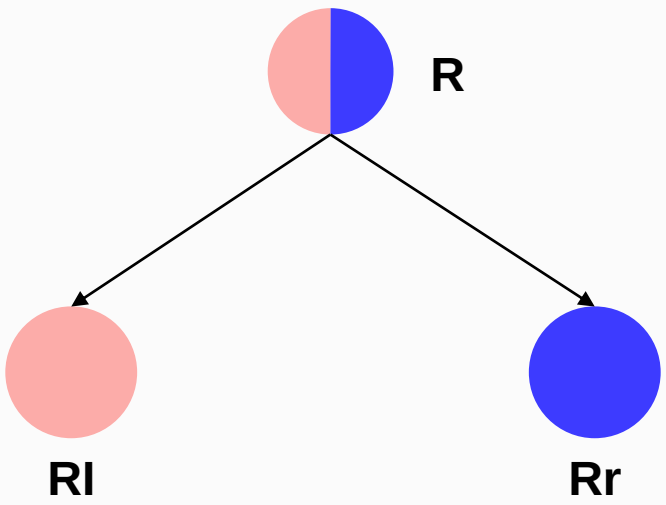
Обучение решающего дерева

Skillbox

Елена Кантонистова

Идея построения дерева

- Подобрать такой предикат, то есть такое условие разбиения выборки на две части, чтобы после разбиения, в идеале, объекты одного класса попали в одну подветку дерева, а объекты другого класса — в другую



- Цель — снизить перемешанность или неоднородность по классам внутри каждой следующей вершины

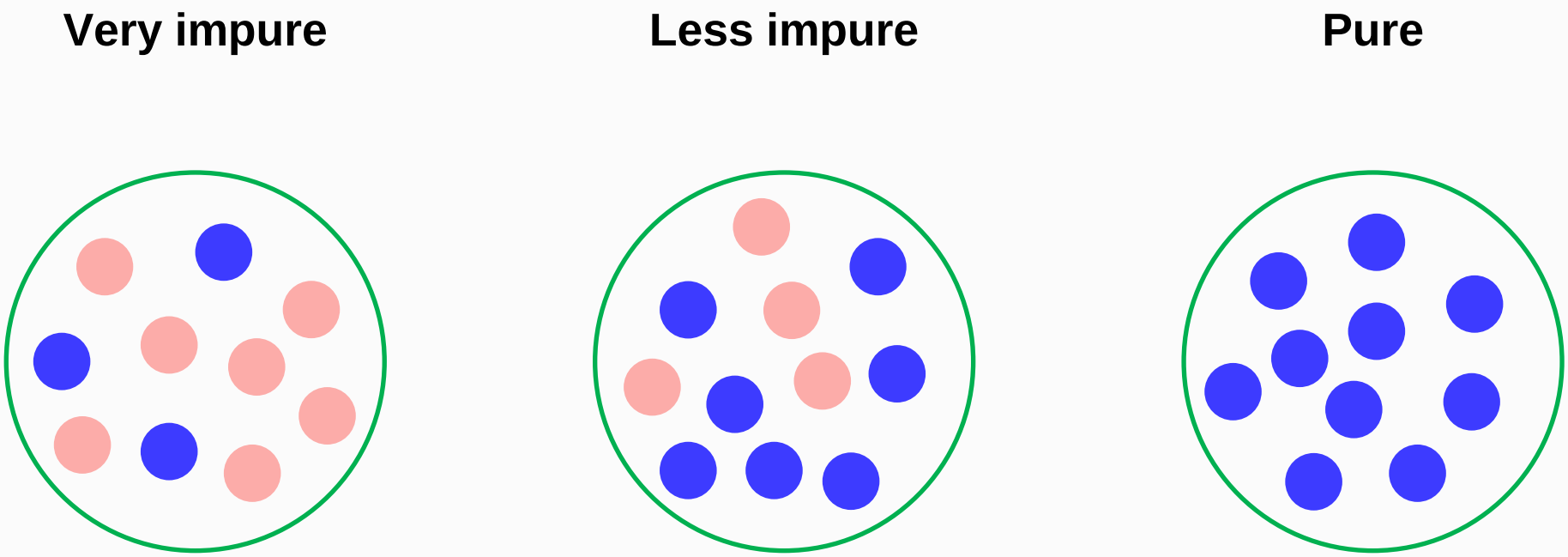
Автор: Кантонистова Елена

Критерий информативности

- Ввести функцию $H(R)$, которая будет измерять неоднородность объектов в вершине — критерий информативности
- Если в вершине половина объектов одного класса, а половина другого класса, то это максимальная неоднородность, и значение функции H будет большим; если же все объекты в вершине одного класса, это максимально однородная выборка, и значение функции H будет минимальным, а именно, 0

