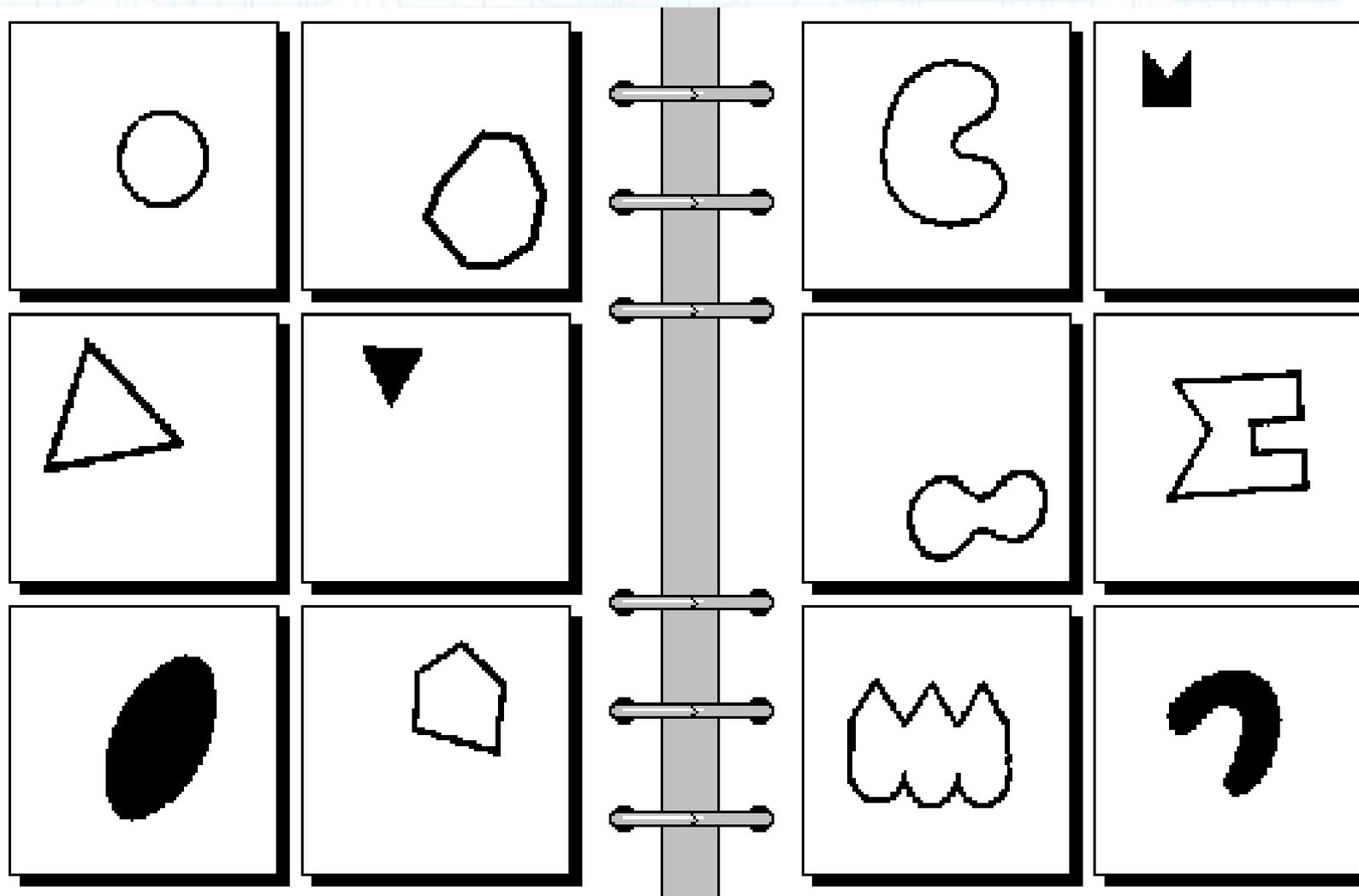


# Машинное обучение

## Логические методы классификации



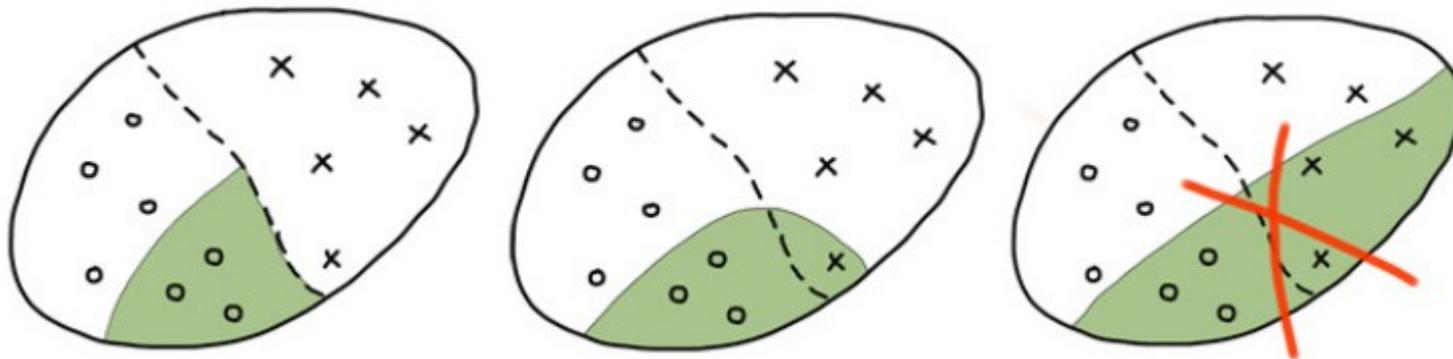
# Содержание лекции

- Понятие закономерности
- Критерий качества закономерностей
- Поиск закономерностей
- Алгоритмы классификации на основе логических закономерностей

# Понятие закономерности

- Предикат  $R: X \rightarrow \{0, 1\}$  – закономерность, если он выделяет ( $R(x)=1$ ) достаточно много объектов одного класса  $C$  и практически не выделяет объектов других классов

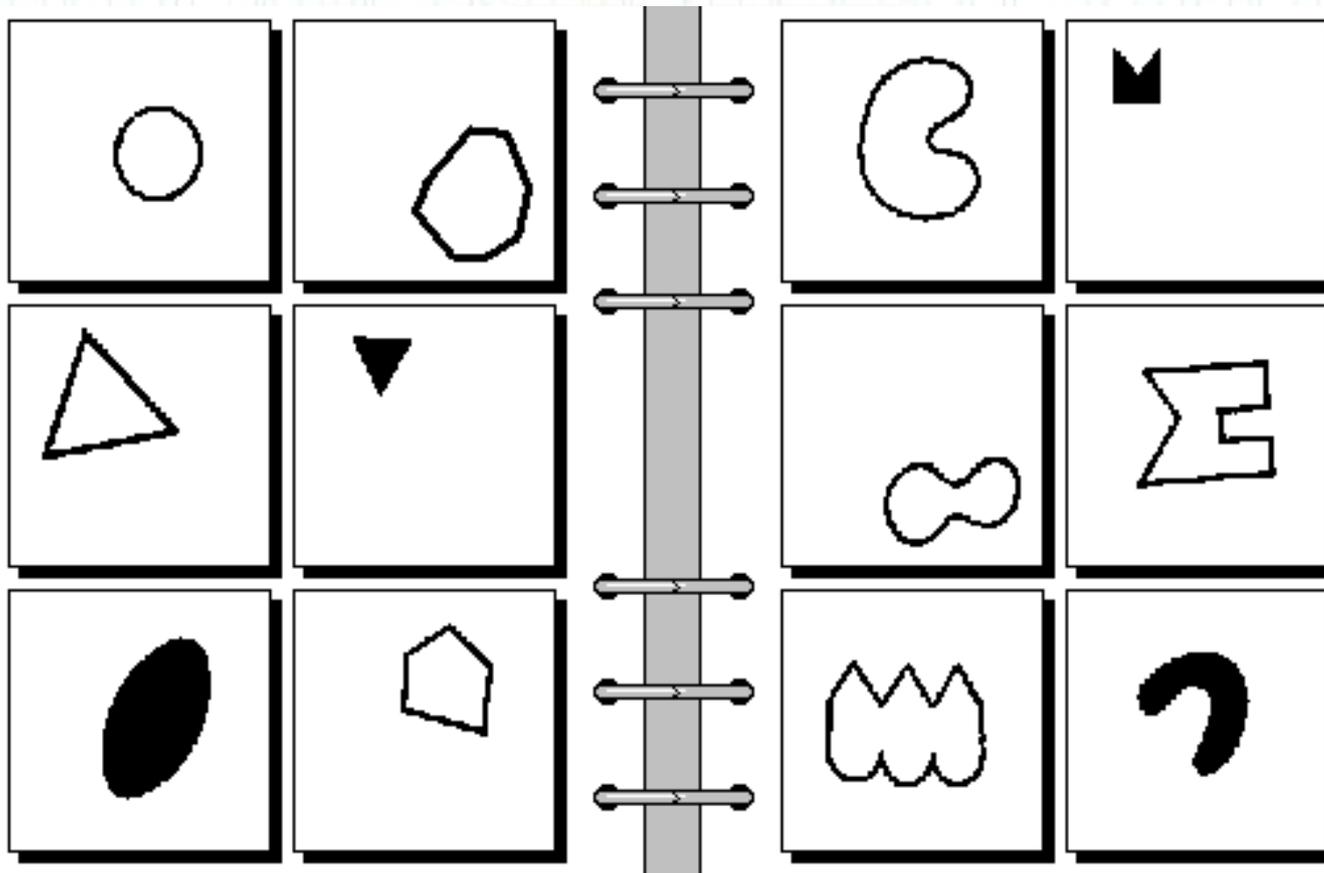
$$p_c(R) = \#\{x_i : R(x_i)=1 \text{ и } y_i=c\} \rightarrow \max;$$
$$n_c(R) = \#\{x_i : R(x_i)=1 \text{ и } y_i \neq c\} \rightarrow \min;$$



