

# Что такое **MACHINE LEARNING?**

Н.Ю. Золотых

[zolotykh@vmk.unn.ru](mailto:zolotykh@vmk.unn.ru)

ИТЛаб, ВМК, ННГУ

26 апреля 2012

# План

- Что такое машинное обучение?
- Обучение с учителем
  - Машина опорных векторов (SVM)
  - Деревья решений
  - Бустинг
    - \* AdaBoost
    - \* Gradient Boosting Trees
  - Баггинг
    - \* Баггинг
    - \* Random Forests
- Обучение без учителя: кластеризация

# 1. Что такое машинное обучение (*machine learning*)?

*Машинное обучение* — процесс, в результате которого машина (компьютер) способна показывать поведение, которое в нее не было явно заложено (запрограммировано).

*A.L. Samuel Some Studies in Machine Learning Using the Game of Checkers*  
// IBM Journal. July 1959. P. 210–229.

Говорят, что компьютерная программа *обучается* на основе опыта  $E$  по отношению к некоторому классу задач  $T$  и мере качества  $P$ , если качество решения задач из  $T$ , измеренное на основе  $P$ , улучшается с приобретением опыта  $E$ .

*T.M. Mitchell Machine Learning. McGraw-Hill, 1997.*

- На практике фаза обучения может предшествовать фазе работы алгоритма (например, детектирование лиц на фотокамере)
- или обучение (и дополнительное обучение) может проходить в процессе функционирования алгоритма (например, определение спама).

## **1.1. Сфера приложения**

- Компьютерное зрение
- Распознавание речи
- Компьютерная лингвистика и обработка естественных языков
- Медицинская диагностика
- Биоинформатика
- Техническая диагностика
- Финансовые приложения
- Поиск и рубрикация текстов
- Интеллектуальные игры
- Экспертные системы
- ...

## **1.2. Смежные области**

Pattern Recognition (распознавание образов)

Data Mining (интеллектуальный анализ данных)

Artificial Intelligence (искусственный интеллект)

## **1.3. Что нужно знать?**

Линейная алгебра

Теория вероятностей и математическая статистика

Методы оптимизации

Численные методы

Математический анализ

Дискретная математика

и др.

## 1.4. Ресурсы

- Wiki-портал <http://www.machinelearning.ru>
- Мой курс: <http://www.uic.unn.ru/~zny/ml>  
(презентации лекций, лабораторные работы, описание системы R, ссылки, ML для «чайников» и др.)
- *Воронцов К.В.* Машинное обучение (курс лекций)  
см. <http://www.machinelearning.ru>
- *Ng A.* Machine Learning Course (video, lecture notes, presentations, labs)  
<http://ml-class.org>
- *Hastie T., Tibshirani R., Friedman J.* The elements of statistical learning: Data Mining, Inference, and Prediction. 2nd Edition. Springer, 2009  
<http://www-stat.stanford.edu/~tibs/ElemStatLearn/>

## 1.5. Software

- Библиотека машинного зрения OpenCV (C, C++, интерфейс для Python) (раздел ML)
- Система для статистических вычислений R
- Библиотека алгоритмов для анализа данных Weka (Java)
- Пакет для решения задач машинного обучения и анализа данных Orange
- Система для решения задач машинного обучения и анализа данных RapidMiner
- ...

## 1.6. Типы задач машинного обучения

- Обучение с учителем (классификация, регрессия и др.)
- Обучение без учителя (кластеризация, визуализация данных, понижение размерности и др.)
- Обучение с подкреплением (reinforcement learning)
- Активное обучение
- ...

## 2. Обучение с учителем

Множество  $\mathcal{X}$  – *объекты, примеры, ситуации, входы* (samples)

Множество  $\mathcal{Y}$  – *ответы, отклики, «метки», выходы* (responses)

Имеется некоторая зависимость (детерминированная или вероятностная), позволяющая по  $x \in \mathcal{X}$  предсказать  $y \in \mathcal{Y}$ .

т. е. если зависимость детерминированная, существует функция  $f^* : \mathcal{X} \rightarrow \mathcal{Y}$ .

Зависимость известна только на объектах из *обучающей выборки*:

$$\{(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)\}$$

Пара  $(x_i, y_i) \in \mathcal{X} \times \mathcal{Y}$  – *прецедент*.

Задача *обучения с учителем*: восстановить (аппроксимировать) зависимость, т. е. построить функцию (*решающее правило*)  $f : \mathcal{X} \rightarrow \mathcal{Y}$ , по новым объектам  $x \in X$  предсказывающую  $y \in Y$ :

$$y = f(x) \approx f^*(x).$$

## Признаковые описания

Вход:

$$x = (\xi_1, \xi_2, \dots, \xi_d) \in \mathcal{X} = Q_1 \times Q_2 \times \dots \times Q_d,$$

где  $Q_j = \mathbf{R}$  или  $Q_j$  — конечно

$\xi_j$  —  $j$ -й признак (*свойство, атрибут*) объекта  $x$ .

- Если  $Q_j$  конечно, то  $j$ -й признак — *номинальный* (*категориальный* или *фактор*).  
Если  $|Q_j| = 2$ , то признак *бинарный* и можно считать, например,  $Q_j = \{0, 1\}$ .
- Если  $Q_j$  конечно и упорядочено, то признак *порядковый*.  
Например,  $Q = \{\text{холодно, прохладно, тепло, жарко}\}$
- Если  $Q_j = \mathbf{R}$ , то признак *количественный*.

Выход:  $y \in \mathcal{Y}$

- $\mathcal{Y} = \mathbf{R}$  — задача восстановления регрессии
- $\mathcal{Y} = \{1, 2, \dots, K\}$  — задача классификации.

## Пример 1. Медицинская диагностика

Имеются данные о 114 лицах с заболеванием щитовидной железы.

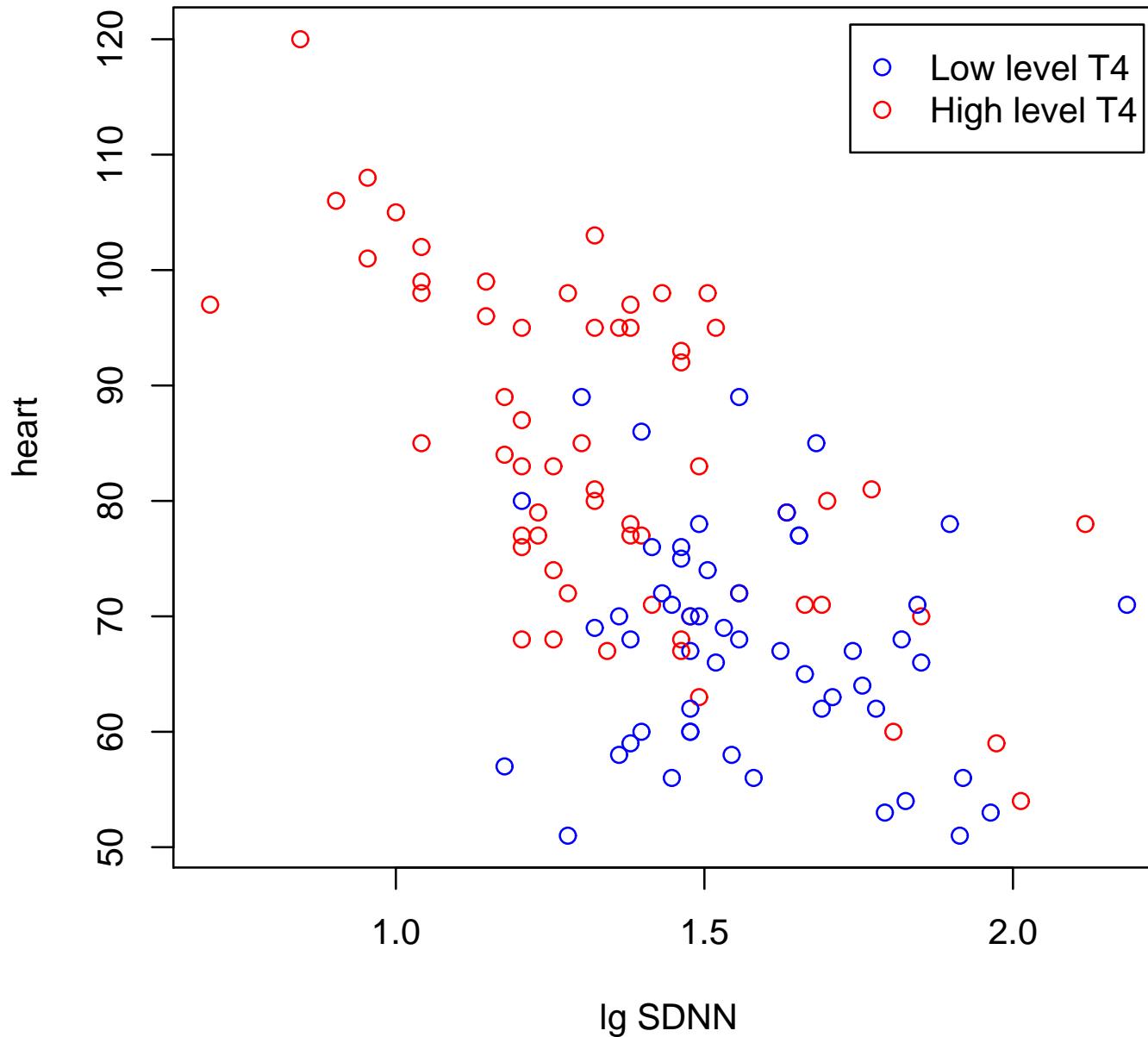
У 61 — повышенный уровень свободного гормона T4,

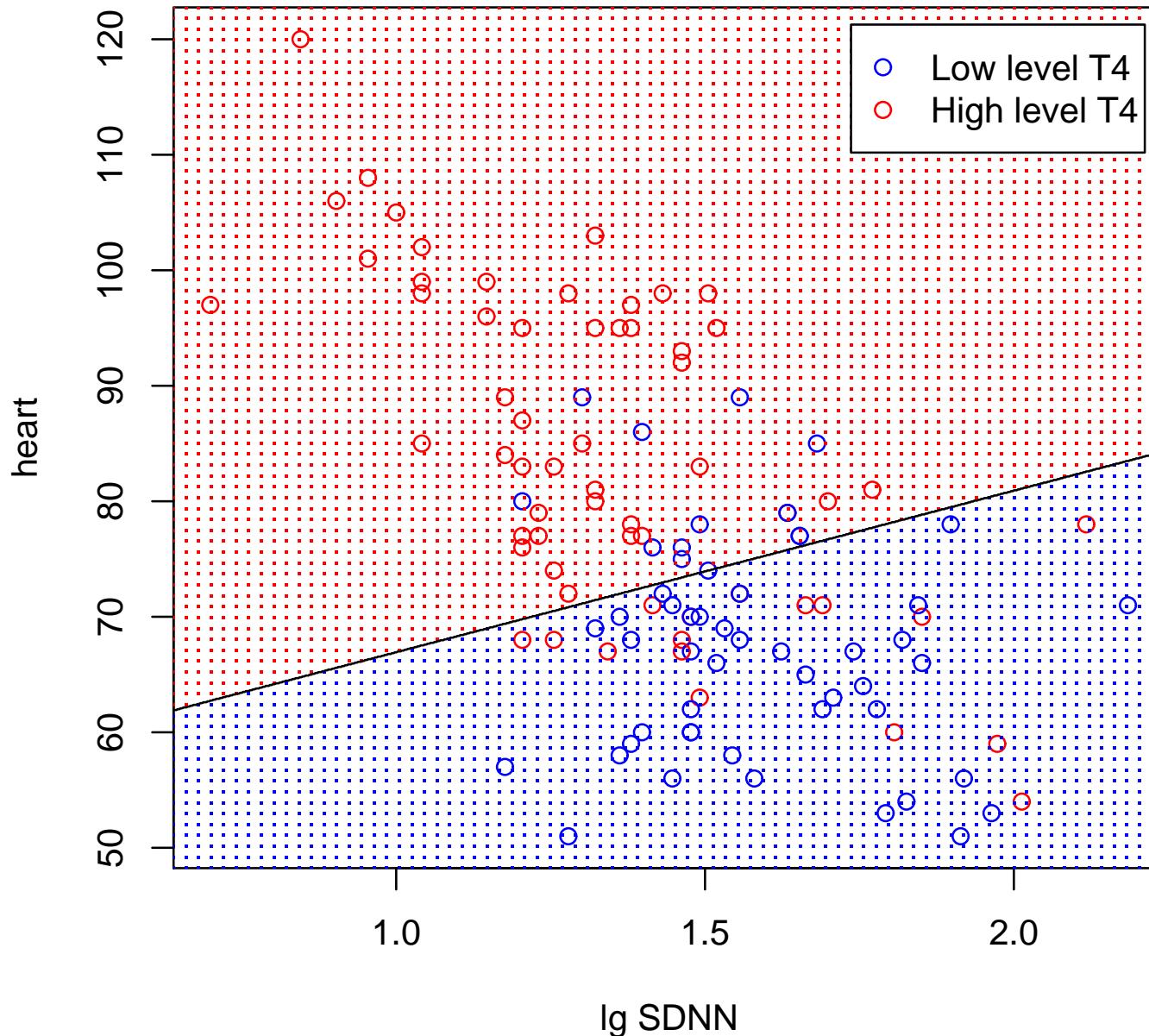
у 53 — уровень гормона в норме.

