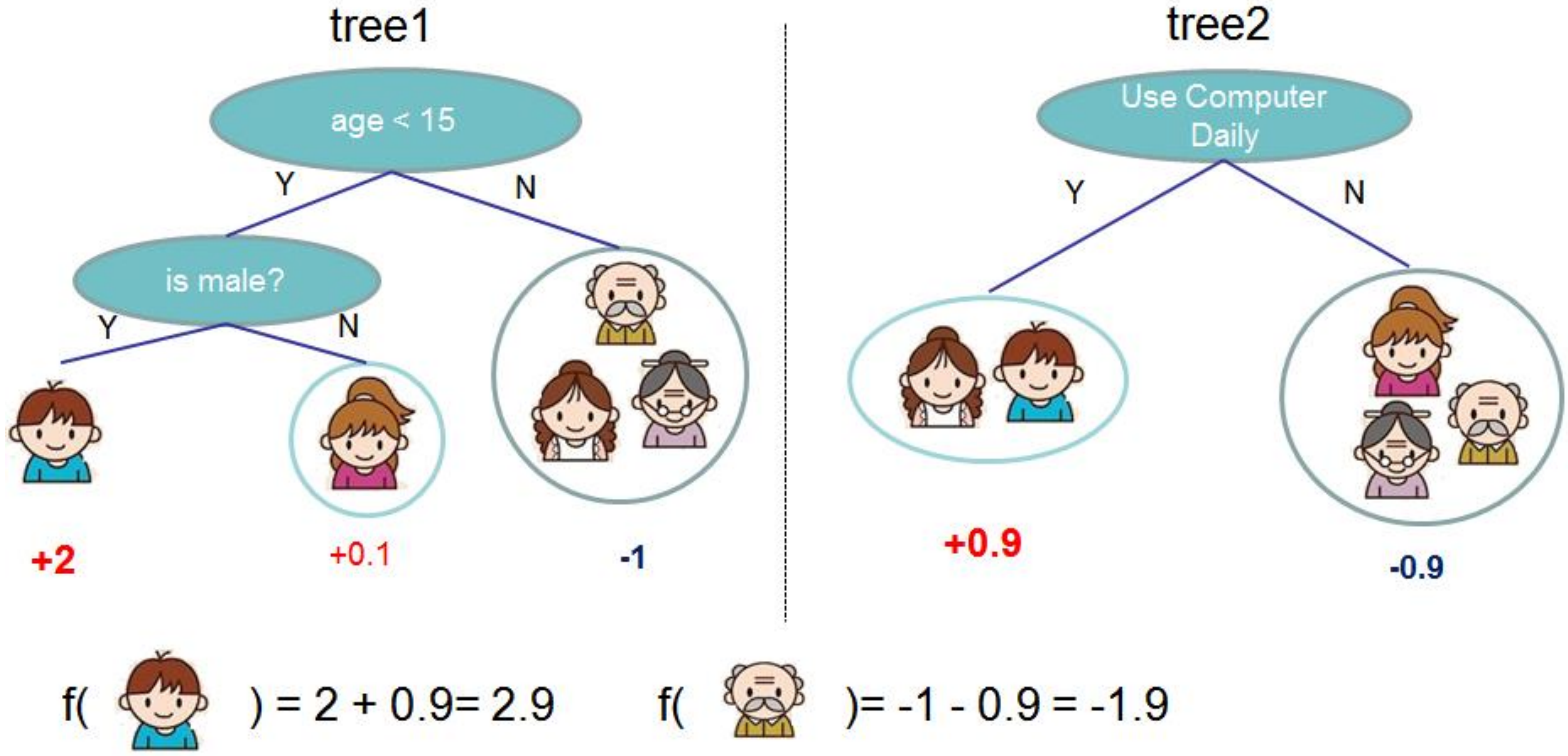
В общем случае:

$$H_{t+1}(\mathrm{x}) = H_t(\mathrm{x}) + h_{t+1}(\mathrm{x}) \to y \;\Rightarrow\; h_{t+1}(\mathrm{x}) \to y - H_t(\mathrm{x})$$