# Примеры

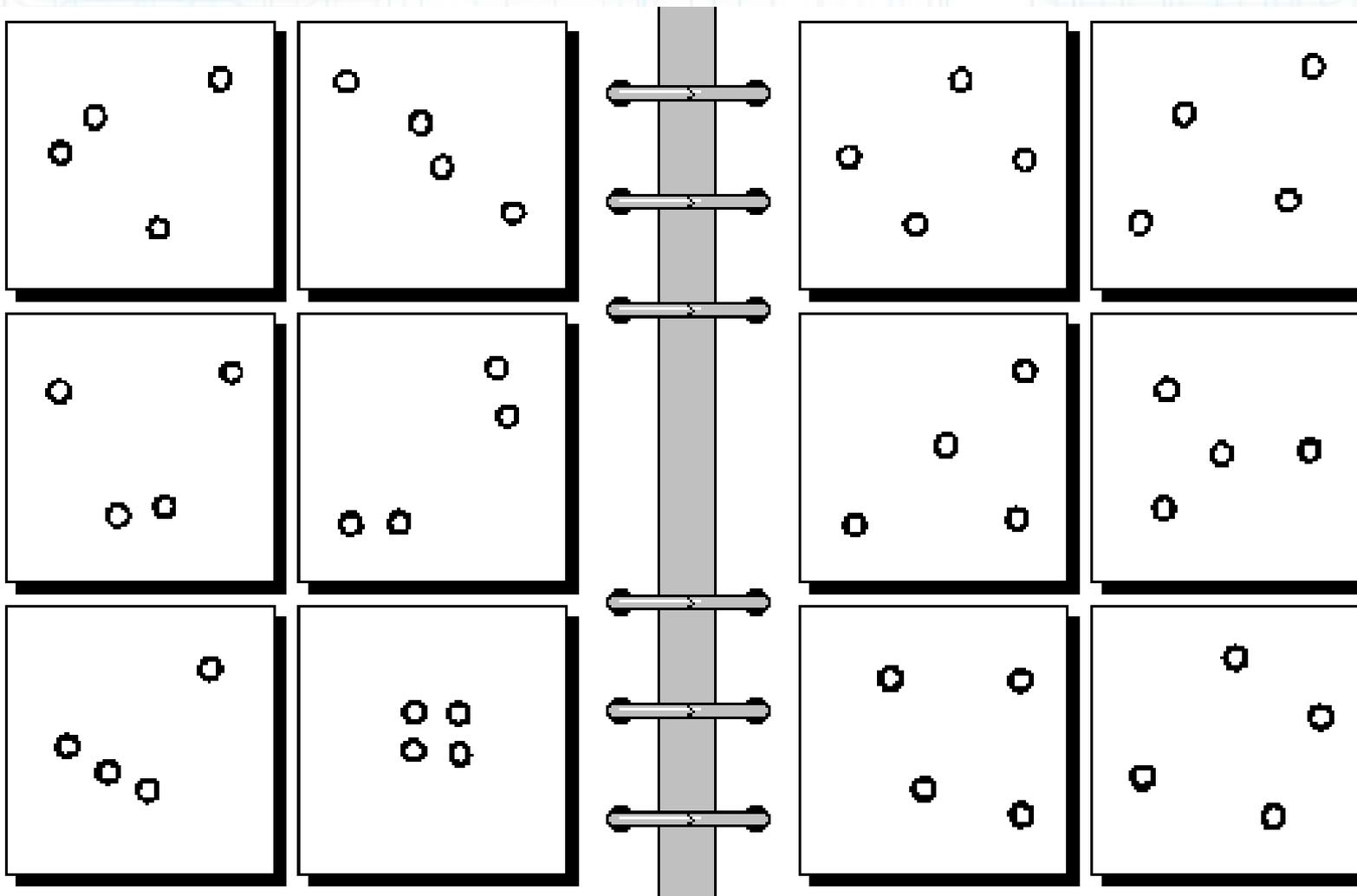
- Если «возраст  $> 60$ » и «пациент ранее перенёс инфаркт», то операцию не делать, риск отрицательного исхода 60%.
- Если «в анкете указан домашний телефон» и «зарплата  $> \$2000$ » и «сумма кредита  $< \$5000$ » то кредит можно выдать, риск дефолта 5%.

# Тесты М.М.Бонгарда

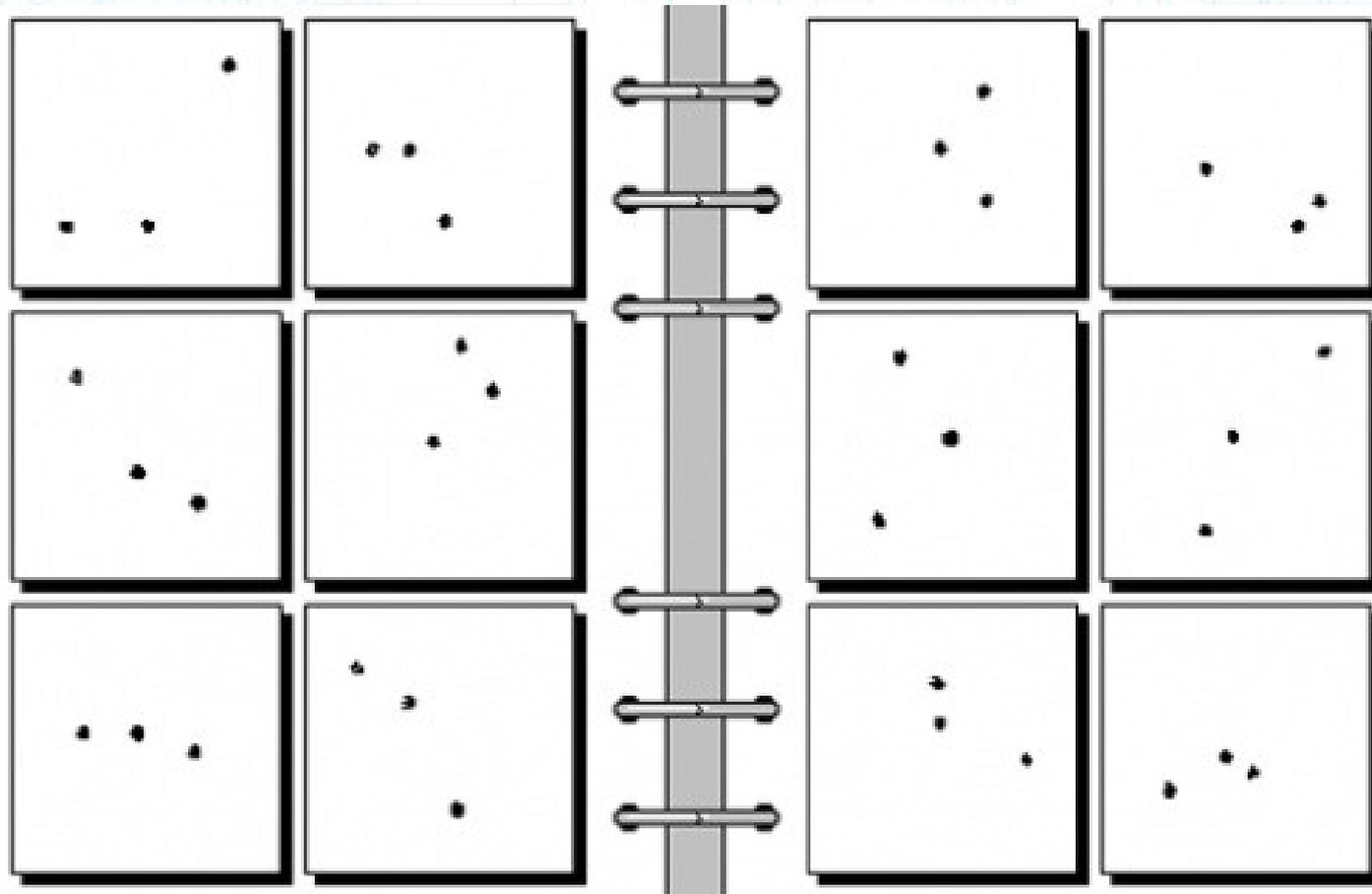


Этот тип головоломки изобрёл выдающийся русский кибернетик, основоположник теории распознавания образов Михаил Моисеевич Бонгард: в 1967-м году он впервые опубликовал одну из них в своей книге "Проблема узнавания"

# Тесты М.М.Бонгарда



# Тесты М.М.Бонгарда



# Как сравнивать закономерности?

$$\begin{cases} p(R) \rightarrow \max \\ n(R) \rightarrow \min \end{cases} \xRightarrow{?} I(p, n) \rightarrow \max$$

$$I(p, n) = \frac{p}{p+n} \rightarrow \max \quad (\text{precision});$$

$$I(p, n) = p - n \rightarrow \max \quad (\text{accuracy});$$

$$I(p, n) = p - Cn \rightarrow \max \quad (\text{linear cost accuracy});$$

$$I(p, n) = \frac{p}{P} - \frac{n}{N} \rightarrow \max \quad (\text{relative accuracy});$$

$P_c = \#\{x_i: y_i=c\}$  — число «своих» во всей выборке;

$N_c = \#\{x_i: y_i \neq c\}$  — число «чужих» во всей выборке.

# Как сравнивать закономерности?

при  $P = 200$ ,  $N = 100$  и различных  $p$  и  $n$

$p$	$n$	$p/(p+n)$	$p-n$	$p-5n$	$p/P-n/N$
10	0	1	10	10	0,05
200	10	0,95	190	150	0,9
10	0	1	10	10	0,05
60	50	0,55	10	-190	-0,2
200	40	0,83	160	0	0,6
5	1	0,83	4	0	0,02
10	0	1	10	10	0,05
200	95	0,68	105	-275	0,05

# Вероятностный подход

- Рассмотрим опыт – отбор предикатом объектов обучающей выборки
- Предикат – закономерность  $T$  к событиям: “объект отобран предикатом” и “объект имеет класс  $c$ ” зависимы
- Качество закономерности = мера зависимости случайных событий

# Точный тест Фишера

- Предположим, что события “объект отобран предикатом” и “объект имеет класс  $c$ ” независимы
- Тогда вероятность отобрать  $r$  объектов класса  $c$  и  $n - r$  – других классов:

# Точный тест Фишера

- Предположим, что события “объект отобран предикатом” и “объект имеет класс  $c$ ” независимы
- Тогда вероятность отобрать  $p$  объектов класса  $c$  и  $n$  – других классов: 
$$\frac{C_P^p C_N^n}{C_{P+N}^{p+n}}$$
- Это правдоподобие гипотезы независимости событий. Чем меньше данная вероятность, тем более зависимы события

$$IStat(p, n) = -\frac{1}{\ell} \log_2 \frac{C_P^p C_N^n}{C_{P+N}^{p+n}} \rightarrow \max$$