Для каждого пациента известны следующие показатели:

- heart — частота сердечных сокращений (пульс),
- SDNN — стандартное отклонение длительности интервалов между синусовыми сокращениями RR.

Можно ли научиться предсказывать (допуская небольшие ошибки) уровень свободного T4 по heart и SDNN?





$$16 \cdot \lg \text{SDNN} - \text{heart} + 50 = 0$$

Ошибка на обучающей выборке — 23 %.

Можно ли было ошибку на обучающей выборке сделать меньше?

Конечно!

Но это вовсе не означает, что мы будем лучше классифицировать *новые* объекты.

Итак, *малая ошибка на данных, по которым построено решающее правило, не гарантирует, что ошибка на новых объектах также будет малой.*

*Обобщающая способность (качество) решающего правила* — это способность решающего правила правильно предсказывать выход для новых объектов, не вошедших в обучающую выборку.

*Переобучение* — решающее правило хорошо решает задачу на обучающей выборке, но имеет плохую обобщающую способность.

## Пример 2. Распознавание рукописных символов (цифр)

Научиться распознавать рукописный символ по его изображению.

Изображение — битовая матрица размера  $32 \times 32$ :

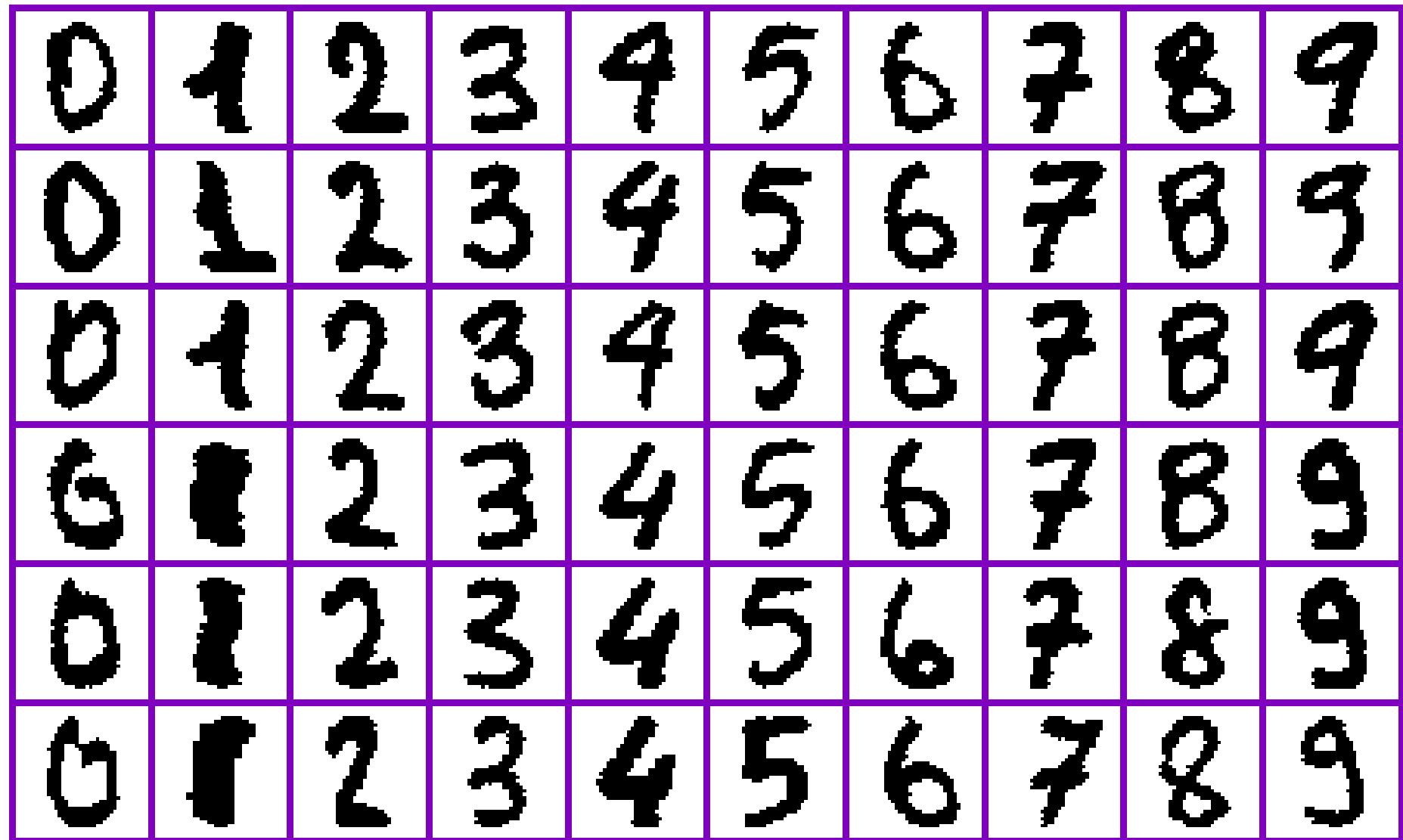
$$x \in \mathcal{X} = \{0, 1\}^{32 \times 32} = \{0, 1\}^{1024}$$

$$\mathcal{Y} = \{0, 1, 2, \dots, 9\}$$

Это задача классификации.

optdigit <http://www.ics.uci.edu/~mlearn/MLRepository.html> – 1934 прецедента.

Некоторые объекты из обучающей выборки



## *Проблема построения признакового описания.*

В задаче распознавания символов можно использовать признаковое описание на основе анализа контура изображения.

В примере letter-recognition

<http://www.ics.uci.edu/~mlearn/MLRepository.html> распознавания печатных заглавных букв (26 классов) для кодирования изображений используется другой подход.

Входы (входы отмасштабированы и округлены, так, чтобы они принимали целые значения от 0 до 15; база содержит 20000 прецедентов):

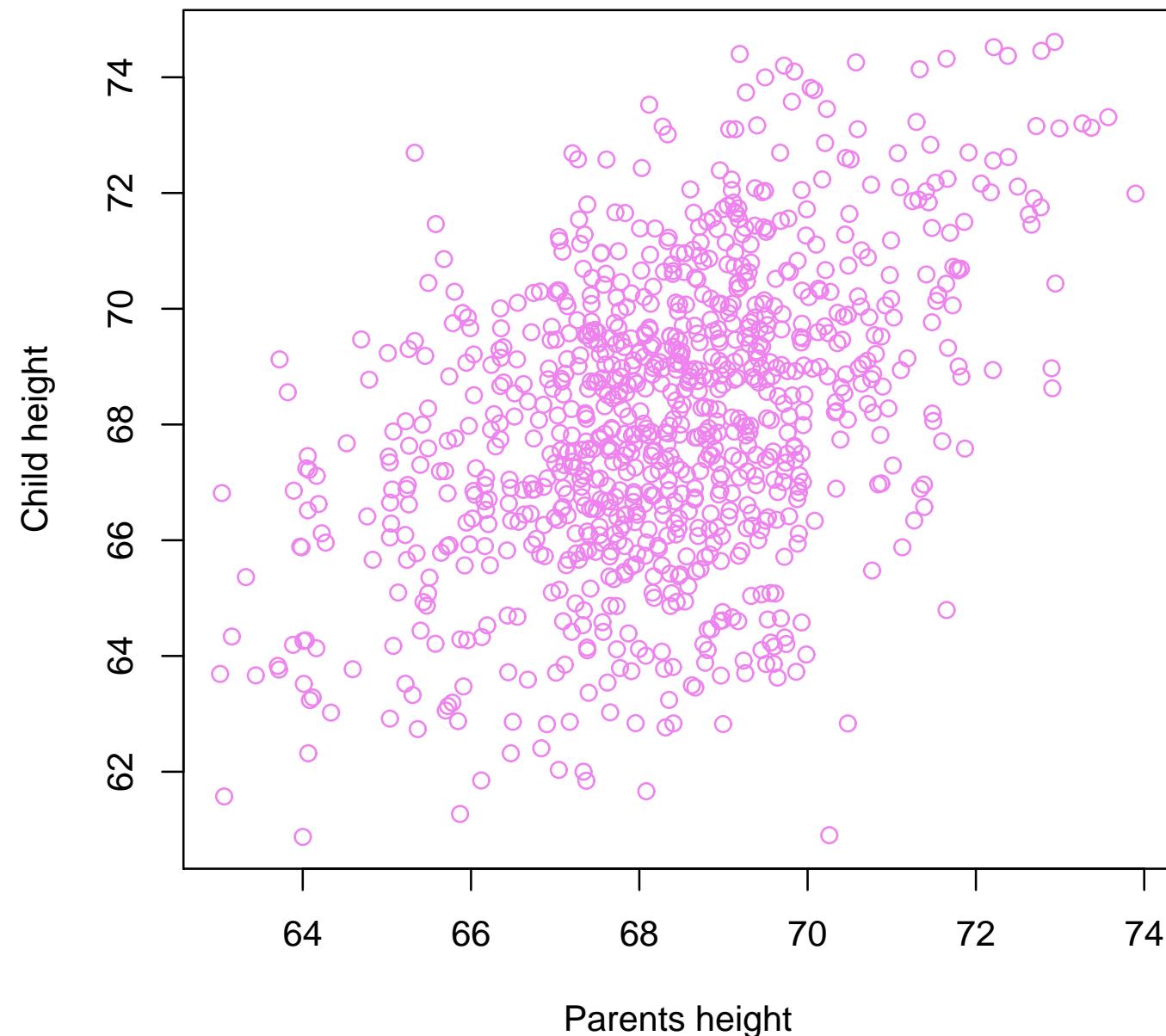
1. x-box — координата  $x$  левого нижнего угла обрамляющего прямоугольника,
2. y-box — координата  $y$  левого нижнего угла обрамляющего прямоугольника,
3. width — ширина прямоугольника,
4. high — высота прямоугольника,
5. onpix — общее число подсвеченных пикселей
6. x-bar — среднее значение координаты  $x$  для подсвеченных пикселей
7. y-bar — среднее значение координаты  $y$  для подсвеченных пикселей
8. x2bar — стандартное отклонение для координаты  $x$  подсвеченных пикселей
9. y2bar — стандартное отклонение для координаты  $y$  подсвеченных пикселей
10. xybar — коэффициент корреляции  $x$  и  $y$  подсвеченных пикселей
11. x2ybr — среднее значение  $x^2y$
12. xy2br — среднее значение  $xy^2$
13. x-ege — среднее значение числа отрезков при просмотре слева направо
14. hegvy — коэффициент корреляции между средним значением числа отрезков при просмотре слева направо и  $y$
15. y-ege — среднее значение числа отрезков при просмотре снизу вверх
16. yegvx — коэффициент корреляции между средним значением числа отрезков при просмотре снизу вверх и  $x$

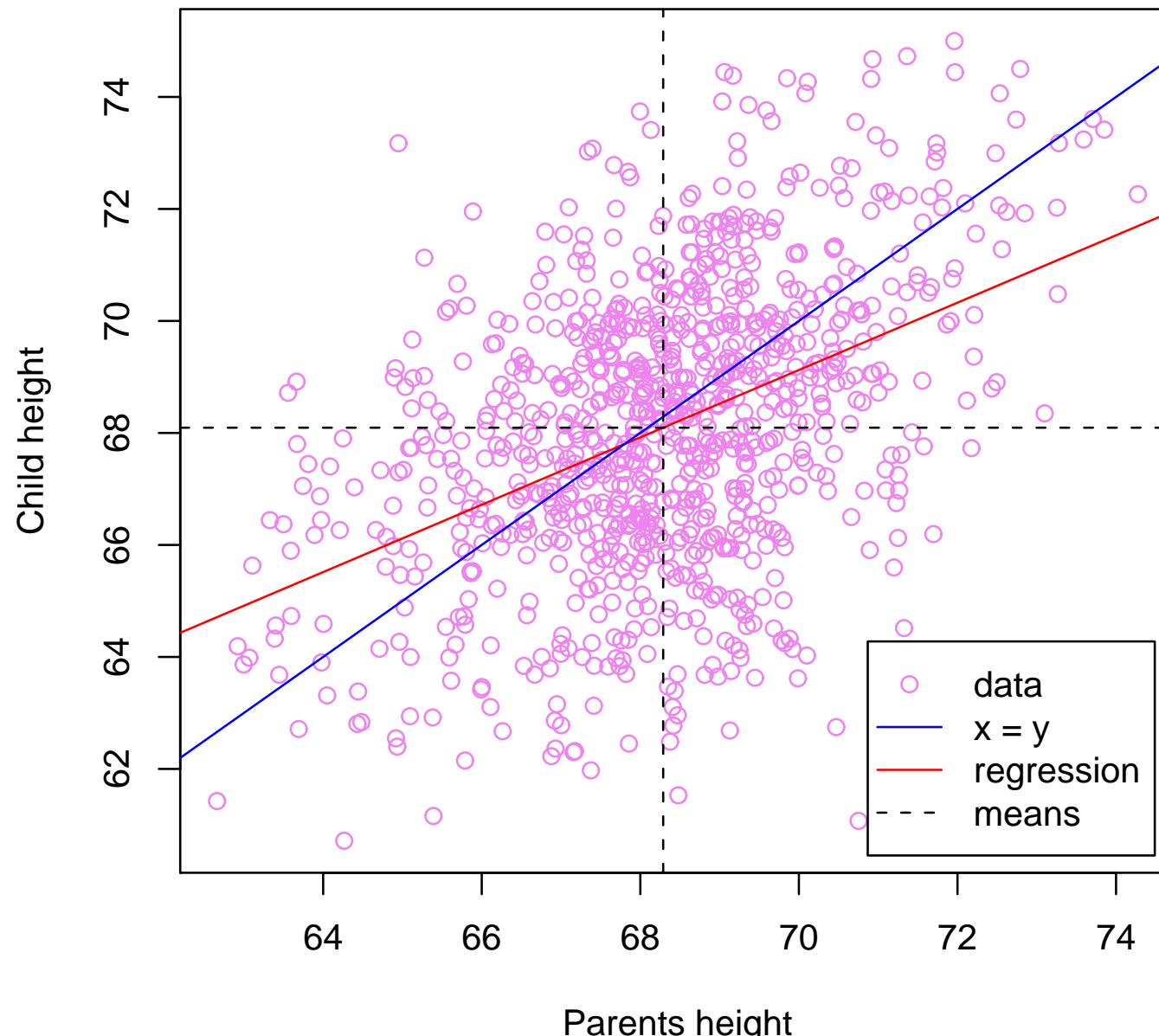
## **Пример 3. «Регрессия к середине» Ф. Гальтона**