Пример для деревьев:




Input: age, gender, occupation, ...
Does the person like computer games
age < 15
Y
N
is male?
Y
N
prediction score in each leaf
+2
+0.1
-1


tree1
age < 15
Y
N
is male?
Y
N
+2
+0.1
-1
tree2
Use Computer Daily
Y
N
+0.9
-0.9
f( ) = 2 + 0.9= 2.9
f( )= -1 - 0.9 = -1.9

eXtreme Gradient Boosting (XGBoost)

$$H_t(\mathrm{x}) = H_{t-1}(\mathrm{x}) + h_t(\mathrm{x}) = \sum_{j=1}^{t} h_j(\mathrm{x})$$

Минимизируем: $$E_t = \sum_{i=1}^{N} L(H_t(\mathrm{x}_i), y_i) + \sum_{j=1}^{t} \Omega(h_j) = \sum_{i=1}^{N} L\left(\left(H_{t-1}(\mathrm{x}_i) + h(\mathrm{x}_i)\right), y_i\right) + \sum_{j=1}^{t-1} \Omega(h_j) + \Omega(h_t)$$

Регуляризация

XGBoost

$$E_t = \sum_{i=1}^{N} L\left(\left(H_{t-1}(\mathrm{x}_i) + \ h(\mathrm{x}_i)\right), y_i\right) + \sum_{j=1}^{t-1} \Omega(h_j) + \Omega(h_t) = \sum_{i=1}^{N} \left(2(H_{t-1}(\mathrm{x}_i) - y_i)h_t(\mathrm{x}_i) + \left(h_t(\mathrm{x}_i)\right)^2\right) + \Omega(h_t) + const$$

В общем случае:
$$E_t = \sum_{i=1}^{N} \left(L(H_{t-1}(\mathrm{x}_i), y_i) + u_i h_t(\mathrm{x}_i) + \frac{1}{2} v_i \left(h_t(\mathrm{x}_i)\right)^2\right) + \Omega(h_t) + const,$$

$$u_i = \partial_{H_{t-1}(\mathrm{x}_i)}\left(L(H_{t-1}(\mathrm{x}_i), y_i)\right)$$

$$v_i = \partial^2_{H_{t-1}(\mathrm{x}_i)}\left(L(H_{t-1}(\mathrm{x}_i), y_i)\right)$$

Минимизируем:
$$E_t = \sum_{i=1}^{N} \left(u_i h_t(\mathrm{x}_i) + \frac{1}{2} v_i \left(h_t(\mathrm{x}_i)\right)^2\right) + \Omega(h_t)$$

Для MSE:
$$E_t = \sum_{i=1}^{N} \left(\left(H_{t-1}(\mathrm{x}_i) + \ h_t(\mathrm{x}_i)\right) - y_i\right)^2 + \sum_{j=1}^{t} \Omega(h_j) = \sum_{i=1}^{N} \left(2(H_{t-1}(\mathrm{x}_i) - y_i)h_t(\mathrm{x}_i) + \left(h_t(\mathrm{x}_i)\right)^2\right) + \Omega(h_t) + const$$

XGBoost

$$\Omega(f) = \gamma M + \frac{1}{2}\lambda\sum_{j=1}^{M} w_j^2\,, M\ -\text{количество листьев}, w\ -\text{коэффициент листа}$$

$$E_t = \sum_{i=1}^{N}\left(u_i w_{q(\mathrm{x}_i)} + \frac{1}{2}v_i\left(w_{q(\mathrm{x}_i)}\right)^2\right) + \gamma M + \frac{1}{2}\lambda\sum_{j=1}^{M} w_j^2$$

Где $q(\mathrm{x}_i)$ – лист, к которому принадлежит x_i.