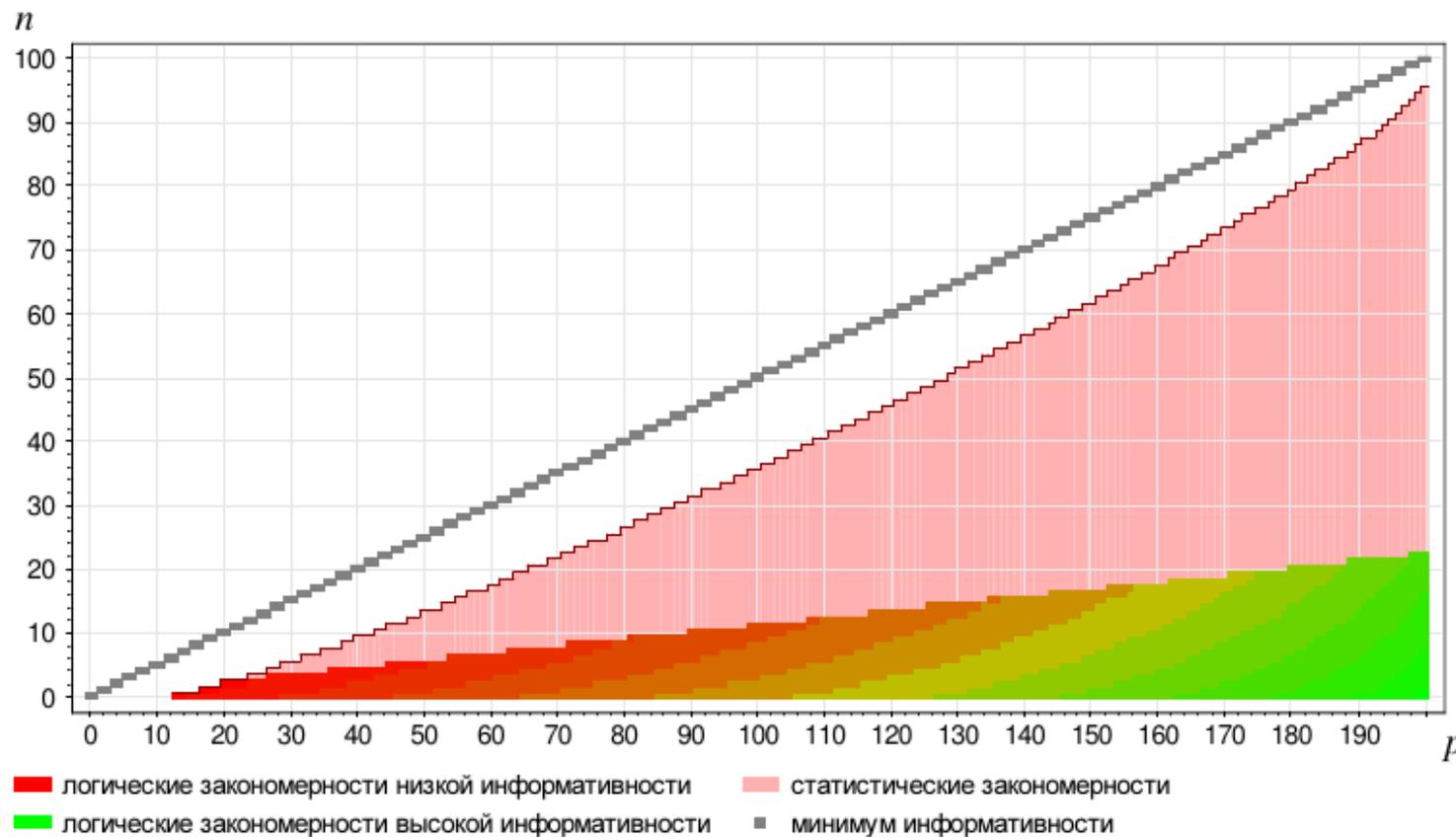
# Точный тест Фишера

$p$	$n$	$p/(p+n)$	$p-n$	$p-5n$	$p/P-n/N$	Вероятность
10	0	<b>1</b>	10	10	0,05	0,016
200	10	<b>0,95</b>	190	150	0,9	8,80E-66
10	0	<b>1</b>	<b>10</b>	10	0,05	0,016
60	50	0,55	<b>10</b>	-190	-0,2	3,50E-04
200	40	0,83	160	<b>0</b>	0,6	1,50E-36
5	1	0,83	4	<b>0</b>	0,02	0,26
10	0	<b>1</b>	10	10	<b>0,05</b>	0,016
200	95	0,68	105	-275	<b>0,05</b>	0,0038

# Логические и статистические закономерности

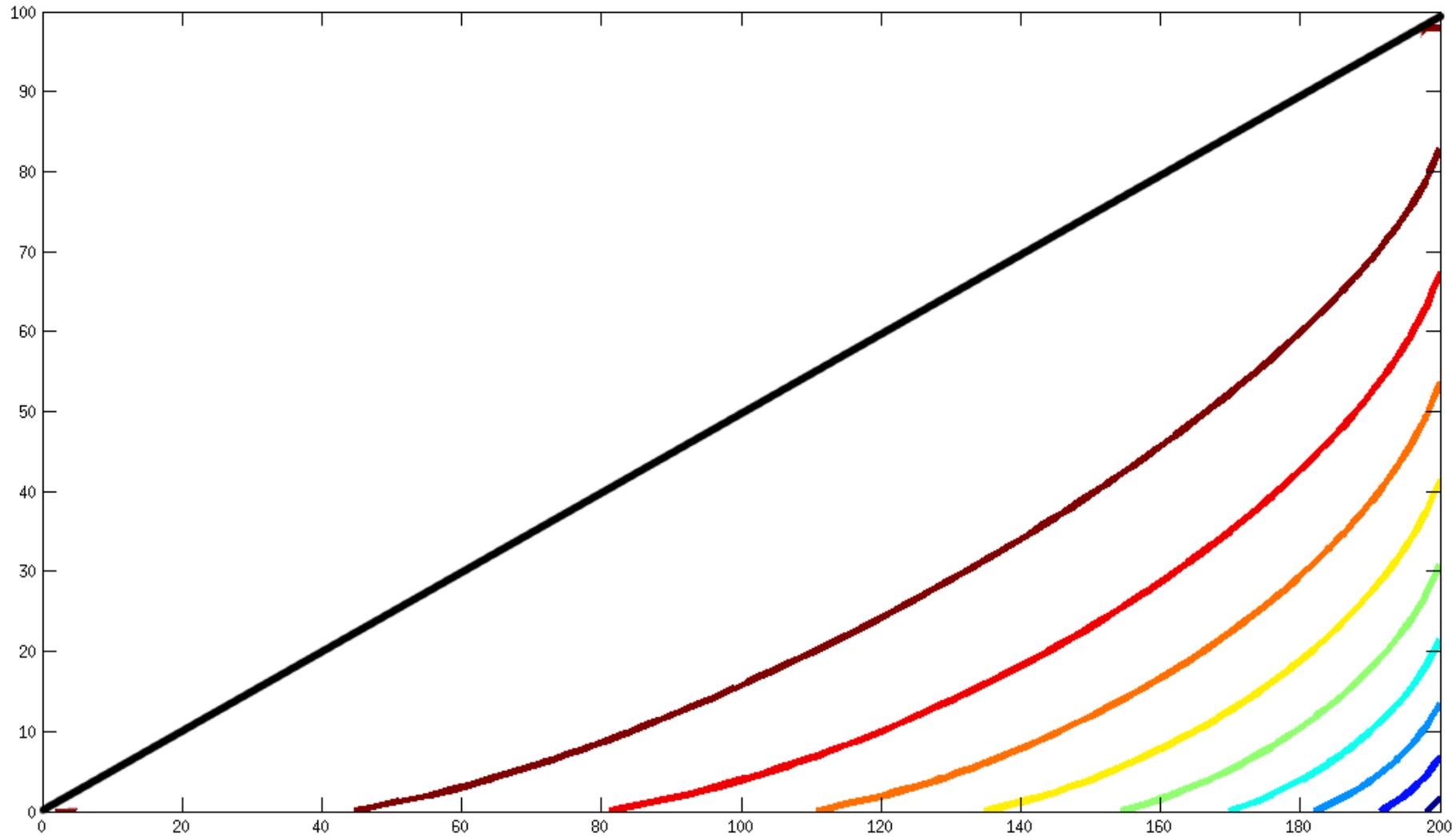
Логические закономерности:  $\frac{n}{p+n} \leq 0.1$ ,  $\frac{p}{P+N} \geq 0.05$ .

Статистические закономерности:  $IStat(p, n) \geq 3$ .



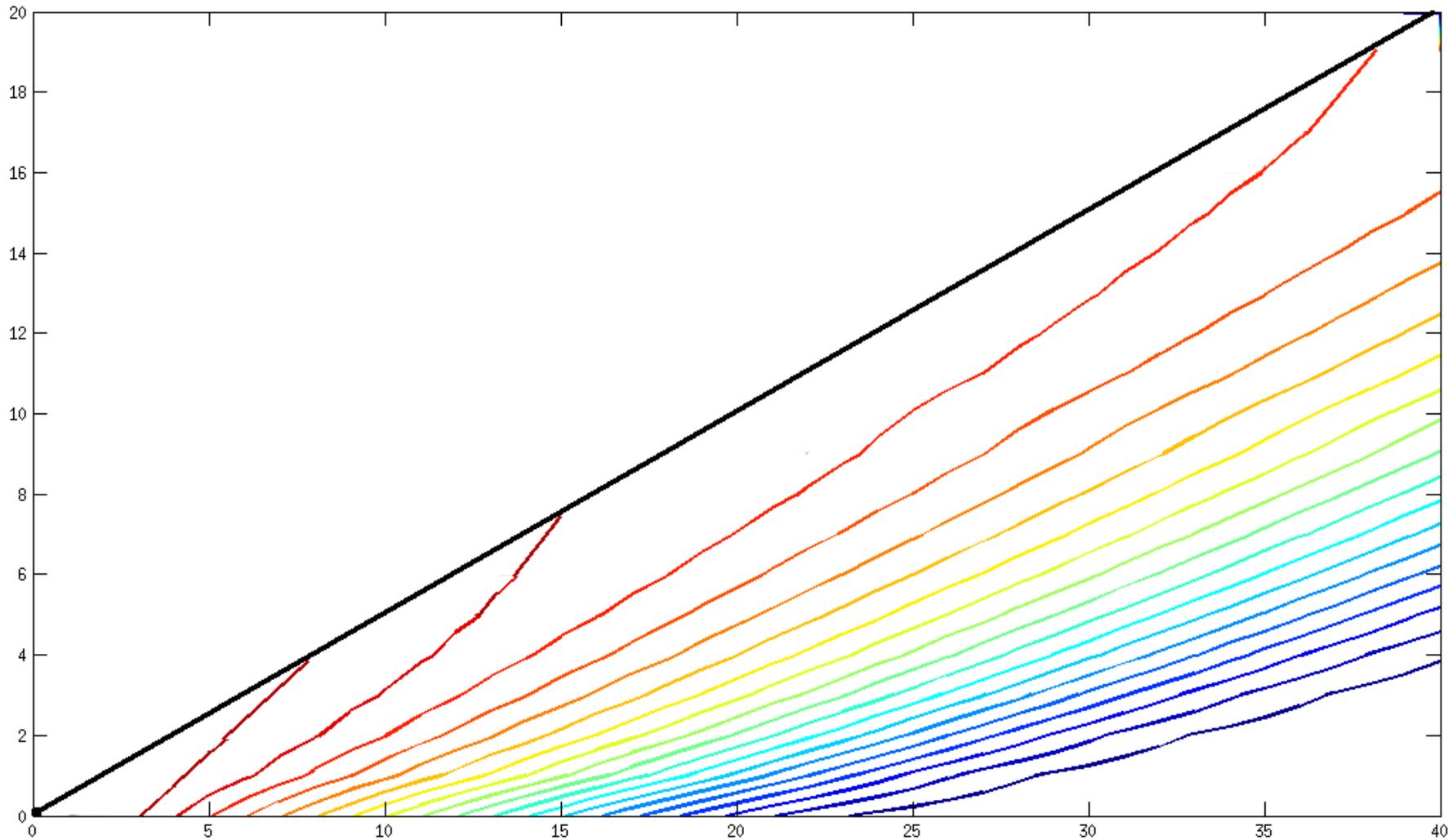
$P = 200$   
 $N = 100$

# Линии уровня теста Фишера



# Линии уровня теста Фишера

## Малые $p$ и $n$



# Энтропийный критерий информативности

Пусть  $\omega_0, \omega_1$  — два исхода с вероятностями  $q$  и  $1 - q$ .

Количество информации:  $I_0 = -\log_2 q$ ,  $I_1 = -\log_2(1 - q)$ .