Фрэнсис Гальтона (1822–1911)

«Регрессия к середине в наследовании роста» (1885)

# Зависимость роста взрослого ребенка от роста родителей в исследовании Ф. Гальтона

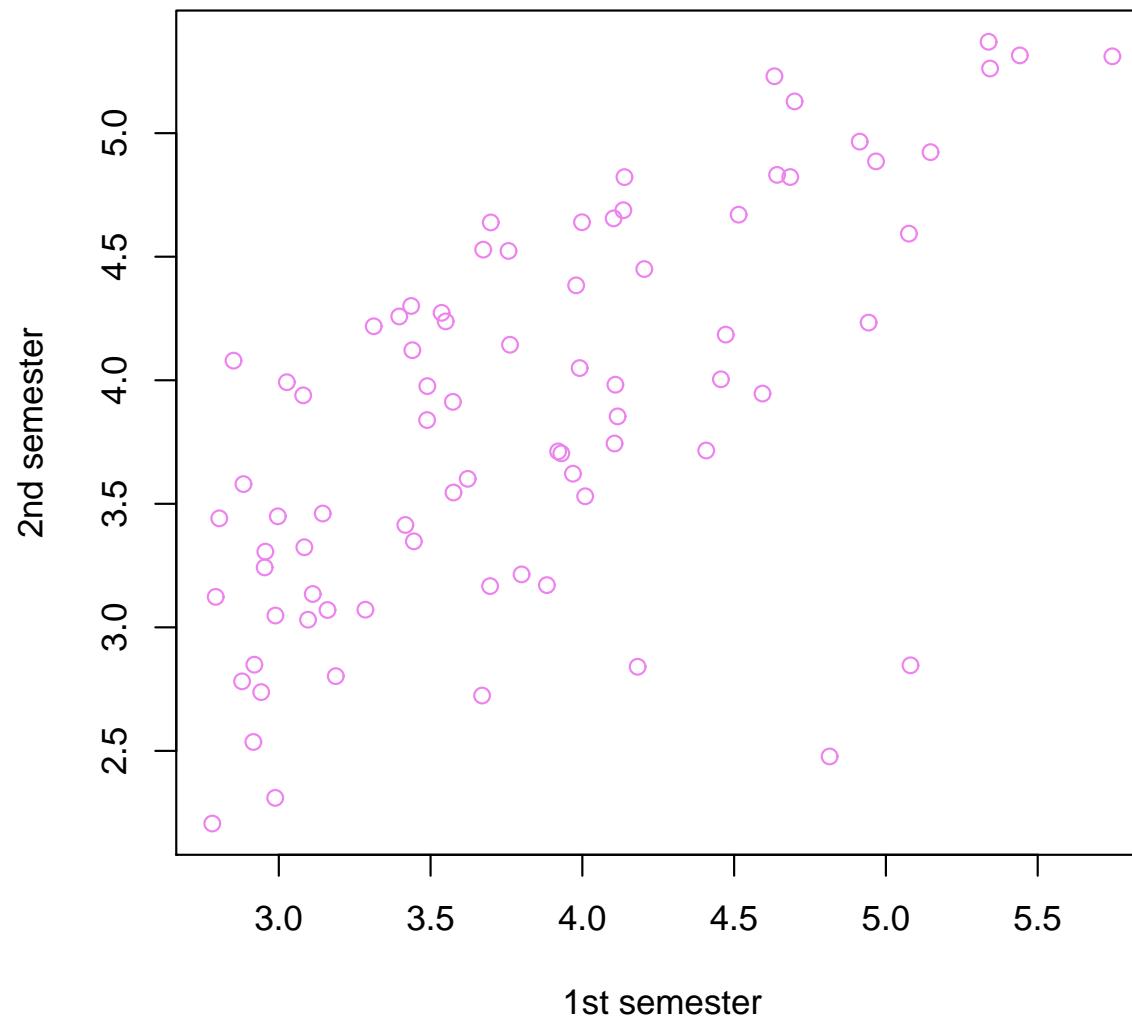




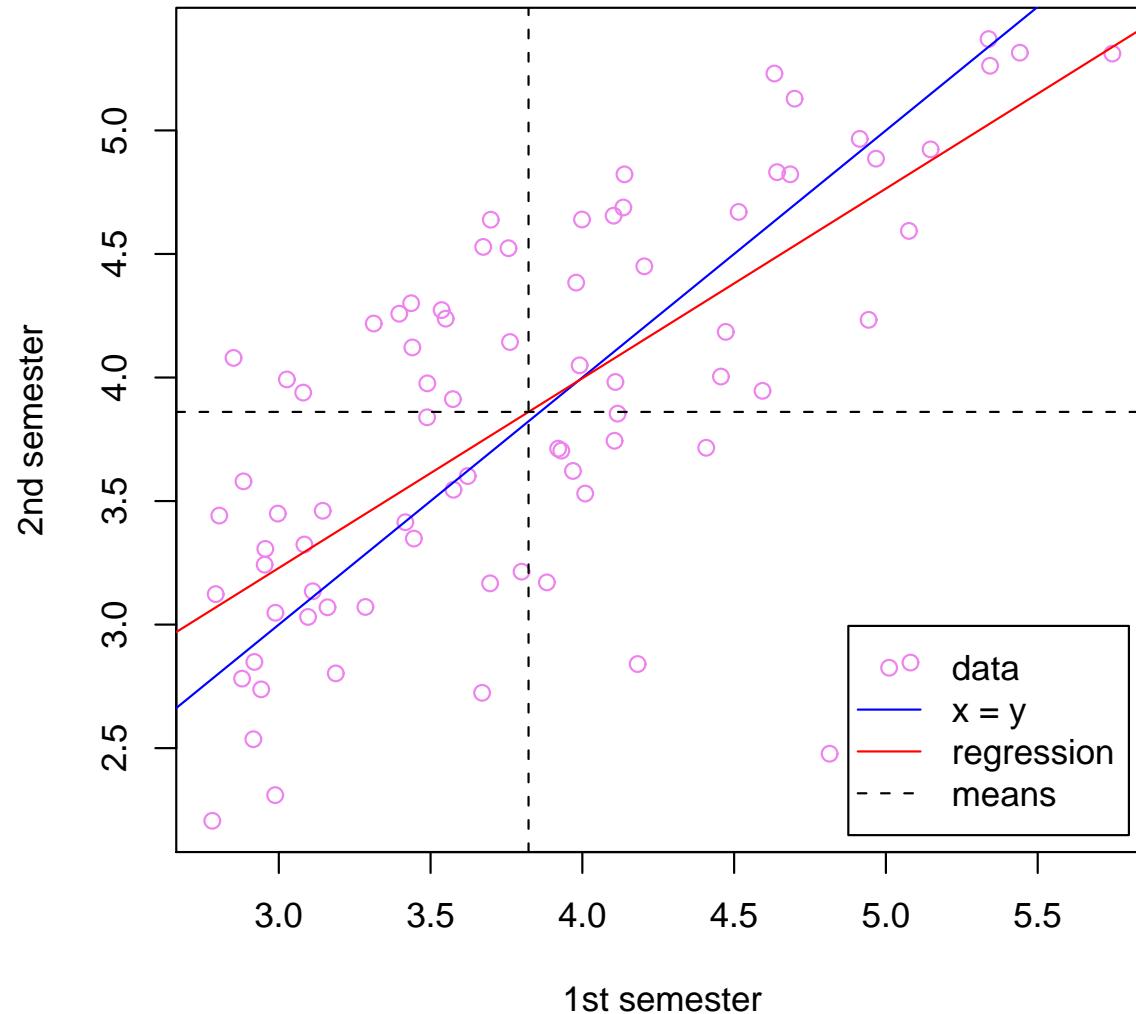
928 наблюдений  $y = 0.65x + 24 = 68.2 + 0.65 \times (x - 68.2)$

$x$  = средняя оценка по мат. анализу и алгебре в 1-м семестре

$y$  = средняя оценка по мат. анализу, алгебре и программированию во 2-м семестре



79 студентов



$$y = 0.93 + 0.77 \times x \approx 3.86 + 0.77 \times (x - 3.82)$$

3.82 — средняя оценка по всем студентам в 1-м семестре

3.86 — средняя оценка по всем студентам во 2-м семестре

## **Пример 4. Оценка стоимости дома**

Предположим, что имеются данные о жилых загородных домах в некоторой местности. Для каждого дома известна его цена, состояние, жилая площадь, количество этажей, количество комнат, время постройки, удаленность до основных магистралей, наличие инфраструктуры, экологическая обстановка в районе и т. п.

Требуется научиться оценить цену по остальной информации.

Объектами являются дома, входами — их характеристики, а выходом — цена дома.

Это задача восстановления регрессии.

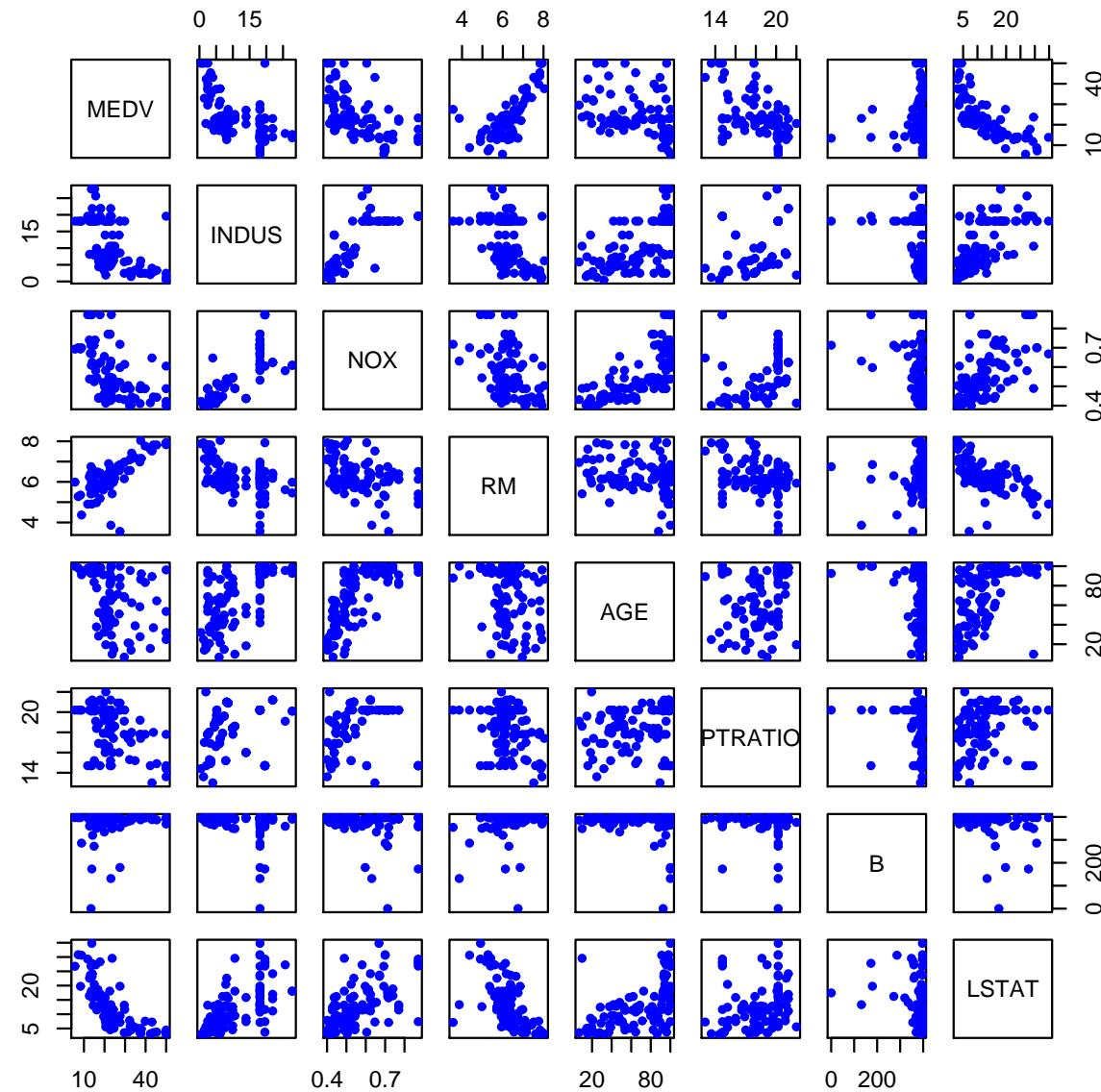
Boston Housing Data <http://archive.ics.uci.edu/ml/datasets/Housing>

Информация агрегирована: территория поделена на участки и дома, стоящие на одном участке, собраны в группы. Нужно оценить среднюю цену дома. Таким образом, объектами являются сами эти группы. Их общее количество — 506.