Критерий информативности

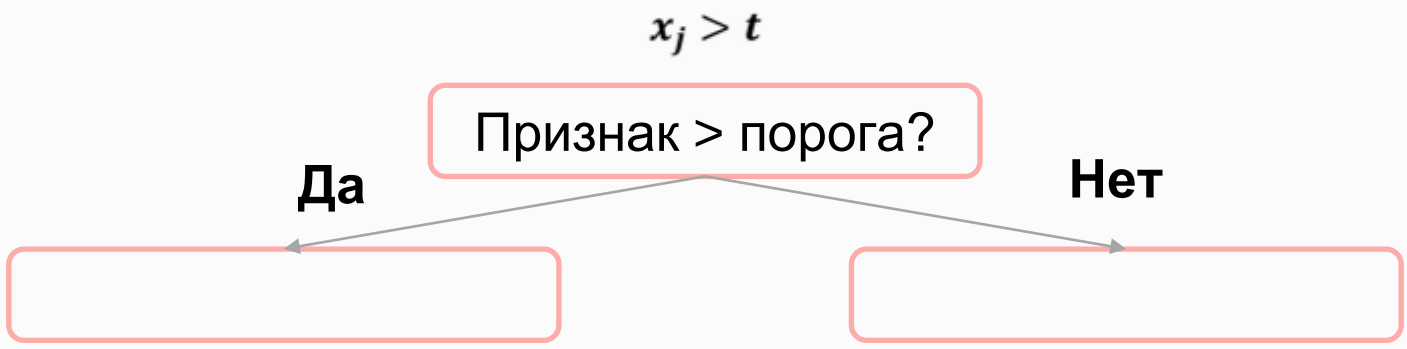
Примеры однородных и неоднородных выборок:



Автор: Кантонистова Елена

Критерий информативности

- Ввести функцию $H(R)$, которая будет измерять неоднородность объектов в вершине — критерий информативности
- Подобрать такой предикат, то есть такую пару (признак, порог), что при разбиении выборки на две части условием



В каждой из двух полученных вершин R_l и R_r значение функции H минимально.

Автор: Кантонистова Елена

Задача оптимизации

- Подобрать такой предикат, то есть такую пару (признак, порог), что при разбиении выборки на две части условием

$$x_j > t$$

- В каждой из двух полученных вершин R_l и R_r значение функции минимально
- То есть

$$\boldsymbol{H(R_l) \to \min_{j,t}, H(R_r) \to \min_{j,t}}$$

Задача оптимизации

- Подобрать такой предикат, то есть такую пару (признак, порог), что при разбиении выборки на две части условием

$$x_j > t$$

- В каждой из двух полученных вершин R_l и R_r значение функции минимально
- То есть

$$H(R_l) \to \min_{j,t}, H(R_r) \to \min_{j,t}$$

- Удобнее решать одну задачу оптимизации вместо двух:

$$\boldsymbol{H(R_l) + H(R_r) \to \min_{j,t}}$$

Задача оптимизации

- Подобрать такой предикат, то есть такую пару (признак, порог), что при разбиении выборки на две части условием

$$x_j > t$$

- В каждой из двух полученных вершин R_l и R_r значение функции минимально
- То есть

$$H(R_l) \to \min_{j,t}, H(R_r) \to \min_{j,t}$$

- Удобнее решать одну задачу оптимизации вместо двух:

$$\boldsymbol{H(R_l) + H(R_r) \to \min_{j,t}}$$

- Эквивалентная задача:

$$\boldsymbol{Q(R, j, t) = H(R) - H(R_l) - H(R_r) \to \max_{j,t}}$$

Information Gain (IG)

- Эквивалентная задача:

$$Q(R, j, t) = H(R) - H(R_l) - H(R_r) \to \max_{j,t}$$

Величина $Q(R, j, t)$ называется Information Gain (прирост информации).

- $Q(R, j, t)$ означает, на сколько вы упорядочили объекты (т. е. классы) после разбиения по условию $x_j > t$

Information Gain (IG)

- Уточнение:

$$Q(R, j, t) = H(R) - \frac{|\boldsymbol{R_l}|}{|\boldsymbol{R}|} H(R_l) - \frac{|\boldsymbol{R_l}|}{|\boldsymbol{R}|} H(R_r) \to \max_{j,t}$$

$|R|, |R_l|, |R_r|$ — количество объектов в исходной вершине R и в двух новых вершинах, полученных после разбиения.

Итоги

- Вы узнали, какая задача оптимизации решается на каждом этапе построения решающего дерева.
  А именно, ищется такой признак x_j и такой порог t, что при разбиении объектов на две группы условием $x_j > t$ значение Information Gain

$$Q(R, j, t) = H(R) - \frac{|R_l|}{|R|} H(R_l) - \frac{|R_l|}{|R|} H(R_r)$$

максимально