Энтропия — математическое ожидание количества информации:

$$h(q) = -q \log_2 q - (1 - q) \log_2(1 - q).$$

Энтропия выборки  $X^\ell$ , если исходы — это классы  $y=c$ ,  $y \neq c$ :

$$H(y) = h\left(\frac{P}{\ell}\right).$$

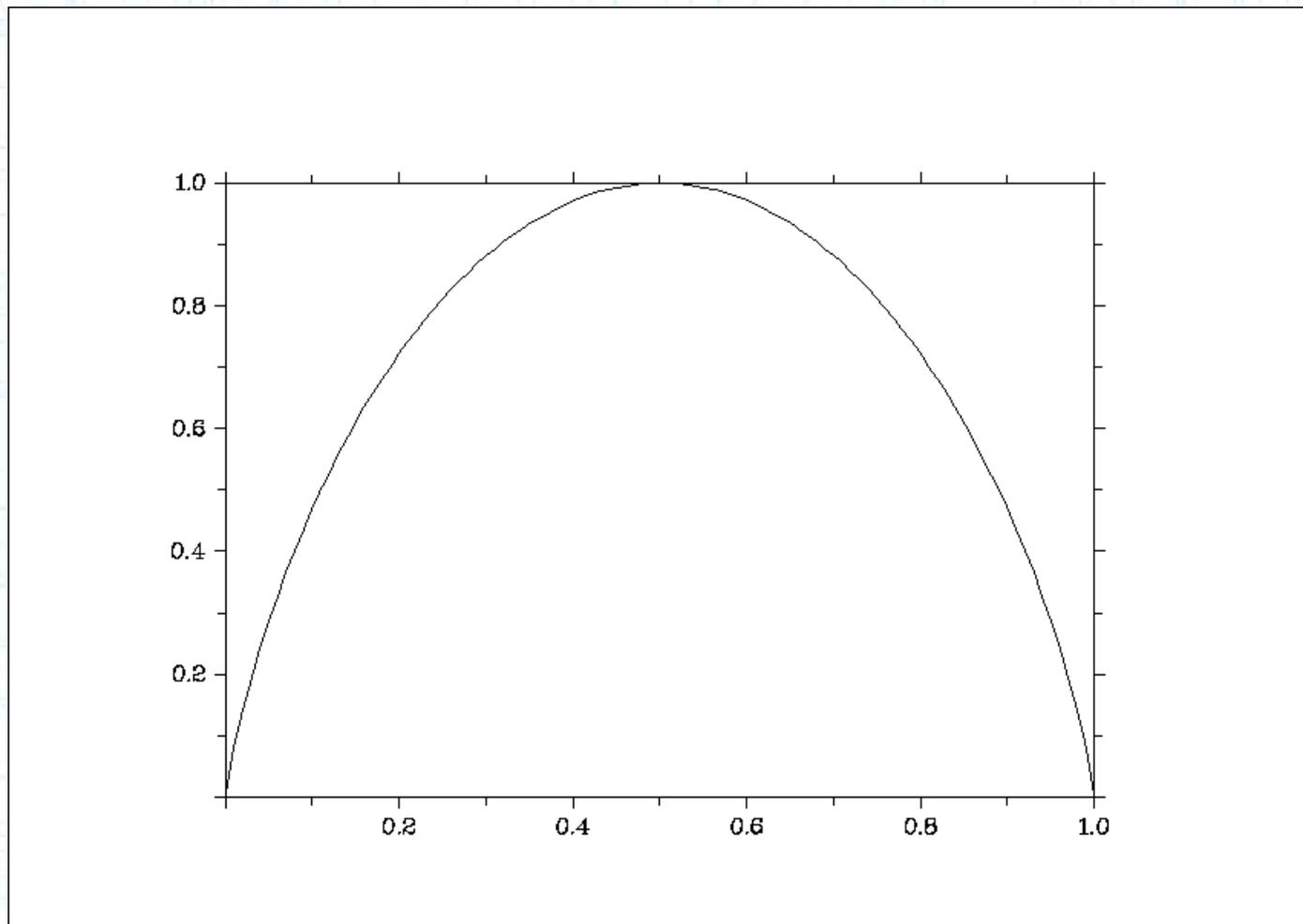
Энтропия выборки  $X^\ell$  после получения информации  $R(x_i)_{i=1}^\ell$ :

$$H(y|R) = \frac{p+n}{\ell} h\left(\frac{p}{p+n}\right) + \frac{\ell-p-n}{\ell} h\left(\frac{P-p}{\ell-p-n}\right).$$

Прирост информации (Information gain, IGain):

$$\text{IGain}(p, n) = H(y) - H(y|R).$$

# Энтропия для различных $q$



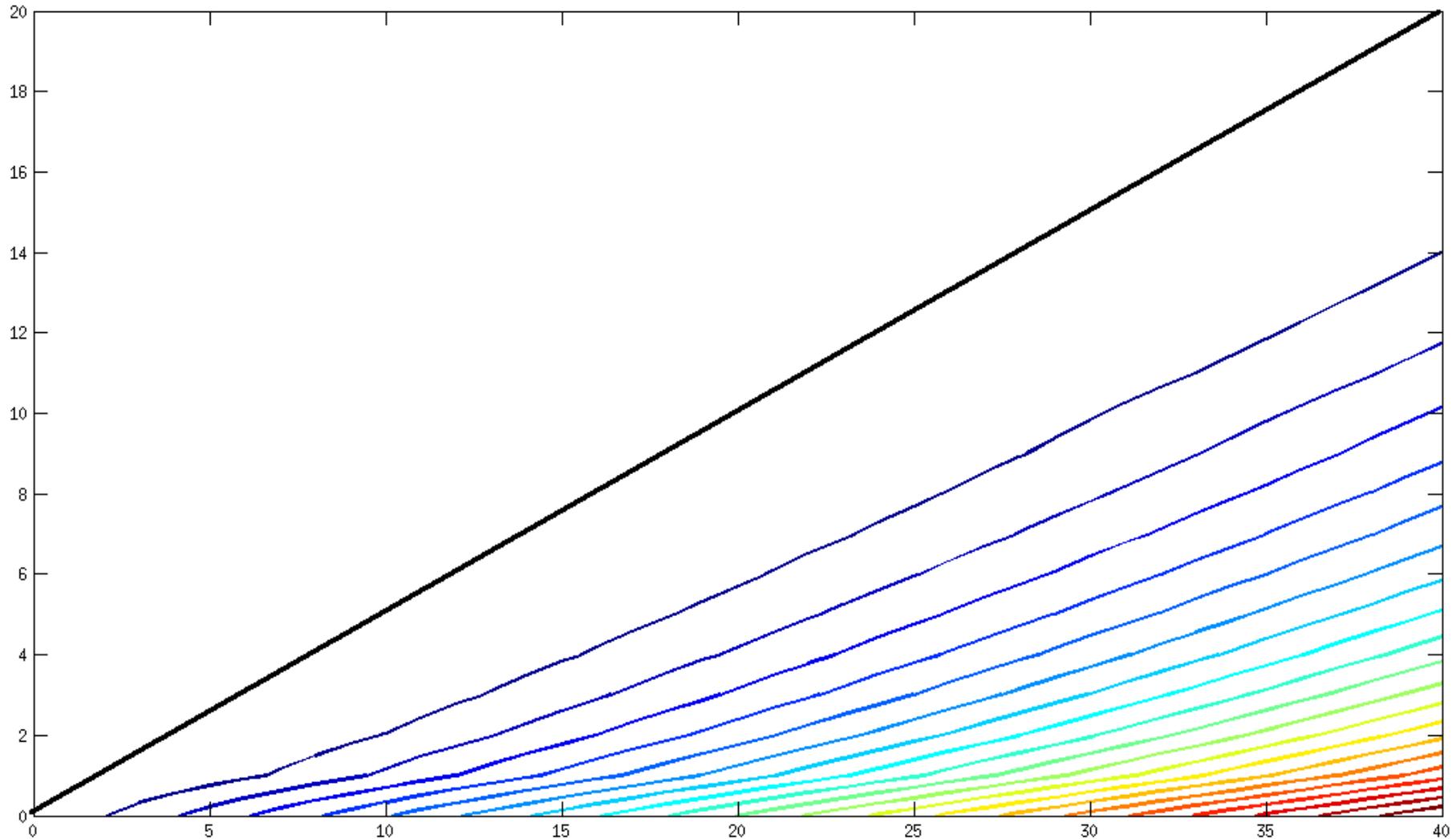
# Соотношение статистического и энтропийного критериев

Энтропийный критерий  $I\text{Gain}$  асимптотически эквивалентен статистическому  $I\text{Stat}$ :

$$I\text{Stat}(p, n) \rightarrow I\text{Gain}(p, n) \quad \text{при } \ell \rightarrow \infty$$

Доказательство: применить формулу Стирлинга к критерию  $I\text{Stat}$ .

# Линии уровня энтропийного критерия. Малые $\rho$ и $n$



# Построение закономерностей (rule induction)

1. Пороговое условие (решающий пень, decision stump):

$$R(x) = [f_j(x) \leq a_j] \text{ или } [a_j \leq f_j(x) \leq b_j].$$

2. Конъюнкция пороговых условий:

$$R(x) = \bigwedge_{j \in J} [a_j \leq f_j(x) \leq b_j].$$

3. Синдром — выполнение не менее  $d$  условий из  $J$ ,  
(при  $d = |J|$  это конъюнкция, при  $d = 1$  — дизъюнкция):

$$R(x) = \left[ \sum_{j \in J} [a_j \leq f_j(x) \leq b_j] \geq d \right],$$

Параметры  $J, a_j, b_j, d$  настраиваются по обучающей выборке путём оптимизации критерия информативности.

# Построение закономерностей (rule induction)

4. *Полуплоскость* — линейная пороговая функция:

$$R(x) = \left[ \sum_{j \in J} w_j f_j(x) \geq w_0 \right].$$

5. *Шар* — пороговая функция близости:

$$R(x) = \left[ r(x, x_0) \leq w_0 \right],$$

ABO — алгоритмы вычисления оценок [Ю. И. Журавлёв, 1971]:

$$r(x, x_0) = \max_{j \in J} w_j |f_j(x) - f_j(x_0)|.$$

SCM — машины покрывающих множеств [M. Marchand, 2001]:

$$r(x, x_0) = \sum_{j \in J} w_j |f_j(x) - f_j(x_0)|^\gamma.$$

Параметры  $J, w_j, w_0, x_0$  настраиваются по обучающей выборке путём оптимизации критерия информативности.