## Признаки

1. CRIM — уровень преступности на душу населения,
2. ZN — процент земли, застроенной жилыми домами (только для участков площадью свыше 25000 кв. футов),
3. INDUS — процент деловой застройки,
4. CHAS — 1, если участок граничит с рекой; 0 в противном случае (бинарный признак),
5. NOX — концентрация оксида азота, деленная на  $10^7$ ,
6. RM — среднее число комнат (по всем домам рассматриваемого участка),
7. AGE — процент домов, построенных до 1940 г. и занимаемых владельцами,
8. DIS — взвешенное расстояние до 5 деловых центров Бостона,
9. RAD — индекс удаленности до радиальных магистралей,
10. TAX — величина налога в \$10000,
11. PTRATIO — количество учащихся, приходящихся на одного учителя (по городу),
12.  $B = 1000(AA - 0.63)^2$ , где AA — доля афро-американцев,
13. LSTAT — процент жителей с низким социальным статусом.

Диаграммы рассеяния для каждой пары переменных MEDV, INDUS, NOX, RM, AGE, PTRATIO, B. Значение переменной MEDV нужно научиться предсказывать по значениям остальных переменных. Изображены только по 100 случайных точек.



# Некоторые методы обучения с учителем

- Линейный метод наименьших квадратов
- Линейный и квадратичный дискриминантный анализ
- Логистическая регрессия
- Метод  $k$  ближайших соседей
- Наивный байесовский классификатор
- Деревья решений (C4.5, CART и др.)
- Персепtron и нейронные сети
- Машина опорных векторов (SVM)
- Ансамбли решающих правил (бустинг, баггинг и т. п.)
- ...

См., например,

Top 10 algorithms in data mining // Knowl. Inf. Syst. 2008. № 14. P. 1–37

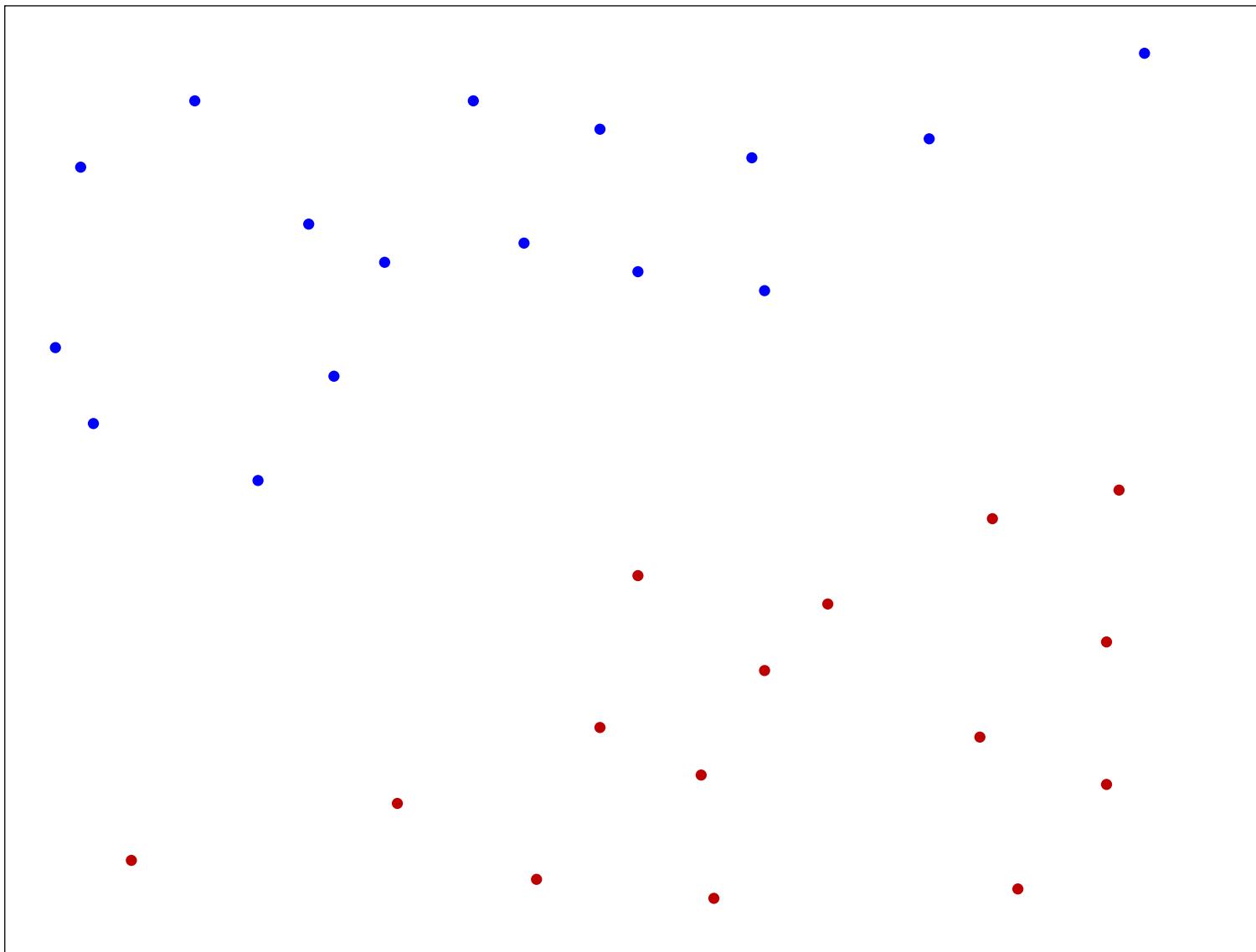
## **2.1. Машина опорных векторов**

Support Vector Machine (SVM)

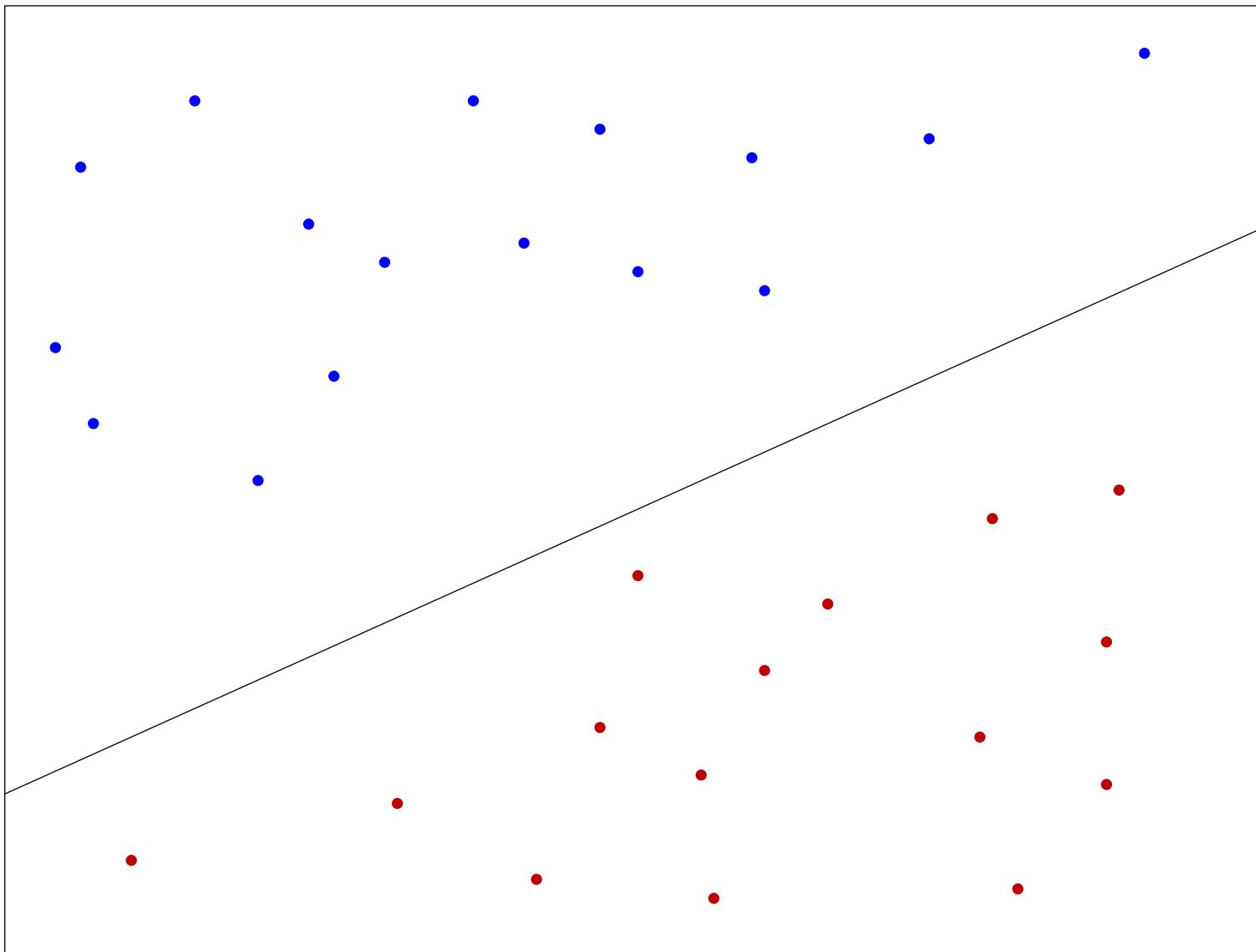
Машина опорных векторов (support vector machine) — один из методов построения решающего правила.

Основы метода (оптимальная разделяющая гиперплоскость) предложены В.Н. Вапником и А.Я. Червоненкисом в 1974.

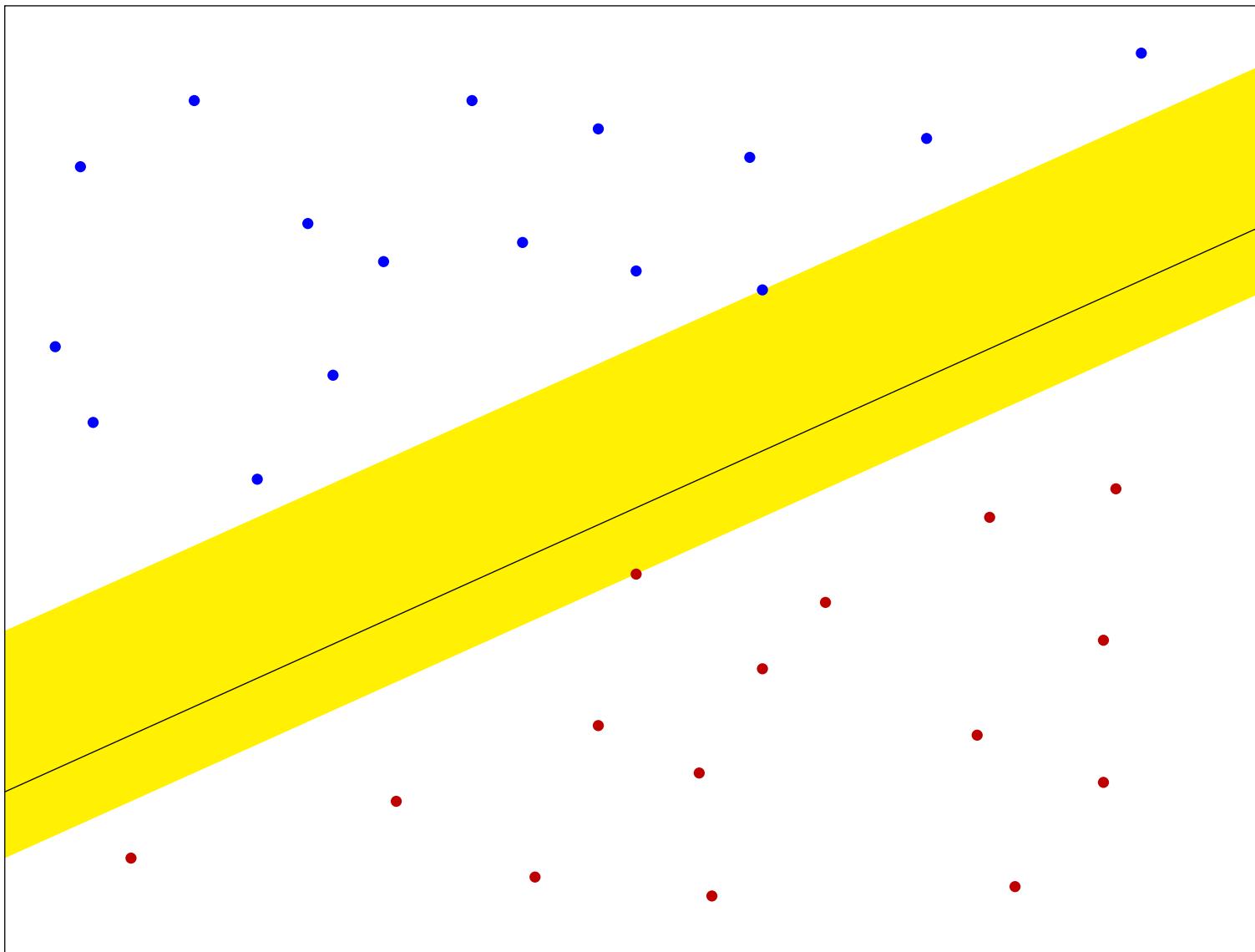
Современные очертания (использование ядер) метод принял в начале 90-х гг. (В.Н. Вапник и др.)