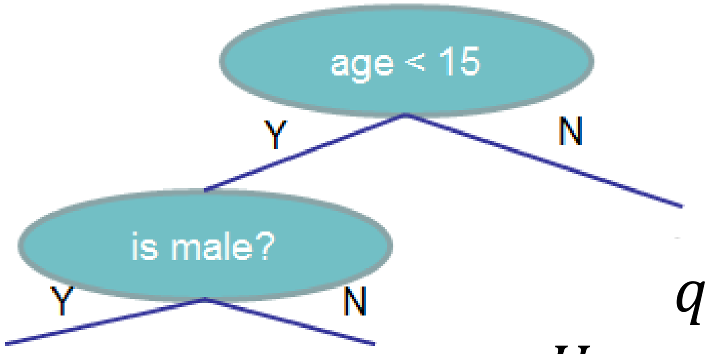
Сгруппируем по листьям:

$$E_t = \sum_{j=1}^{M}\left(\sum_{q(\mathrm{x}_i)=j} u_i\, w_j + \frac{1}{2}\left(\sum_{q(\mathrm{x}_i)=j} v_i + \lambda\right)w_j^2\right) + \gamma M = \sum_{j=1}^{M}\left(U_j w_j + \frac{1}{2}\left(V_j + \lambda\right)w_j^2\right) + \gamma M$$

$$U_j = \sum_{q(\mathrm{x}_i)=j} u_i \qquad V_j = \sum_{q(\mathrm{x}_i)=j} v_i$$

XGBoost

Instance index	gradient statistics
1	u_1, v_1
2	u_2, v_2
3	u_3, v_3
4	u_4, v_4
5	u_5, v_5



$$q(\mathrm{x}) = 1 \quad U_1 = u_1 \quad V_1 = v_1$$

$$q(\mathrm{x}) = 2 \quad U_2 = u_2 \quad V_1 = v_2$$

$$q(\mathrm{x}) = 3 \quad U_3 = u_3 + u_4 + u_5 \quad V_3 = v_3 + v_4 + v_5$$

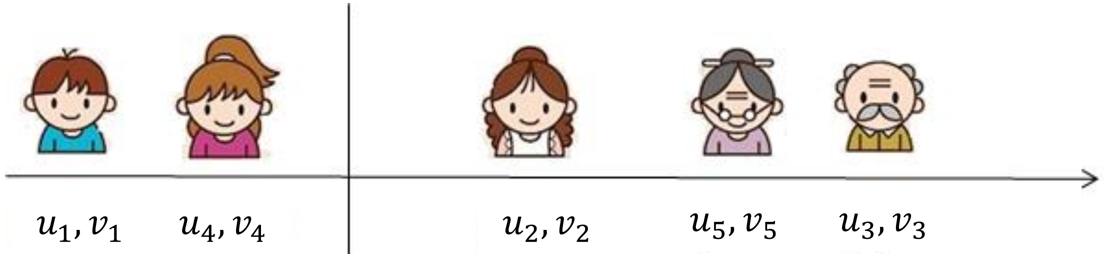
$$U_j = \sum_{q(\mathrm{x}_i)=j} u_i \qquad V_j = \sum_{q(\mathrm{x}_i)=j} v_i$$

XGBoost

$$E_t = \sum_{j=1}^{M}\left(U_j w_j + \frac{1}{2}(V_j + \lambda) w_j^2\right) + \gamma M$$

$$w_j^{opt} = -\frac{U_j}{V_j + \lambda} \qquad E_t^{opt} = -\frac{1}{2}\sum_{j=1}^{M}\frac{U_j^2}{V_j + \lambda} + \gamma M$$

$$Gain = \frac{1}{2}\left[\frac{U_L^2}{V_L + \lambda} + \frac{U_R^2}{V_R + \lambda} - \frac{(U_L + U_R)^2}{V_L + V_R + \lambda}\right] - \gamma$$



$$U_L = u_1 + u_4 \qquad U_R = u_2 + u_3 + u_5$$

$$V_L = v_1 + v_4 \qquad V_R = v_2 + v_3 + v_5$$