# Поиск информативных закономерностей

**Вход:** выборка  $X^l$ ;

**Выход:** множество закономерностей  $Z$ ;

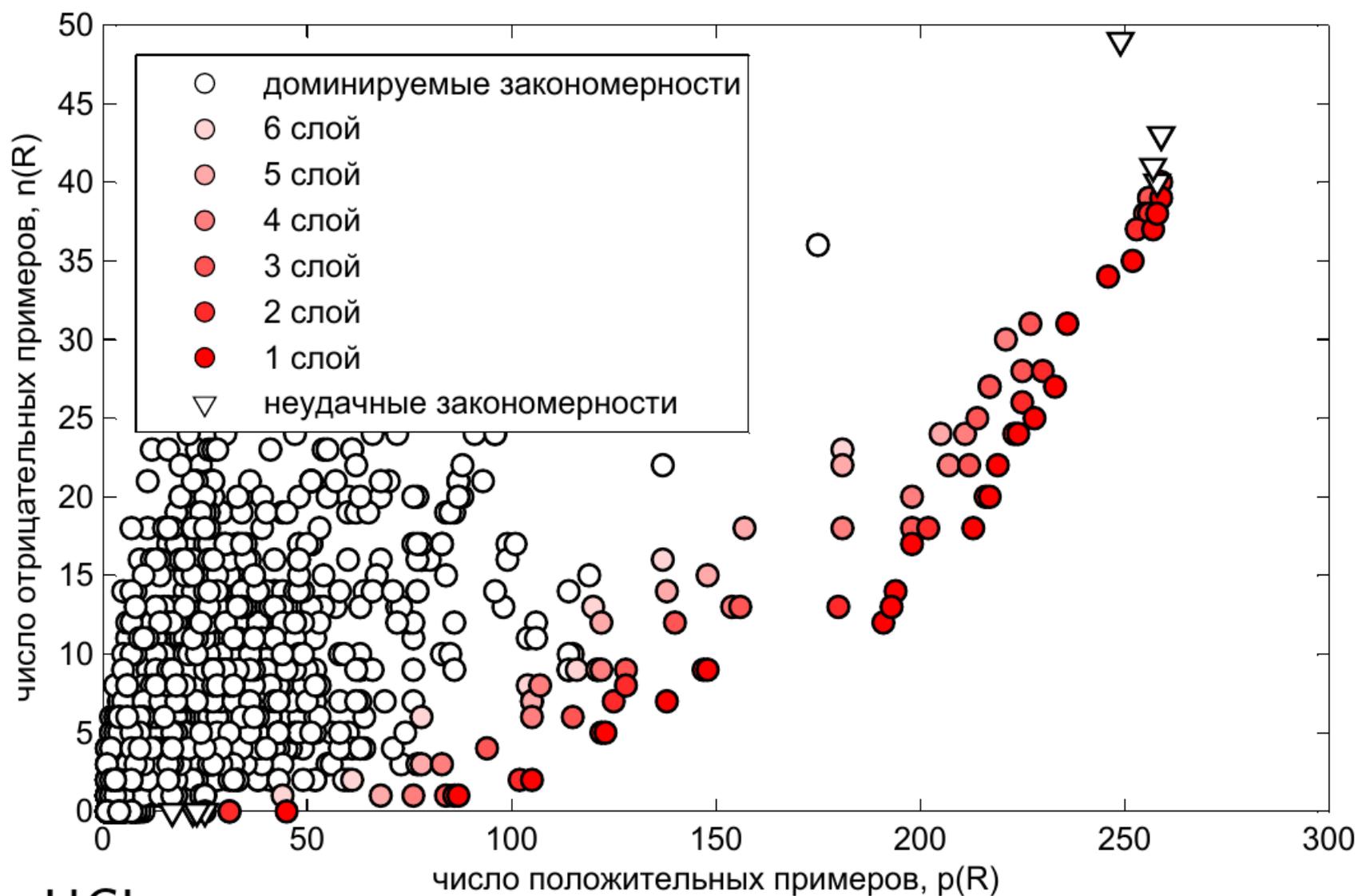
---

- 1: начальное множество правил  $Z$ ;
- 2: **повторять**
- 3:  $Z' :=$  множество модификаций правил  $R \in Z$ ;
- 4: удалить слишком похожие правила из  $Z \cup Z'$ ;
- 5: оценить информативность всех правил  $R \in Z'$ ;
- 6:  $Z :=$  наиболее информативные правила из  $Z \cup Z'$ ;
- 7: **пока** правила продолжают улучшаться
- 8: **вернуть**  $Z$ .

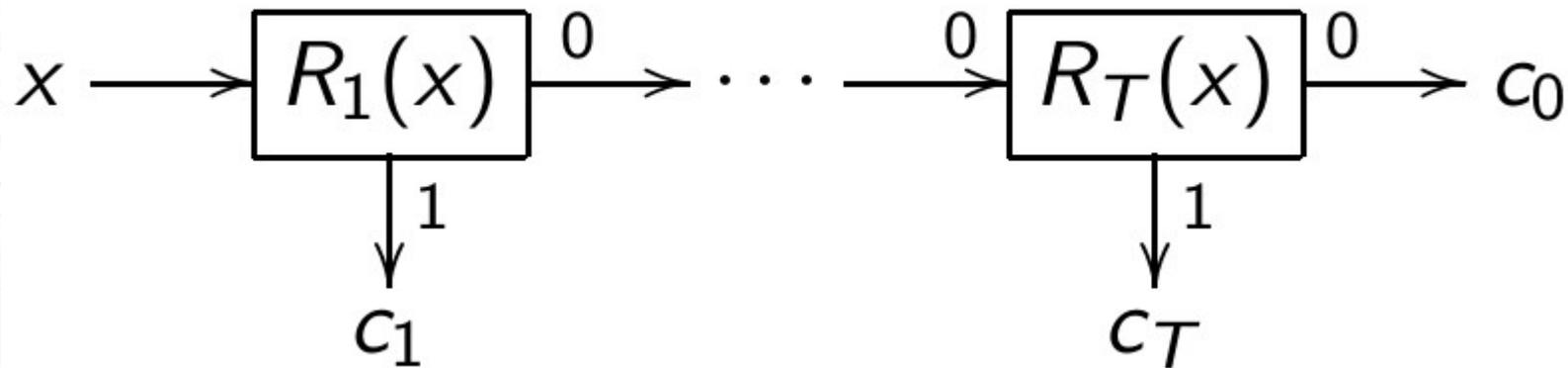
**Частные случаи:**

- стохастический локальный поиск (SLS)
- генетические (эволюционные) алгоритмы
- метод ветвей и границ

# Отбор закономерностей по Парето



# Алгоритмы классификации. Решающий список



- 1: для всех  $t = 1, \dots, T$
- 2: если  $R_t(x) = 1$  то
- 3: вернуть  $c_t$ ;
- 4: вернуть  $c_0$  — отказ от классификации объекта  $x$

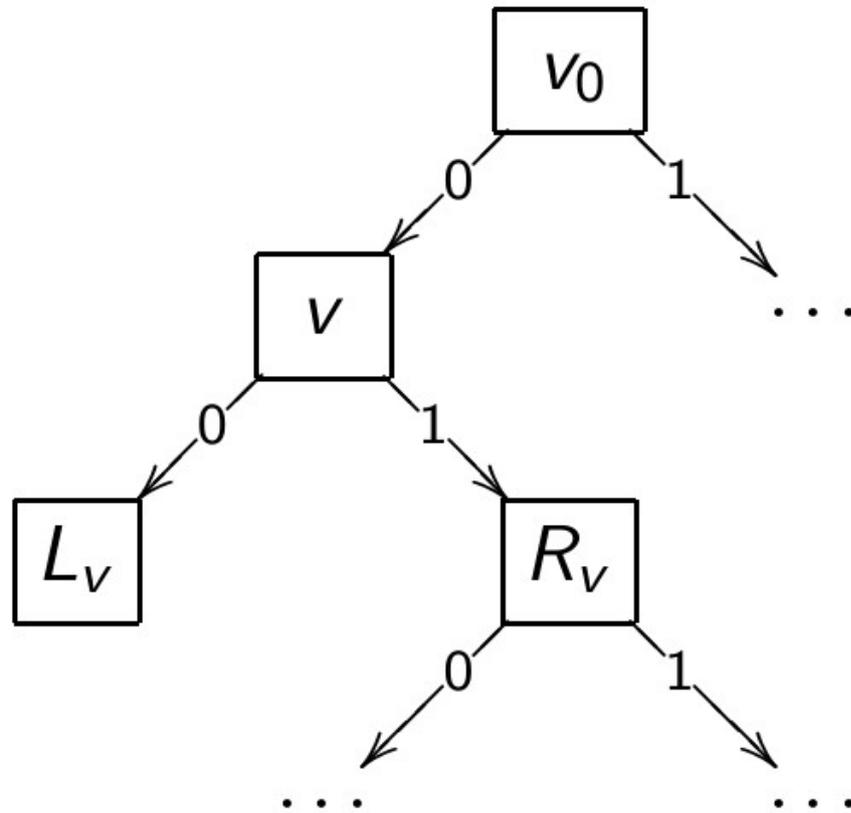
# Жадный алгоритм построения решающего списка

- $U := X_\ell; \quad t := 1$
- Повторять:
  - Выбор класса  $=: c_t$
  - Выбор предиката  $R_t: I(R_t, U) \rightarrow \max$
  - $U := \{ x \in U \mid R_t(x) = 0 \}$
- Пока:  $I > I_{\min}; \quad |U| > \ell_0$

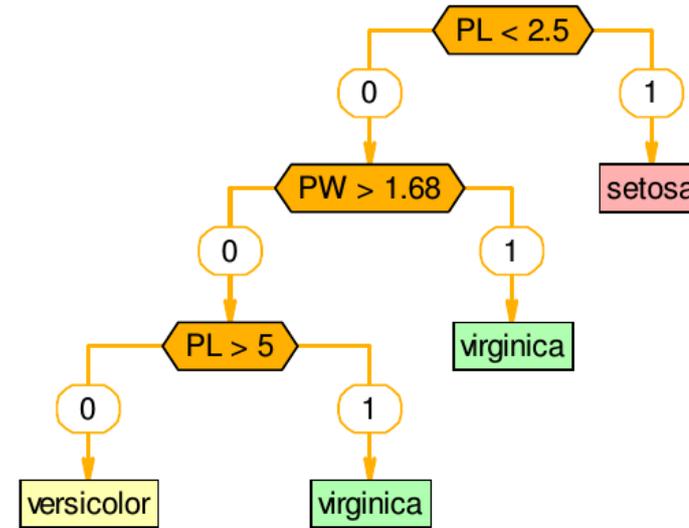
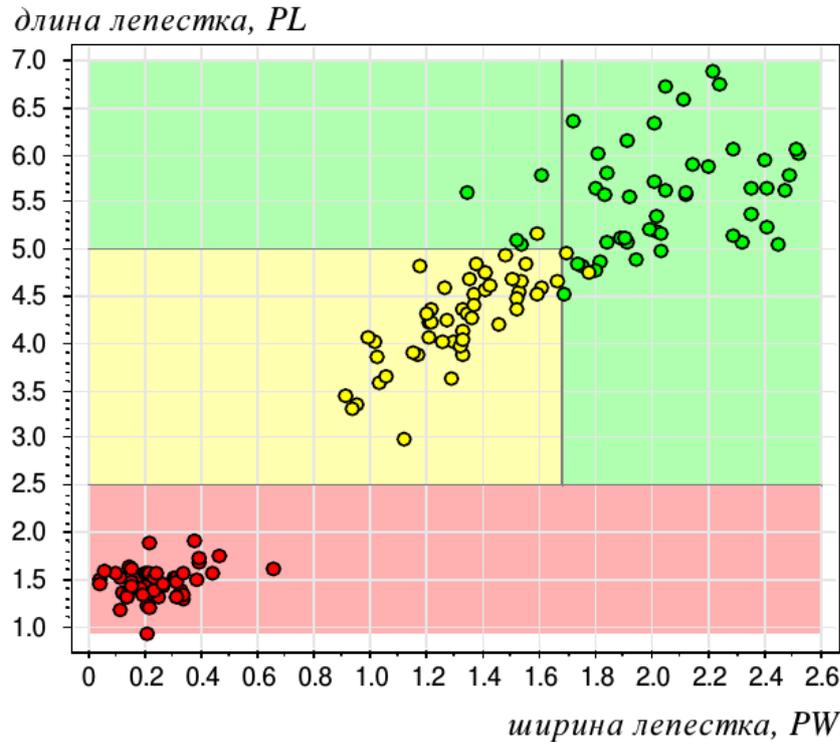
Правильно ли выбирать самый информативный предикат? Ведь у него может оказаться очень много ошибок (т.е. он отнесет много объектов других классов к  $c_t$ , и эти объекты не попадут на другие блоки списка)?

# Решающее дерево

- 1:  $v := v_0$ ;
- 2: **пока**  $v \in V_{\text{внутр}}$
- 3:   **если**  $\beta_v(x) = 1$  **то**
- 4:     переход вправо:  
       $v := R_v$ ;
- 5:   **иначе**
- 6:     переход влево:  
       $v := L_v$ ;
- 7: **вернуть**  $c_v$ .