Два класса



Разделяющая гиперплоскость



Желтая – разделяющая («нейтральная») полоса

*Оптимальная разделяющая гиперплоскость* — это гиперплоскость, разделяющая объекты двух классов, такая, что расстояние от нее до ближайшей точки (из каждого класса) максимально.

Т. е. оптимальная разделяющая гиперплоскость лежит в центре разделяющей полосы и толщина этой полосы максимальна.

Она максимизирует *зазор (margin)* между плоскостью и данными из обучающей выборки — это приводит, как правило, к хорошим результатам и на тестовых данных.

Обучающая выборка:

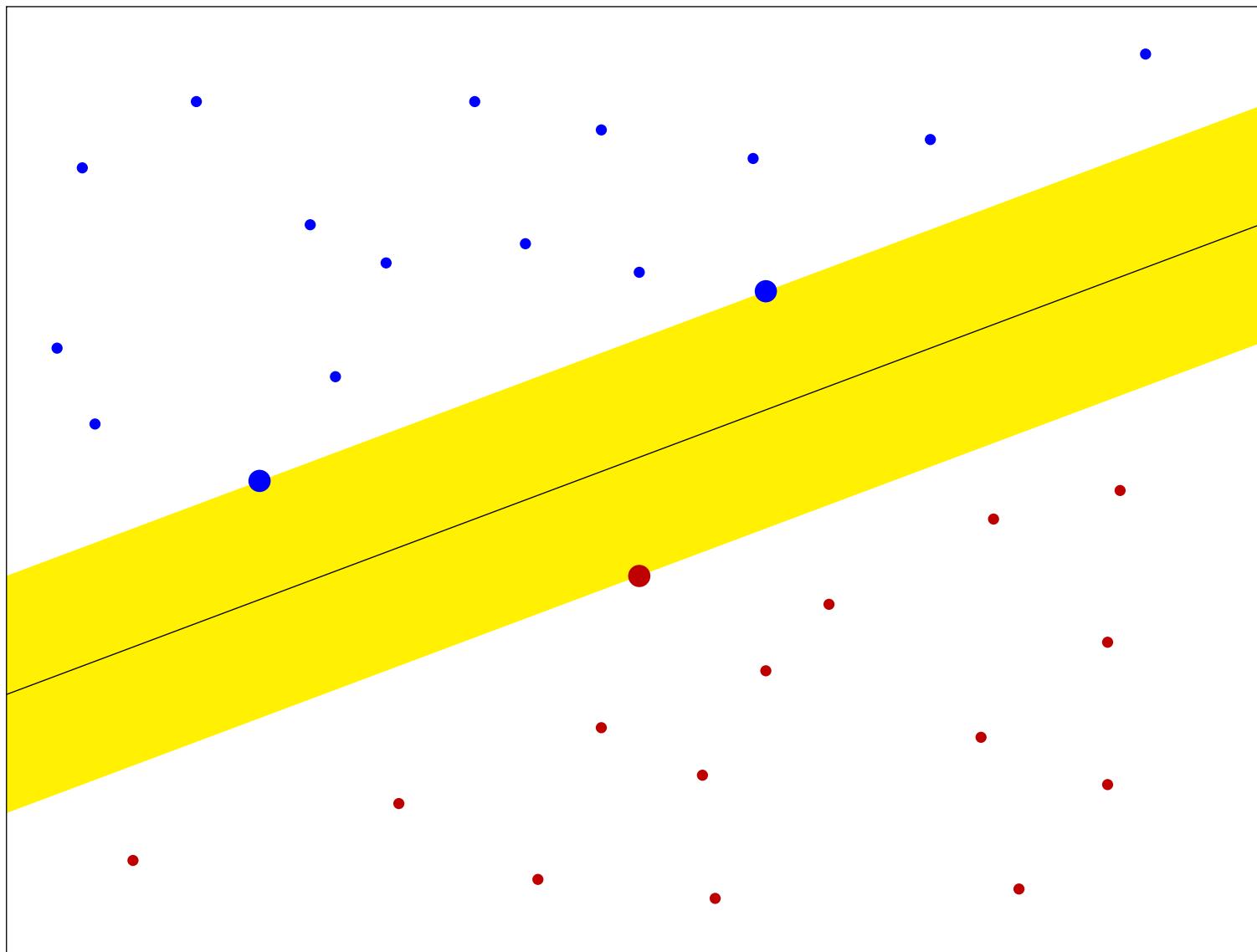
$$(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N),$$

Задача нахождения оптимальной гиперплоскости записывается следующим образом:

$$\max_{\beta, \beta_0, \|\beta\|=1} C$$

при ограничениях

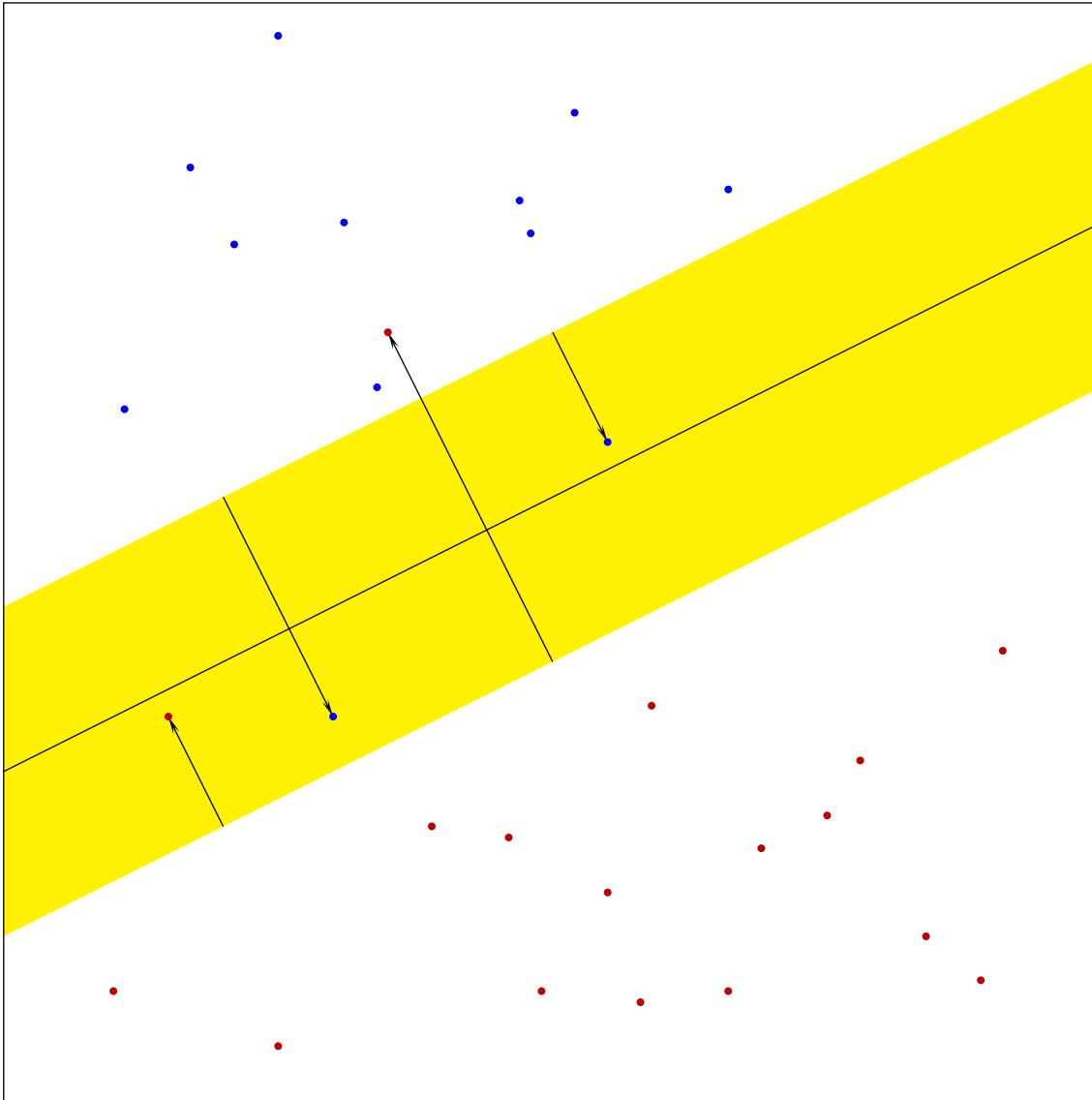
$$y_i(x_i^\top \beta + \beta_0) \geq C \quad (i = 1, 2, \dots, N)$$



Что делать, если точки нельзя разделить гиперплоскостью на два заданных класса?

В этом случае решают несколько измененную задачу.

В ее формулировке фигурирует некоторый настроечный параметр, с помощью которого можно регулировать количество точек из обучающей выборки, которые попадут в нейтральную полосу или окажутся на «чужой территории»



## 2.2. Деревья решений

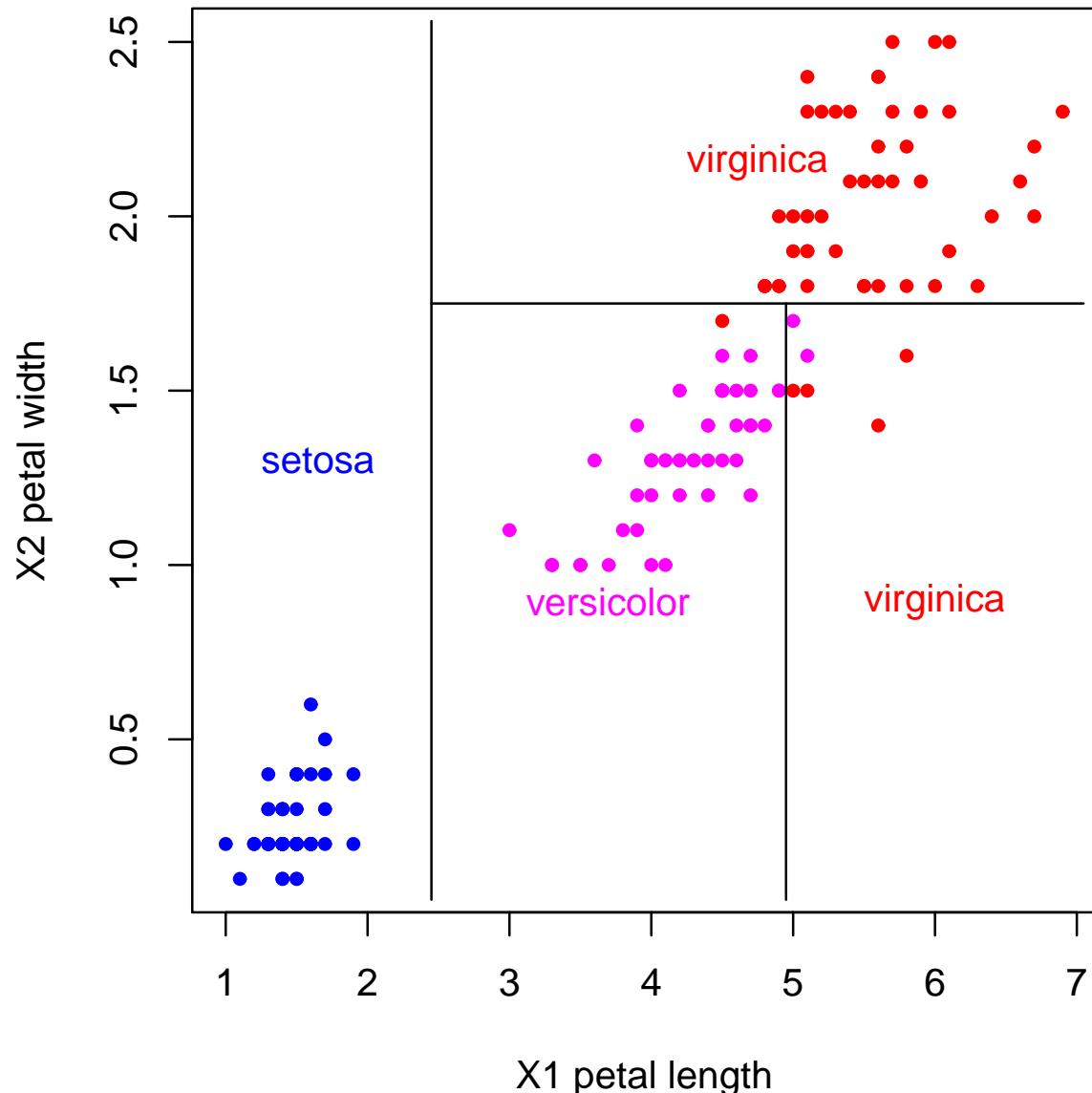
Пространство признаков разбивается на параллелепипеды со сторонами, параллельными осям координат (ящики).

В каждом ящике ответ аппроксимируется с помощью некоторой простой модели, например, константой.

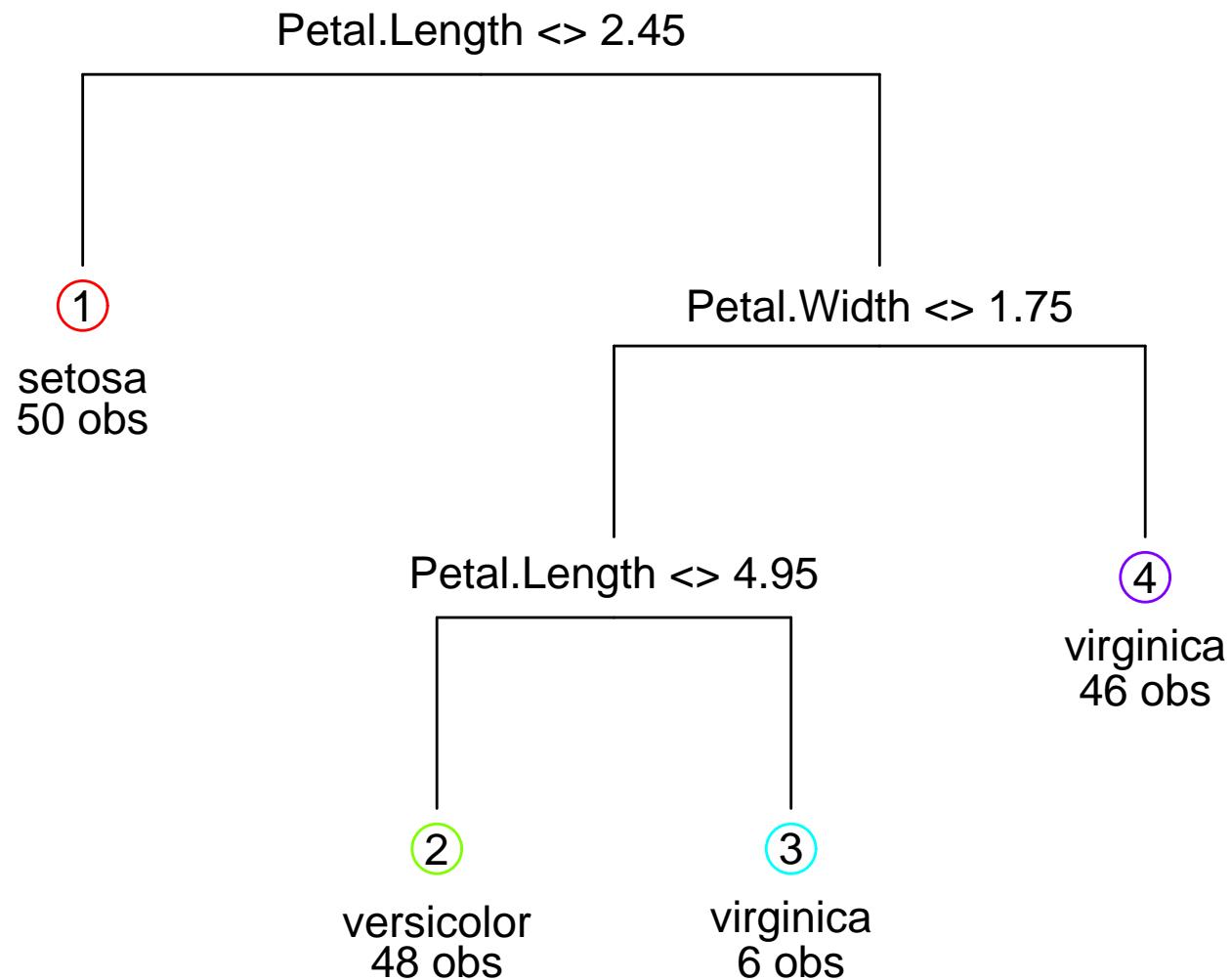
Используются только рекурсивные гильотинные разбиения.

Задача классификации цветов ириса (Fisher, 1936).

$X_1, X_2$  – длина и ширина чашелистика.



Дерево решений:



## Популярные алгоритмы построения деревьев решений

- See5/C5.0 [Quinlan et., 1997] ← C4.5 [Quinlan, 1993] ← ID3 [Quinlan, 1979] ← CLS [Hunt, Marin, Stone, 1966]
- CART – Classification and Regression Trees [Breiman, Friedman, Olshen, Stone, 1984]

# Достоинства и недостатки деревьев решений

*Достоинства:*

- Поддерживают работу с входными переменными разных (смешанных) типов
- Возможность обрабатывать данные с пропущенными значениями
- Устойчивы к выбросам
- Нечувствительность к монотонным преобразованиям входных переменных
- Поддерживают работу с большими выборками
- Возможность интерпретации построенного решающего правила

*Основной недостаток* — плохая предсказательная (обобщающая) способность.

## 2.3. Ансамбли решающих правил

Ансамбль, или *комитет*, решающих правил,

или *аркинг* (arcing – adaptive reweighting and combining) — комбинирование решающих правил.

Рассмотрим задачу классификации на  $K$  классов.

$$\mathcal{Y} = \{1, 2, \dots, K\}.$$

Пусть имеется  $M$  классификаторов («экспертов»)  $f_1, f_2, \dots, f_M$

$$f_m : \mathcal{X} \rightarrow \mathcal{Y}, \quad f_m \in \mathcal{F}, \quad (m = 1, 2, \dots, M)$$

Построим новый классификатор:

простое голосование:

$$f(x) = \max_{k=1, \dots, K} \sum_{m=1}^M I(f_m(x) = k),$$

взвешенное (выпуклая комбинация классификаторов, или «смесь экспертов»):

$$f(x) = \max_{k=1, \dots, K} \sum_{m=1}^M \alpha_m \cdot I(f_m(x) = k), \quad \alpha_m \geq 0, \quad \sum_{m=1}^M \alpha_m = 1,$$

В задаче восстановления регрессии

простое голосование:

$$f(x) = \frac{1}{M} \sum_{m=1}^M f_m(x),$$

взвешенное голосование («смесь экспертов»):

$$f(x) = \sum_{m=1}^M \alpha_m \cdot f_m(x), \quad \alpha_m \geq 0, \quad \sum_{m=1}^M \alpha_m = 1.$$

Пример:  $K = 2$ ,  $M = 3$ .

Решение принимается с использованием простого голосования.

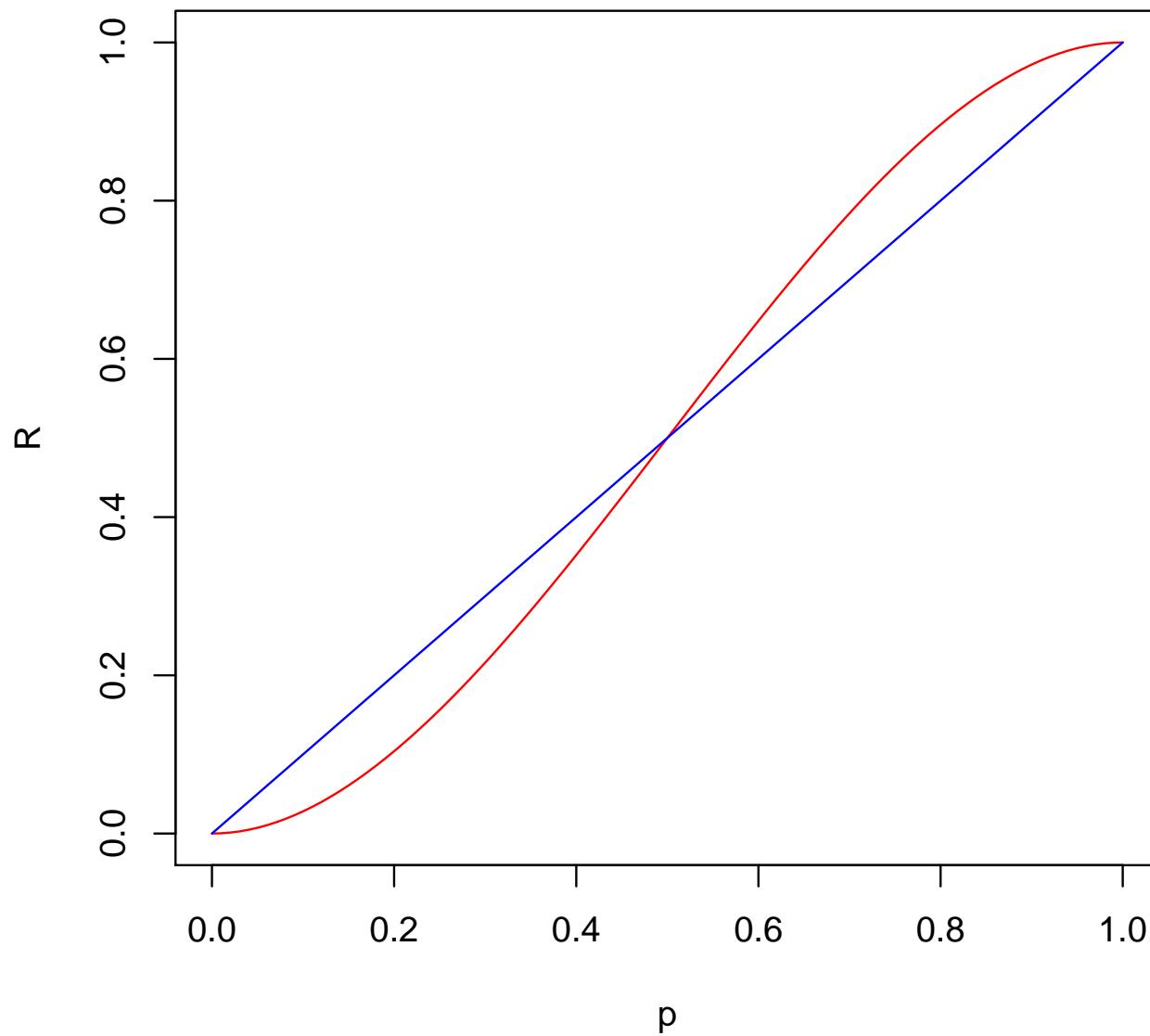
*Пусть классификаторы независимы* (на практике недостижимое требование!).

$p$  — вероятность ошибки каждого отдельного классификатора.

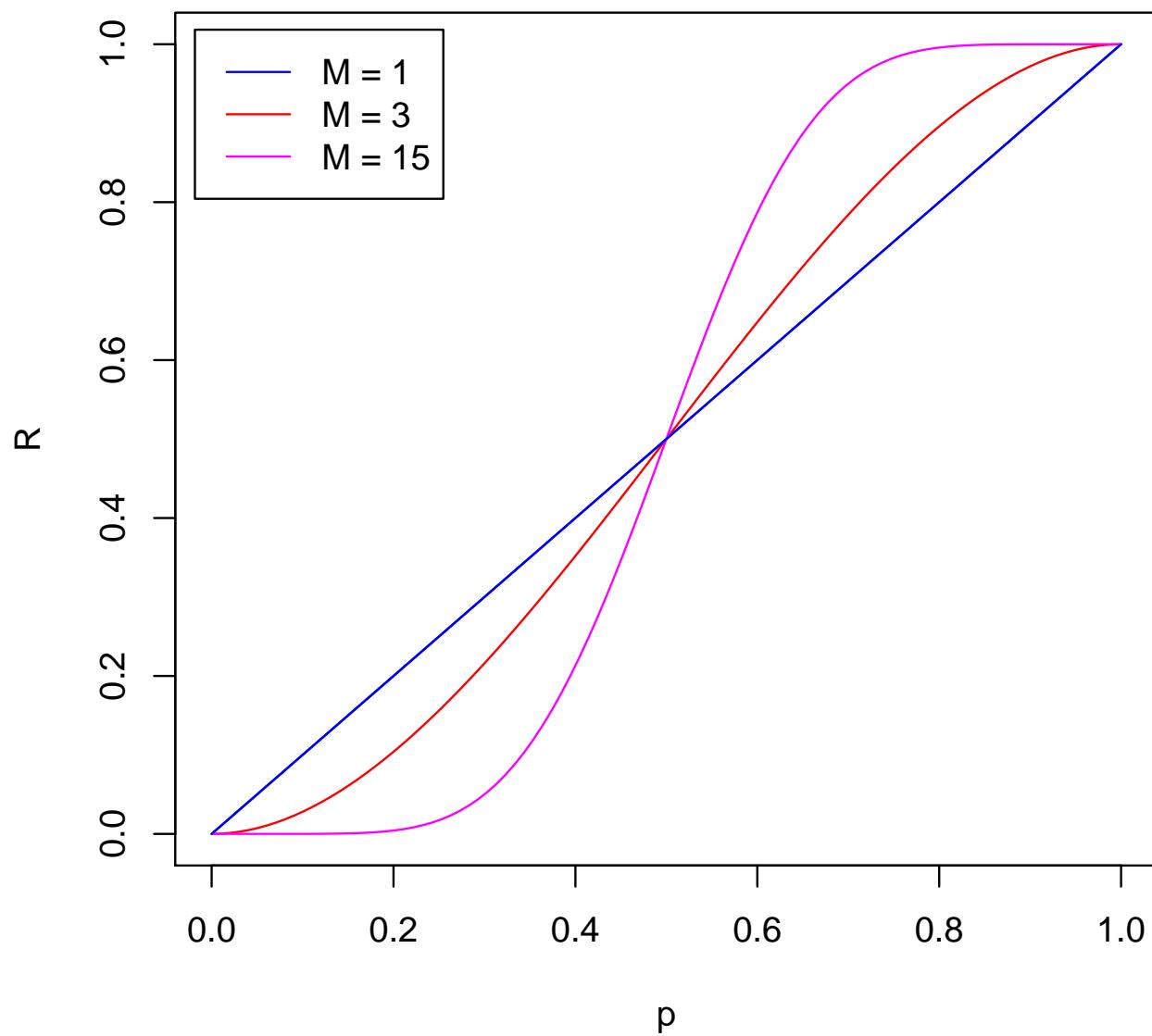
Тогда вероятность ошибки общего решения (ожидаемый риск) равен

$$R = p^3 + 3p^2(1 - p) = 3p^2 - 2p^3.$$

$$R = 3p^2 - 2p^3$$



$$M = 1, 3, 15$$



$K = 2$

- $p > \frac{1}{2}$  — плохой классификатор,
- $p = \frac{1}{2}$  — toss up a coin
- $p = \frac{1}{2} - \varepsilon$  — хороший, но слабый классификатор ( $\varepsilon$  мало)
- $p = \varepsilon$  — сильный классификатор

Можно ли научиться комбинировать слабые классификаторы, чтобы получить сильный [Kearns, Valiant, 1988]?