# Решающее дерево → покрывающий набор конъюнкций



setosa	$r_1(x) = [PL \leq 2.5]$
virginica	$r_2(x) = [PL > 2.5] \wedge [PW > 1.68]$
virginica	$r_3(x) = [PL > 5] \wedge [PW \leq 1.68]$
versicolor	$r_4(x) = [PL > 2.5] \wedge [PL \leq 5] \wedge [PW < 1.68]$

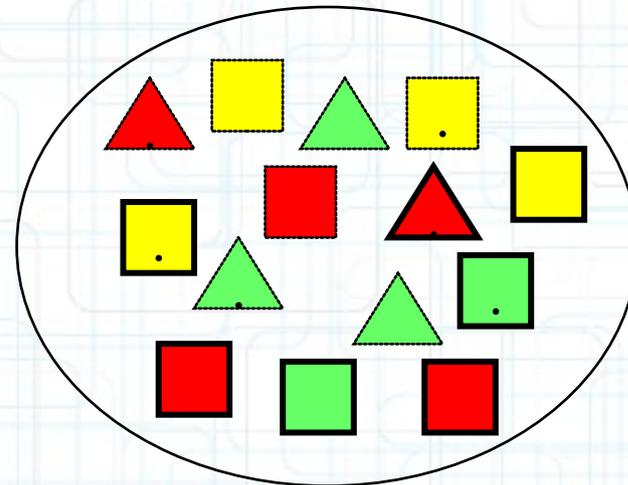
# Жадный алгоритм построения решающего дерева

- Функция:
- Tree buildTree(U) {
  - Выбор предиката  $\beta_v: I(\beta_v, U) \rightarrow \max$
  - $U_0 := \{x \in U \mid \beta_v(x) = 0\}$
  - $U_1 := \{x \in U \mid \beta_v(x) = 1\}$
  - Если  $|U_0| < \ell_0$  или  $|U_1| < \ell_0$  вернуть лист
  - Иначе:
    - $L_v := \text{buildTree}(U_0)$
    - $R_v := \text{buildTree}(U_1)$
- }

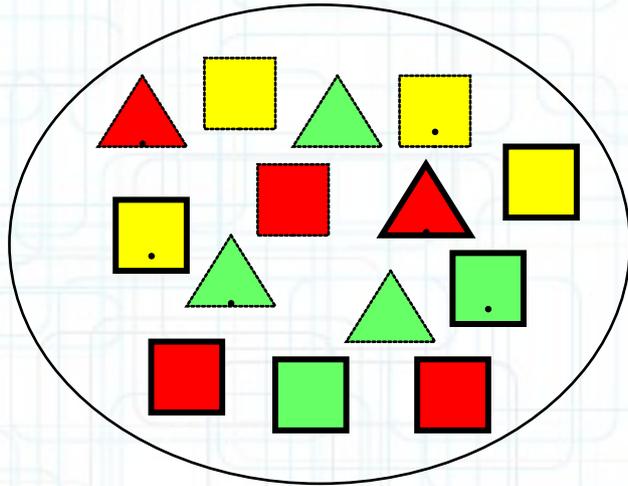
# Пример: треугольники и квадраты

#	Attribute			Shape
	Color	Outline	Dot	
1	green	dashed	no	triange
2	green	dashed	yes	triange
3	yellow	dashed	no	square
4	red	dashed	no	square
5	red	solid	no	square
6	red	solid	yes	triange
7	green	solid	no	square
8	green	dashed	no	triange
9	yellow	solid	yes	square
10	red	solid	no	square
11	green	solid	yes	square
12	yellow	dashed	yes	square
13	yellow	solid	no	square
14	red	dashed	yes	triange

Обучающая выборка



# Энтропия



- 5 треугольников
- 9 квадратов
- Вероятности классов

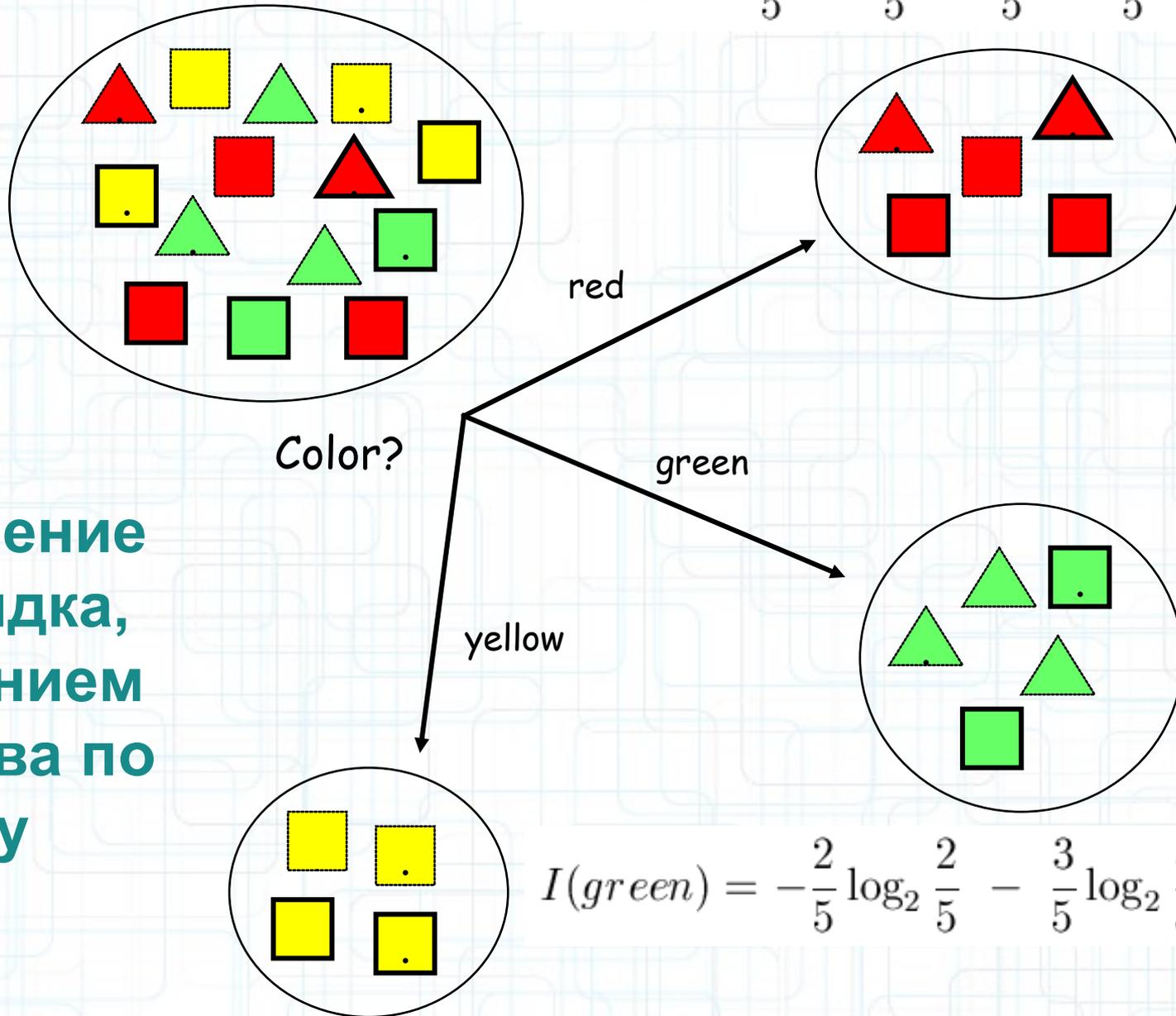
$$p(\square) = \frac{9}{14}$$

$$p(\Delta) = \frac{5}{14}$$

энтропия

$$I = -\frac{9}{14} \log_2 \frac{9}{14} - \frac{5}{14} \log_2 \frac{5}{14} = 0.940 \text{ bits}$$

$$I(\text{red}) = -\frac{3}{5} \log_2 \frac{3}{5} - \frac{2}{5} \log_2 \frac{2}{5} = 0.971 \text{ bits}$$

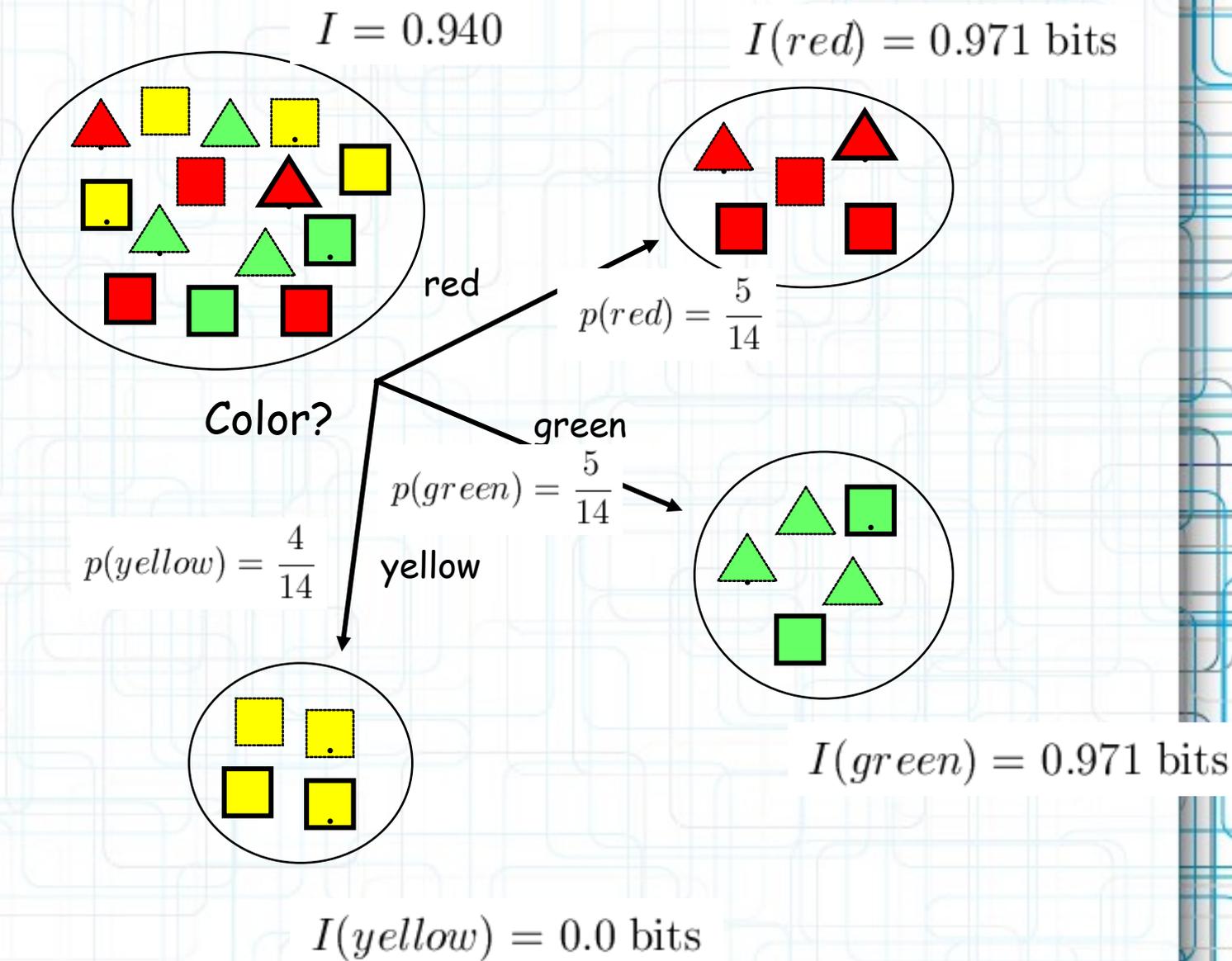


$$I(\text{green}) = -\frac{2}{5} \log_2 \frac{2}{5} - \frac{3}{5} \log_2 \frac{3}{5} = 0.971 \text{ bits}$$

$$I(\text{yellow}) = -\frac{4}{4} \log_2 \frac{4}{4} - \frac{0}{4} \log_2 \frac{0}{4} = 0.0 \text{ bits}$$

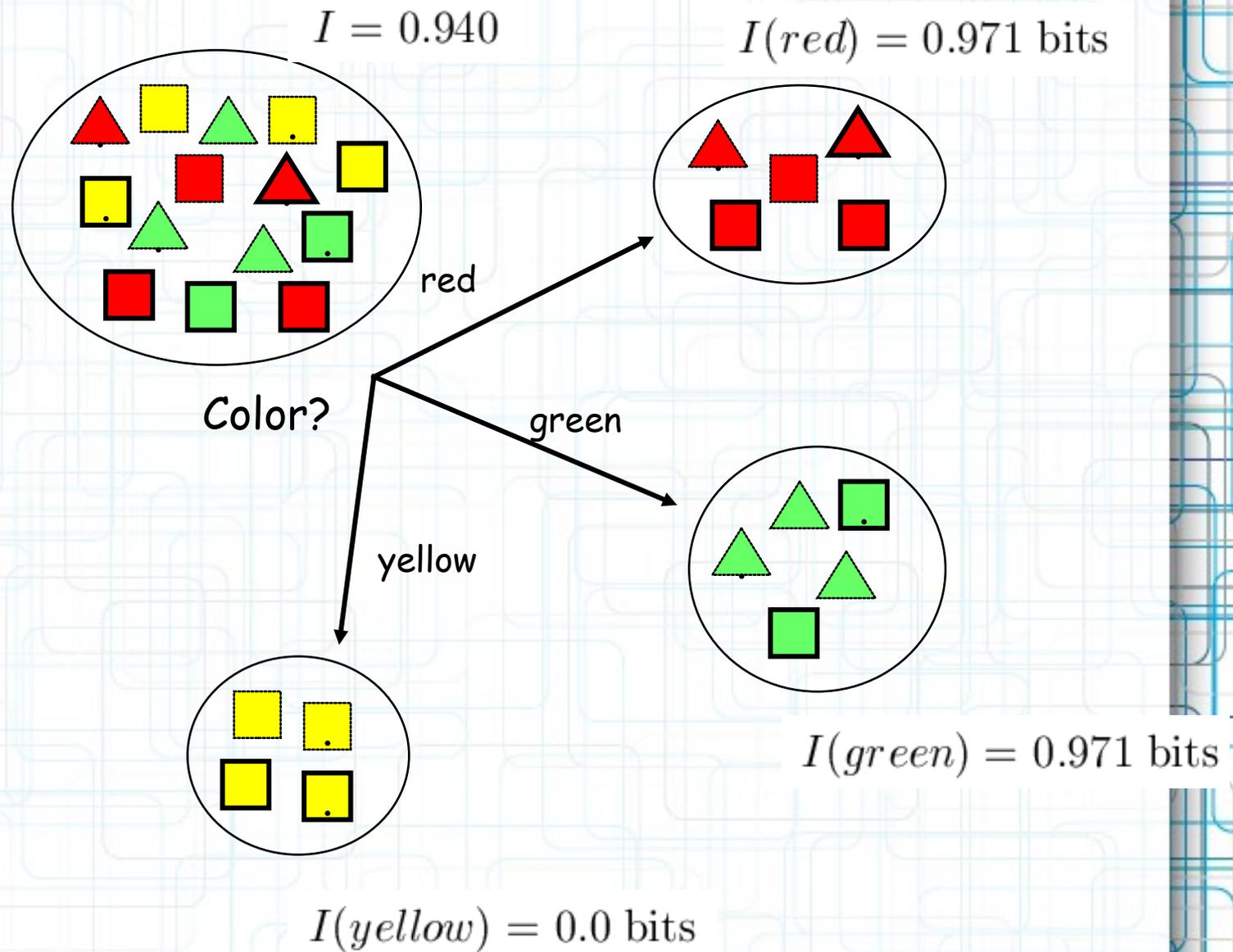
**Уменьшение  
беспорядка,  
разделением  
множества по  
цвету**

# Энтропия после разделения



$$I_{\text{res}}(\text{Color}) = \sum p(v)I(v) = \frac{5}{14}0.971 + \frac{5}{14}0.971 + \frac{4}{14}0.0 = 0.694 \text{ bits}$$

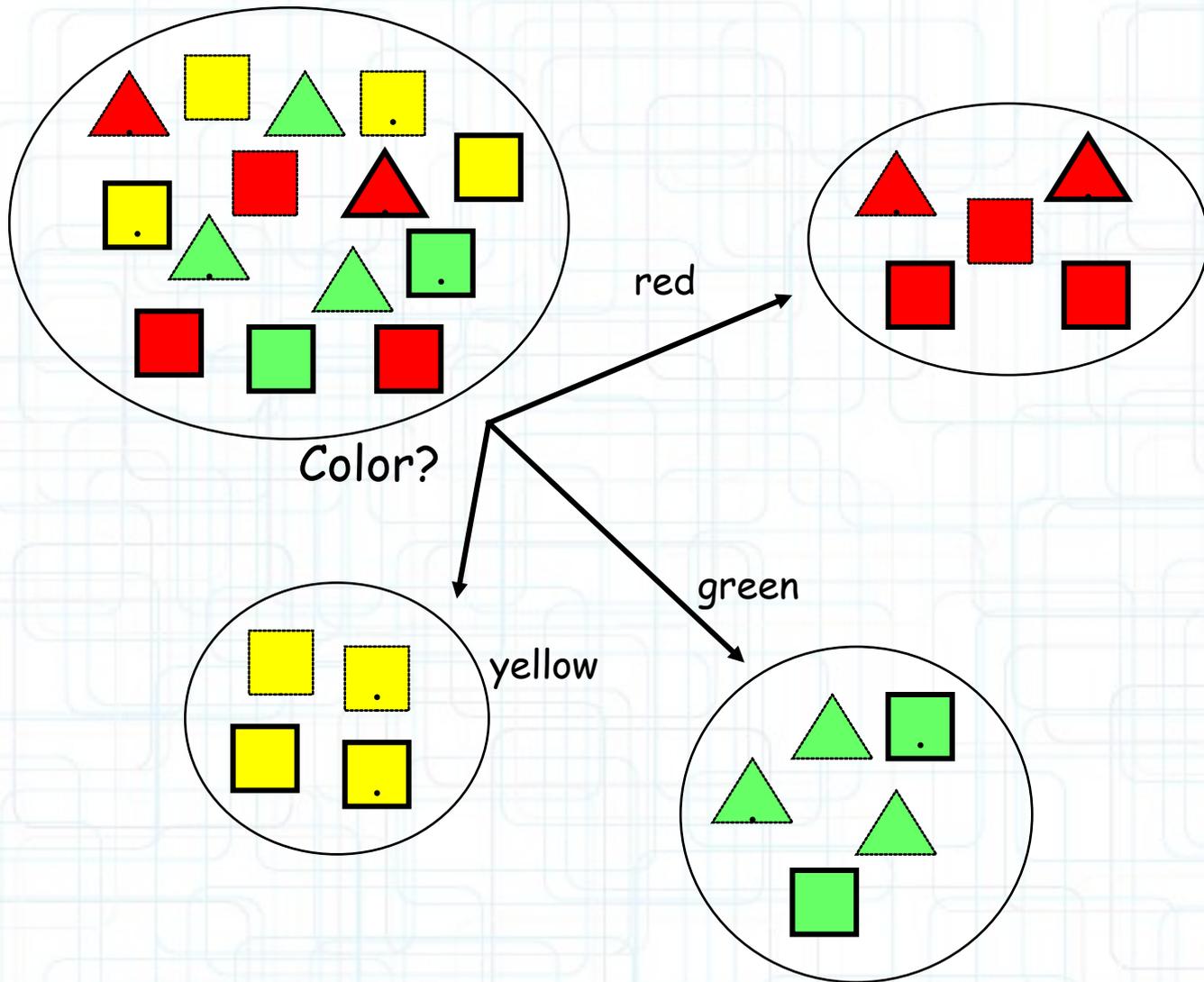
# Прирост информации



$$\text{Gain}(\text{Color}) = I - I_{res}(\text{Color}) = 0.940 - 0.694 = 0.246 \text{ bits}$$

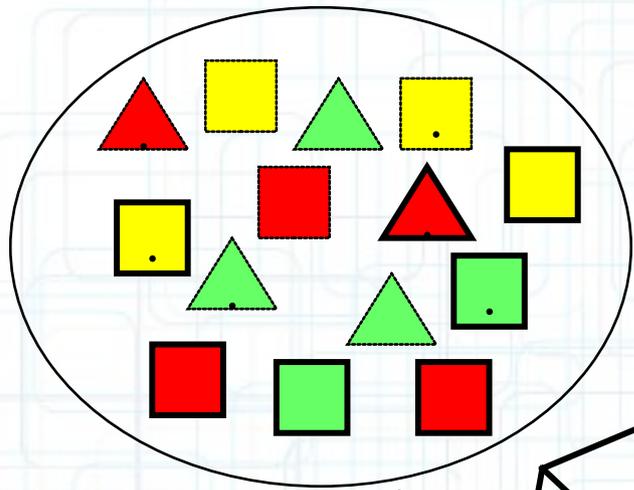
# Прирост информации для каждого признака

- Признаки
  - $\text{Gain}(\text{Color}) = 0.246$
  - $\text{Gain}(\text{Outline}) = 0.151$
  - $\text{Gain}(\text{Dot}) = 0.048$
- Лучше всего разбивать множество по признаку, вносящему наибольший порядок



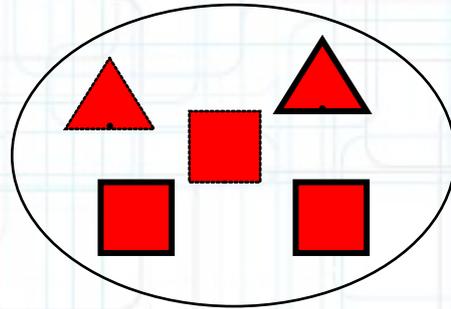
$$\text{Gain(Outline)} = 0.971 - 0 = 0.971 \text{ bits}$$

$$\text{Gain(Dot)} = 0.971 - 0.951 = 0.020 \text{ bits}$$



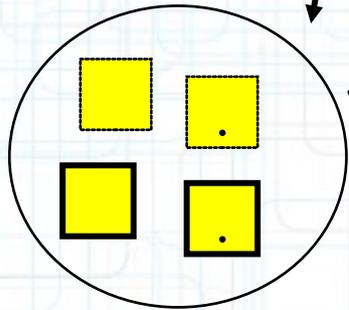
Color?