Два известных подхода:

- *Баггинг* и т. п.: пытаемся снизить зависимость экспертов друг от друга.
- *Бустинг*: эксперты учатся на ошибках других.

## **2.3.1. Бустинг**

То boost — улучшать, повышать, рекламировать.

Строим последовательность решающих правил, каждый из которых осведомлен об ошибках предыдущих.

Пусть только два класса  $\mathcal{Y} = \{-1, 1\}$ .

Простая схема [Schapire, 1990]

AdaBoost [Freund, Schapire, 1995]

## AdaBoost [Freund, Schapire, 1995]

(от Adaptive Boosting)

Будем использовать веса  $w_1, w_2, \dots, w_N$ .

На первой итерации  $w_i = 1/N$  ( $i = 1, 2, \dots, N$ ) и алгоритм построения  $f_1$  работает в обычном режиме.

На  $m$ -й итерации увеличиваются веса тех прецедентов, на которых на  $(m - 1)$ -й итерации была допущена ошибка, и уменьшаются веса тех прецедентов, которые на предыдущей итерации были классифицированы правильно.

Как учитывать веса  $w_i$ ?

— Некоторые алгоритмы обучения принимают на вход веса  $w_i$ .

Если это не возможно, то на каждой итерации генерировать бутстрэп выборку, изымая (с возвращением) из обучающей выборки

$$(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)$$

$i$ -й элемент с вероятностью  $w_i$ .

**begin** AdaBoost

Положить  $w_i \leftarrow 1/N \quad (i = 1, 2, \dots, N)$

**for**  $m = 1, 2, \dots, M$

Найти классификатор  $h_m \in \mathcal{F}_m$ , минимизирующий ошибку

$$\text{err}_m = \sum_{i=1}^N w_i \cdot I(y_i \neq h_m(x_i))$$

Вычислить  $\beta_m \leftarrow \frac{1}{2} \ln \frac{1 - \text{err}_m}{\text{err}_m}$

$$w'_i \leftarrow w_i \cdot e^{-\beta_m y_i h_m(x_i)} = w_i \cdot \left( \sqrt{\frac{\text{err}_m}{1 - \text{err}_m}} \right)^{-y_i h_m(x_i)} \quad (i = 1, 2, \dots, N)$$

$$w_i \leftarrow w'_i / \sum_{i=1}^N w'_i \quad (i = 1, 2, \dots, N)$$

**end**

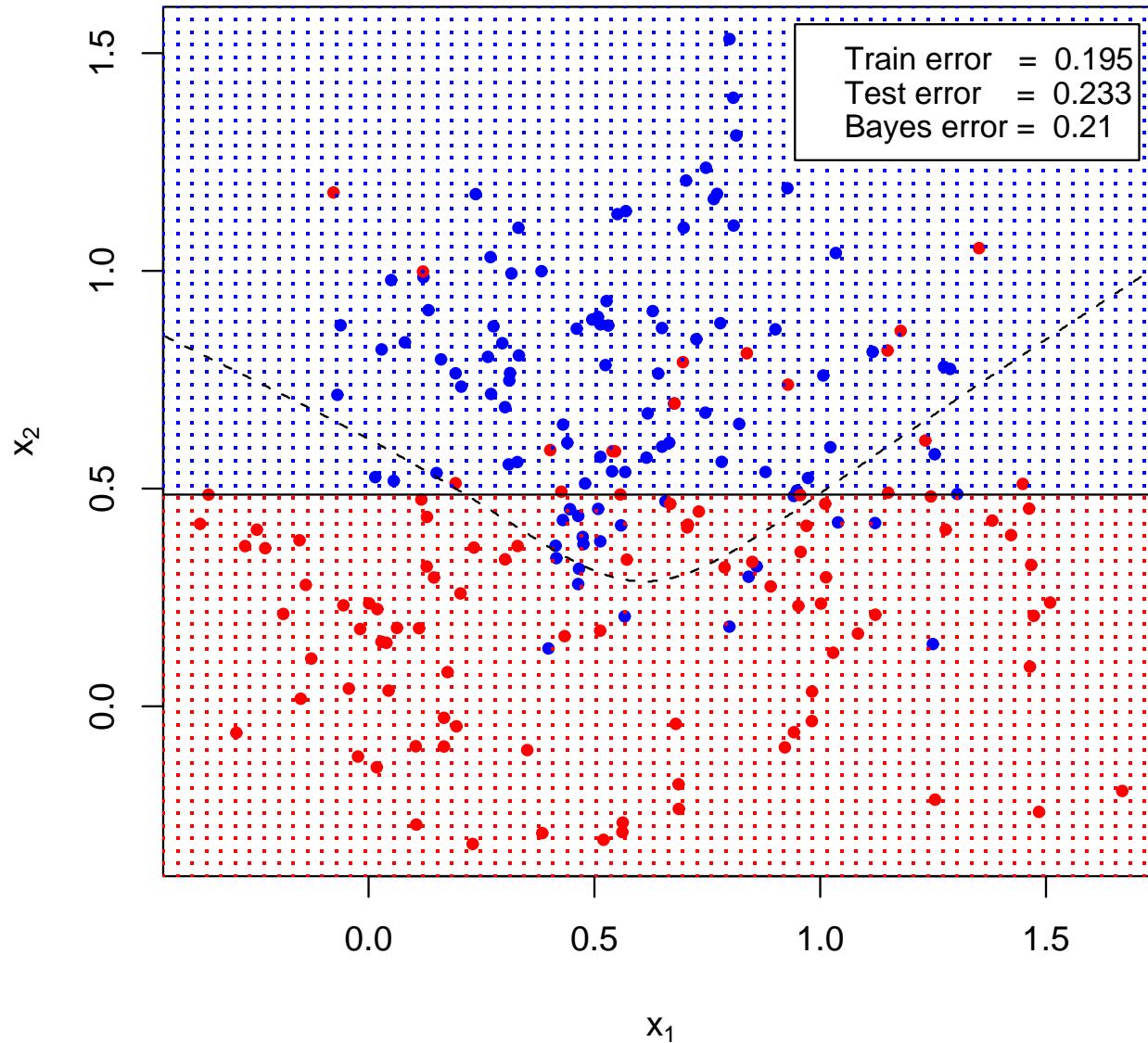
**return**  $f = \text{sign } g$ , где  $g = \sum_{m=1}^M \beta_m h_m$

**end**

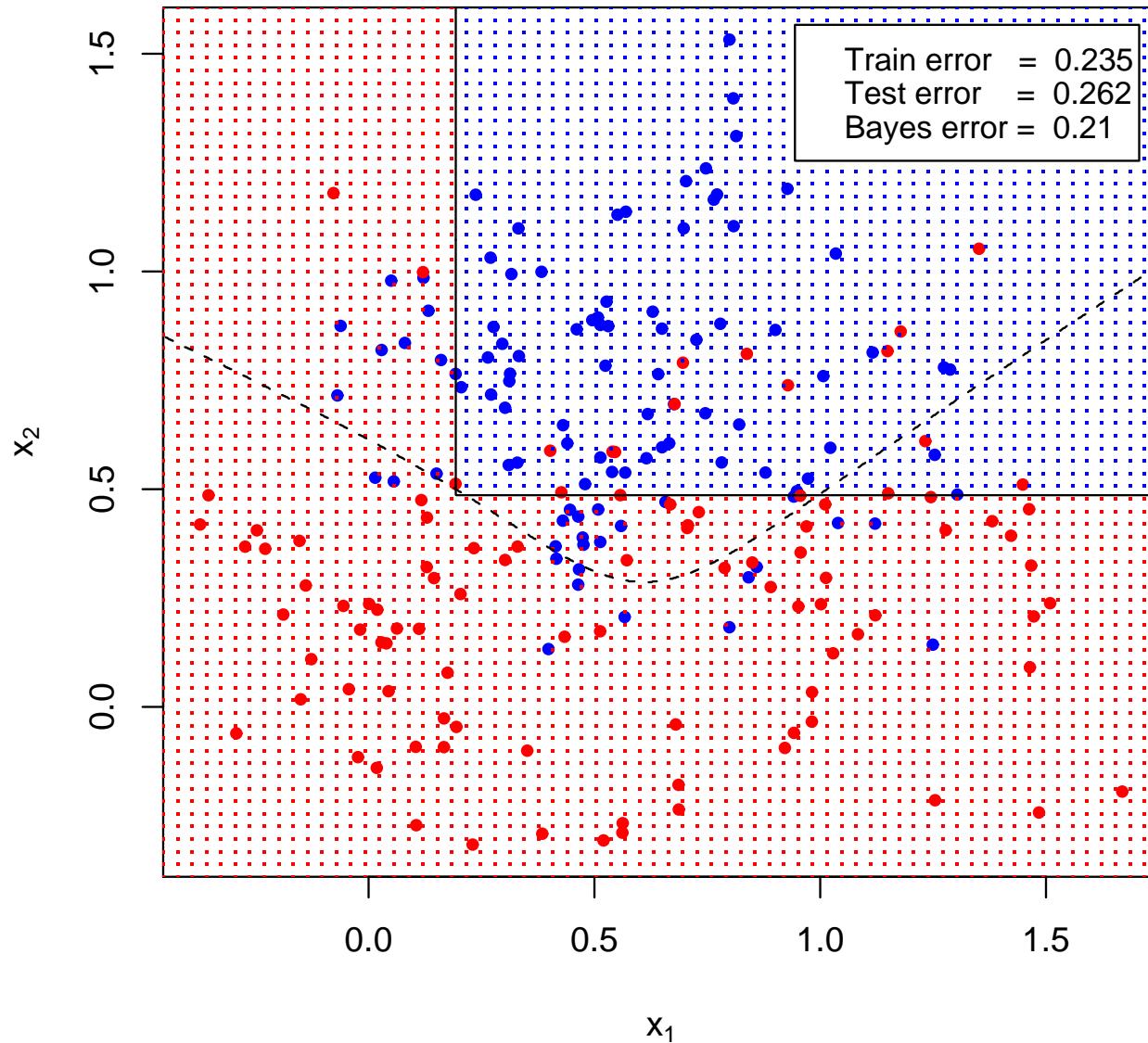
## Пример

Слабые классификаторы — деревья решений высоты 1 (*stumps*)