red

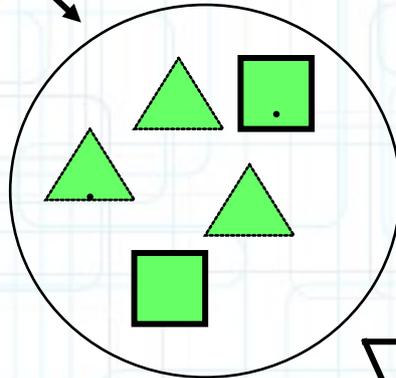


$Gain(Outline) = 0.971 - 0.951 = 0.020$  bits  
 $Gain(Dot) = 0.971 - 0 = 0.971$  bits

green

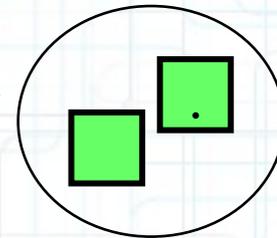


yellow

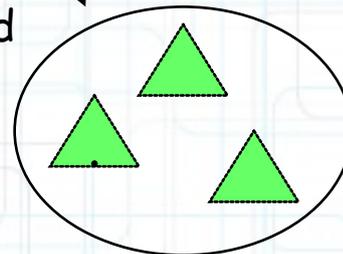


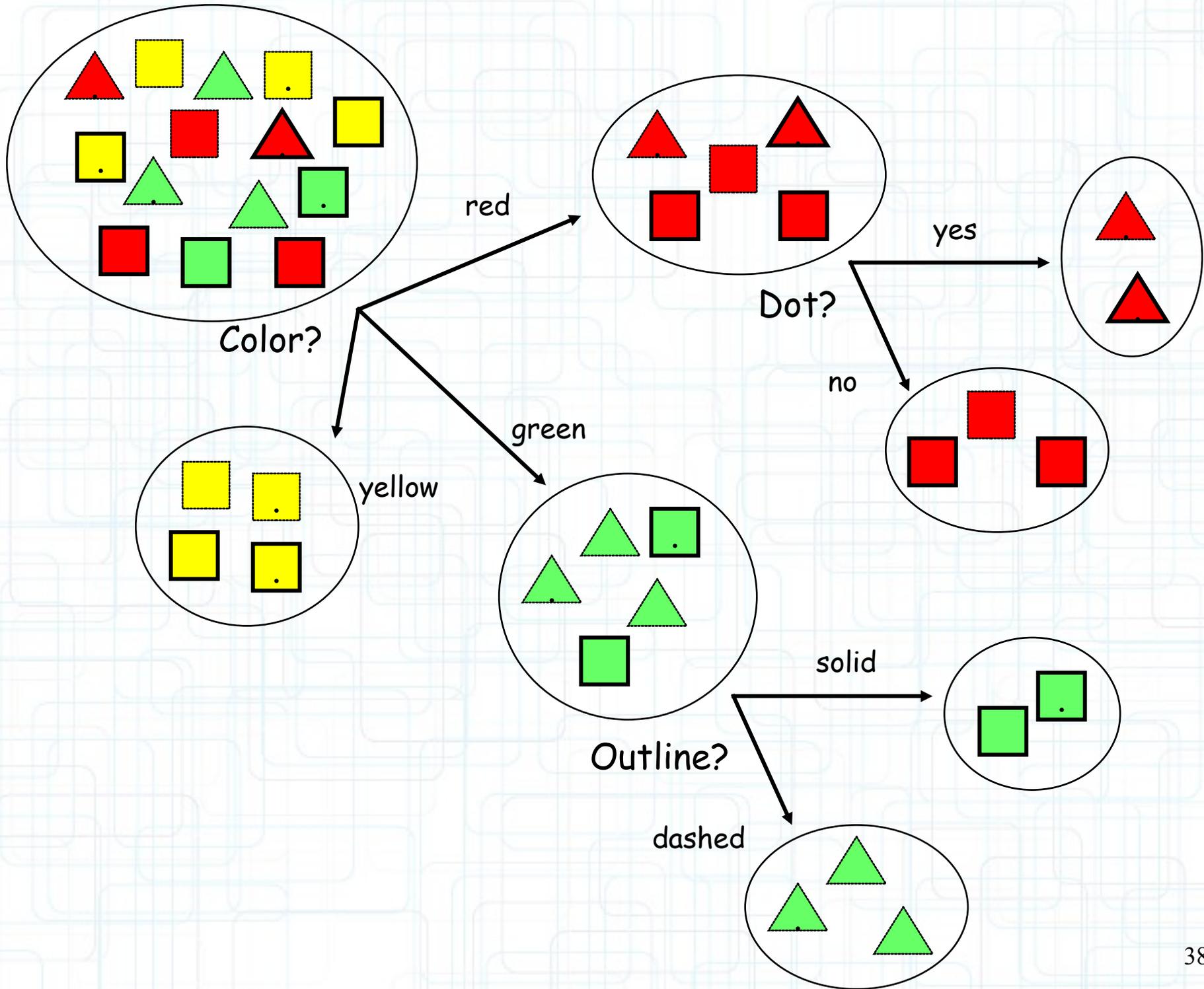
Outline?

solid

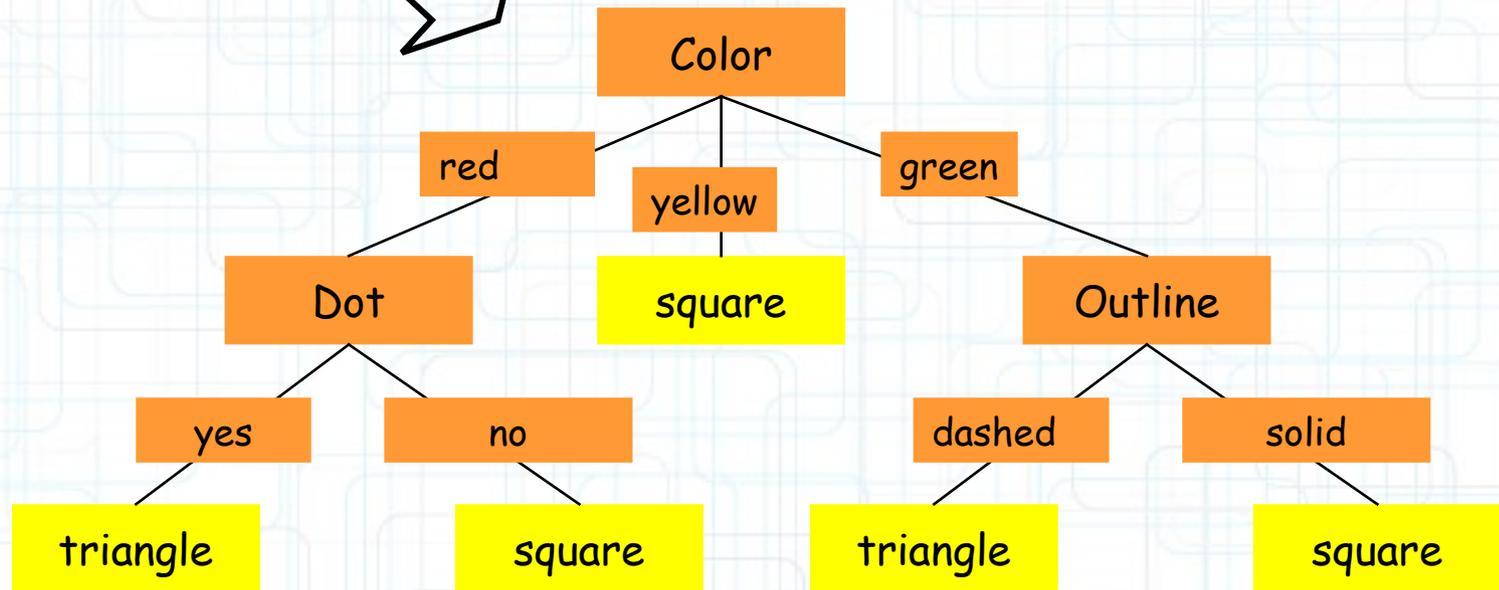
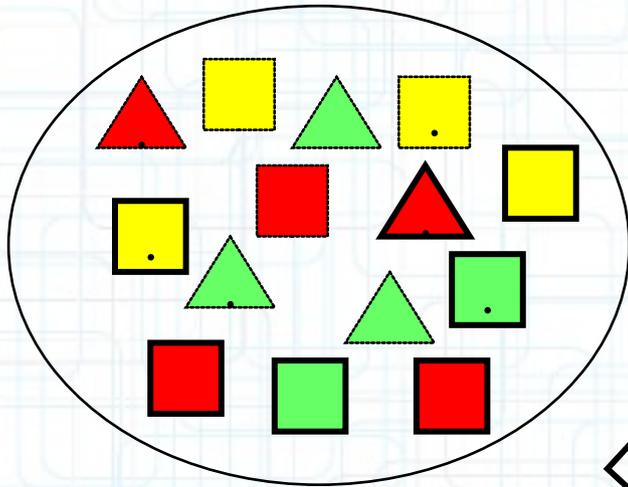


dashed





# Итоговое дерево



# Редукция дерева (pruning)

- Pre-pruning – критерий раннего останова. Досрочное прекращение ветвления, если информативность  $<$  порога или глубина велика.
- Post-pruning – пост-редукция. Просматриваем все внутренние вершины дерева и проверяем их качество на тестовой выборке (OutOfBag error). Заменяем листом, где качество после разделения ухудшается

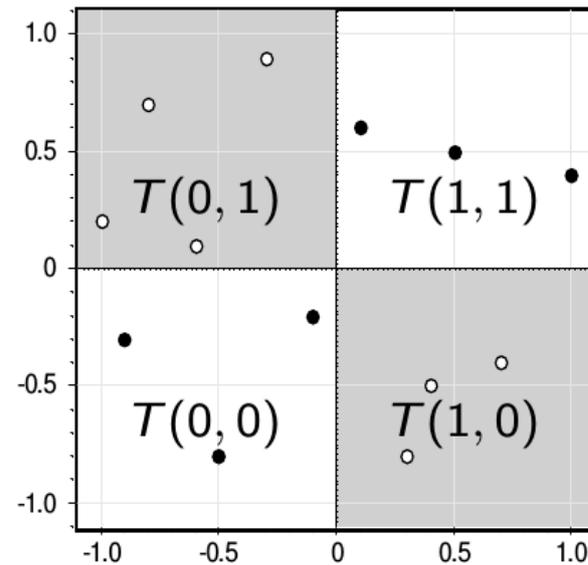
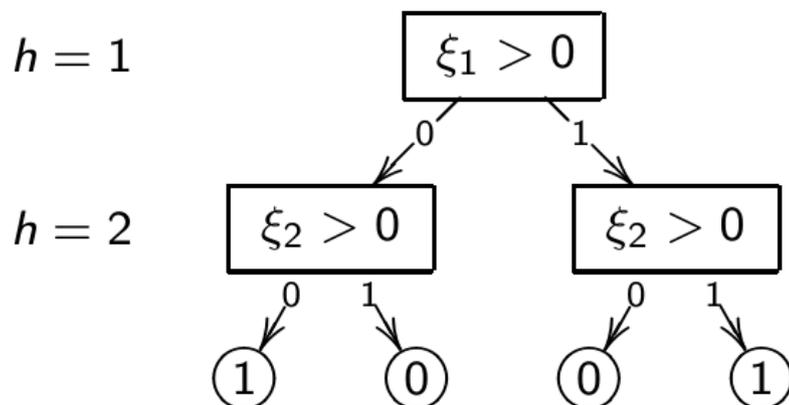
# Обобщение на случай задачи регрессии

- В каждом листе целевое значение определяется по методу наименьших квадратов
- Критерий информативности – среднеквадратическая ошибка

# Небрежные решающие деревья (Oblivious Decision Tree)

- Для всех узлов на глубине  $h$  условие ветвления одинаково
- Дерево получается сбалансированным, на глубине  $h$  ровно  $2^{h-1}$  вершин

Пример: задача XOR,  $H = 2$ .



# Вопросы для самоконтроля

- Сгенерируйте случайную выборку из 10 объектов с классами +1 и -1 и одним количественным признаком. Придумайте предикат (например, пороговый  $x < x_0$ ) и вычислите для него параметры  $p, n$
- Примените точный тест Фишера к придуманному выше предикату
- Вычислите прирост информации  $IGain$  для вашего предиката
- Найдите для вашей выборки пороговые предикаты, формирующие лучший решающий список. Всегда ли первый из этих предикатов будет являться самым информативным?

# Вопросы для самоконтроля

- Сгенерируйте случайную выборку из 10 объектов с классами +1 и -1 и одним количественным признаком. Постройте лучший список с двумя пороговыми предикатами.
- Для придуманной выборки постройте лучшее дерево глубины 2 (обычное/небрежное).
- Оцените качество классификации построенного списка и деревьев