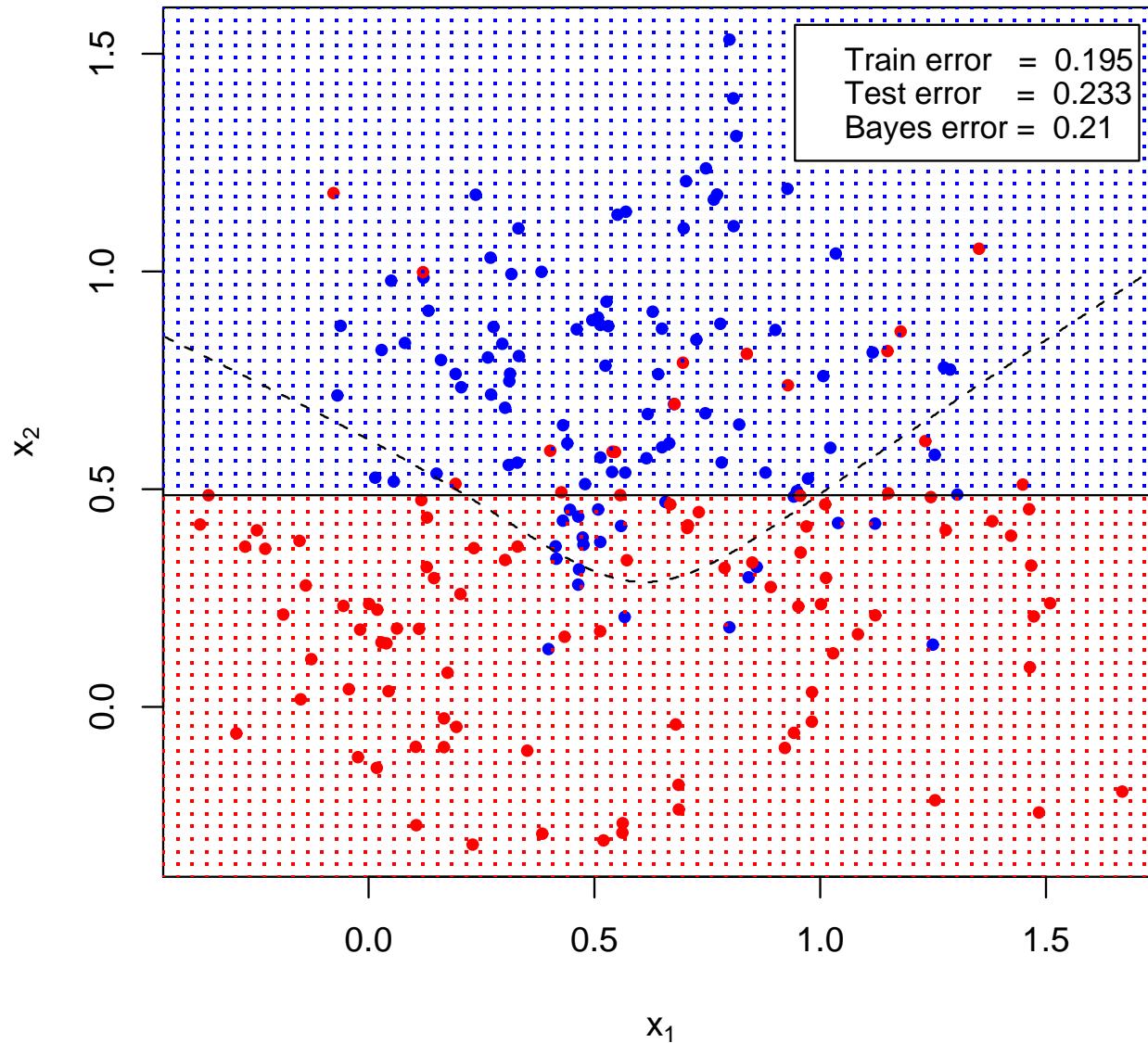
$M = 1$



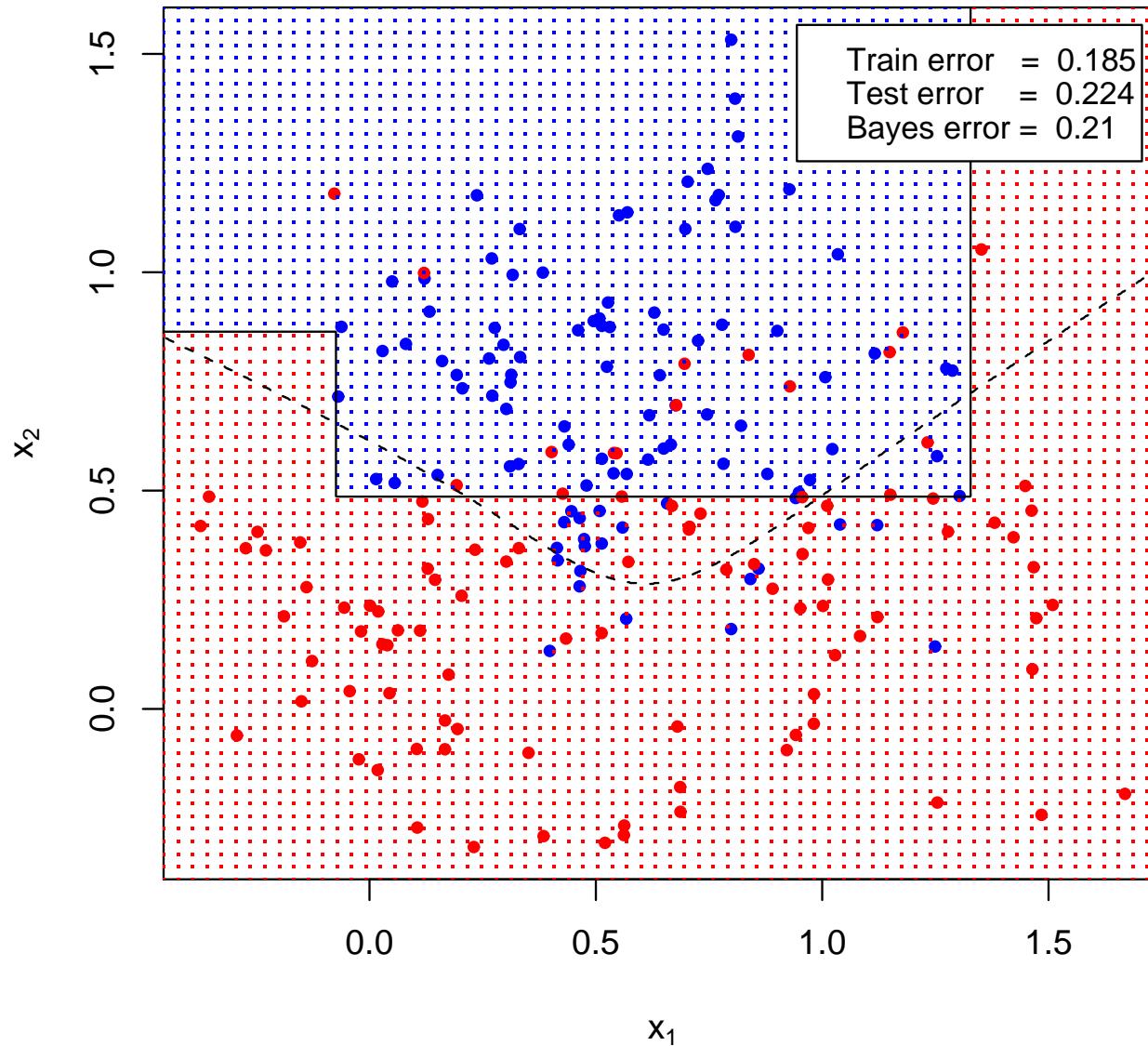
$M = 5$



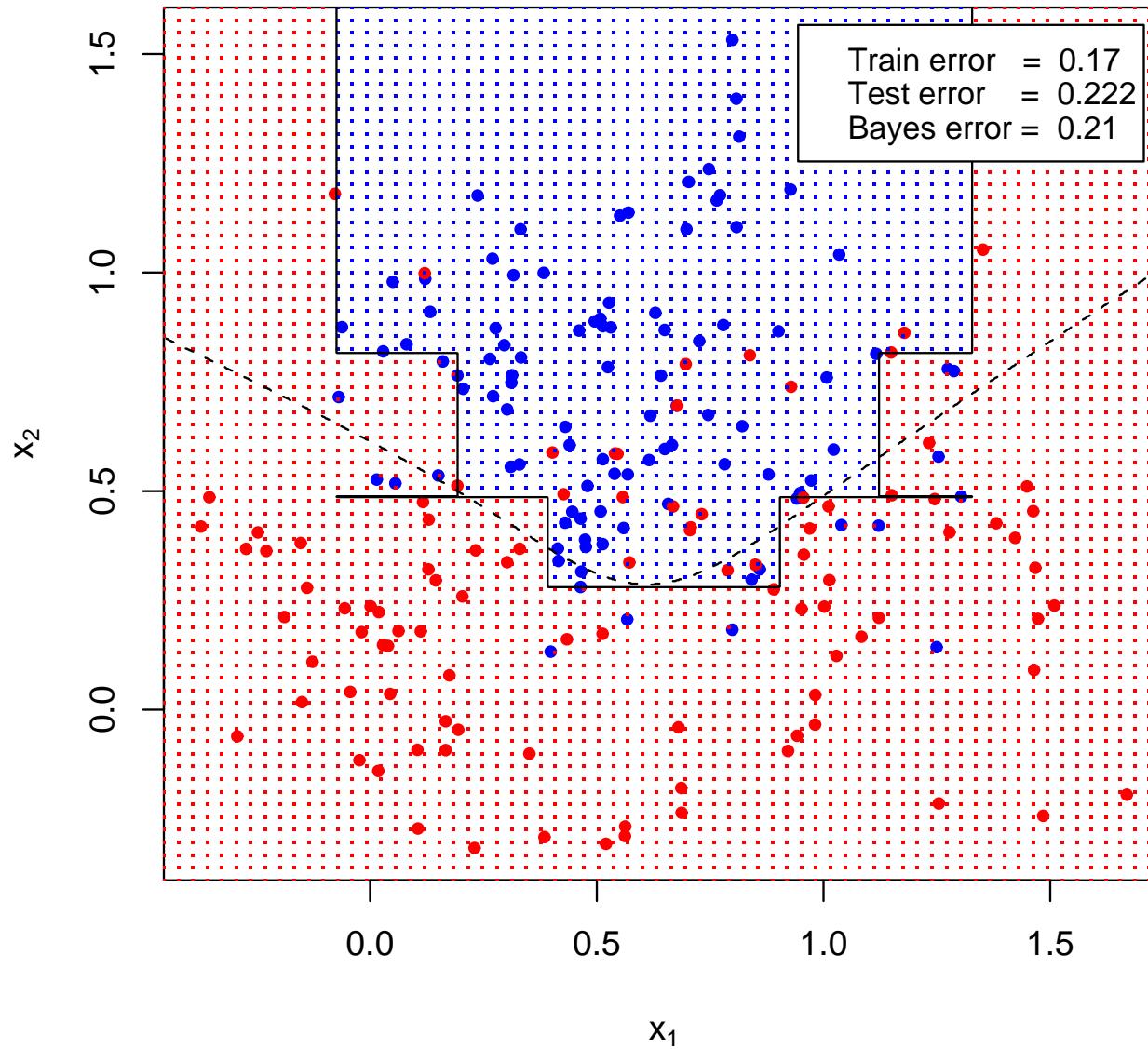
$$M = 10$$



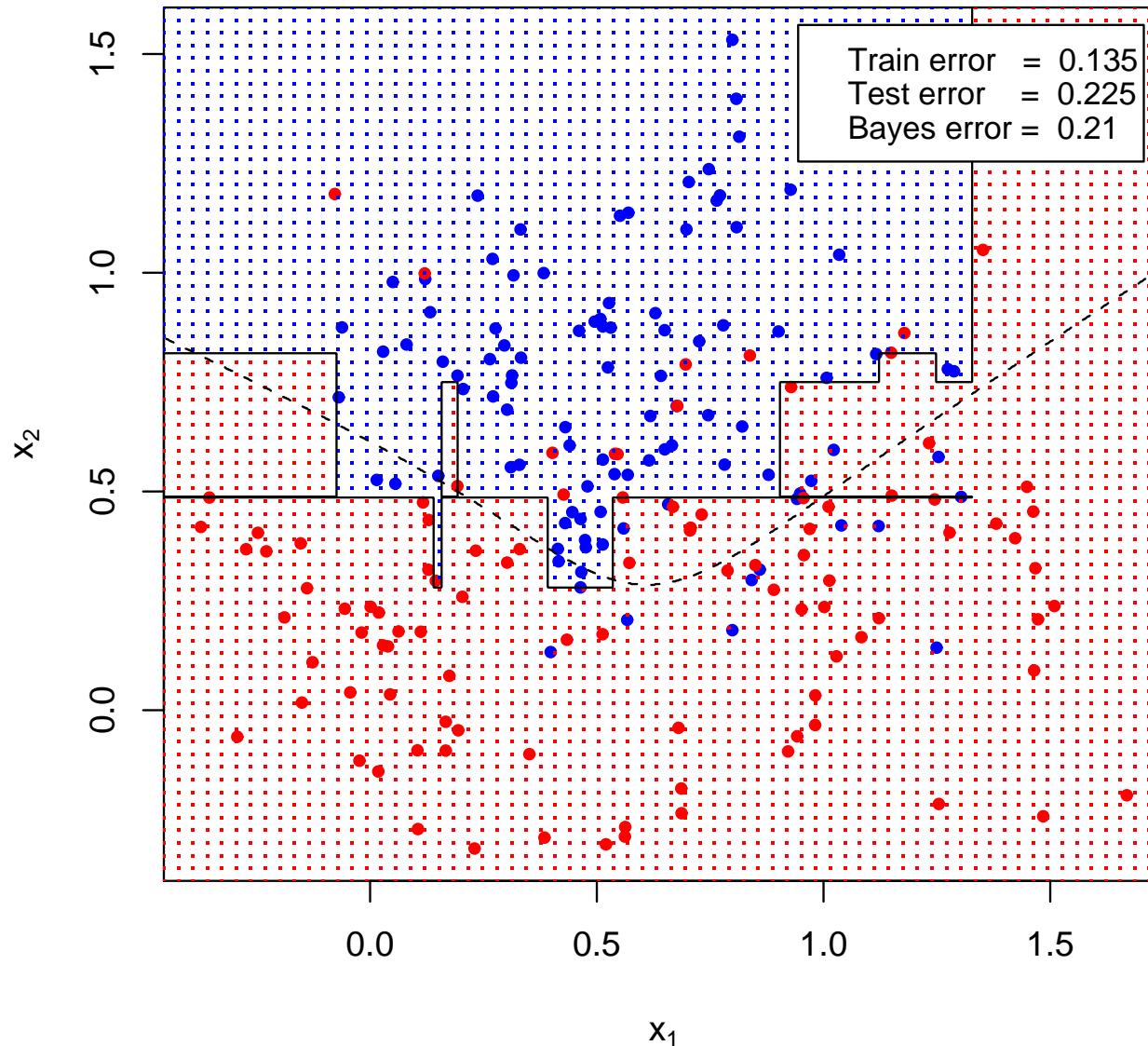
$M = 15$



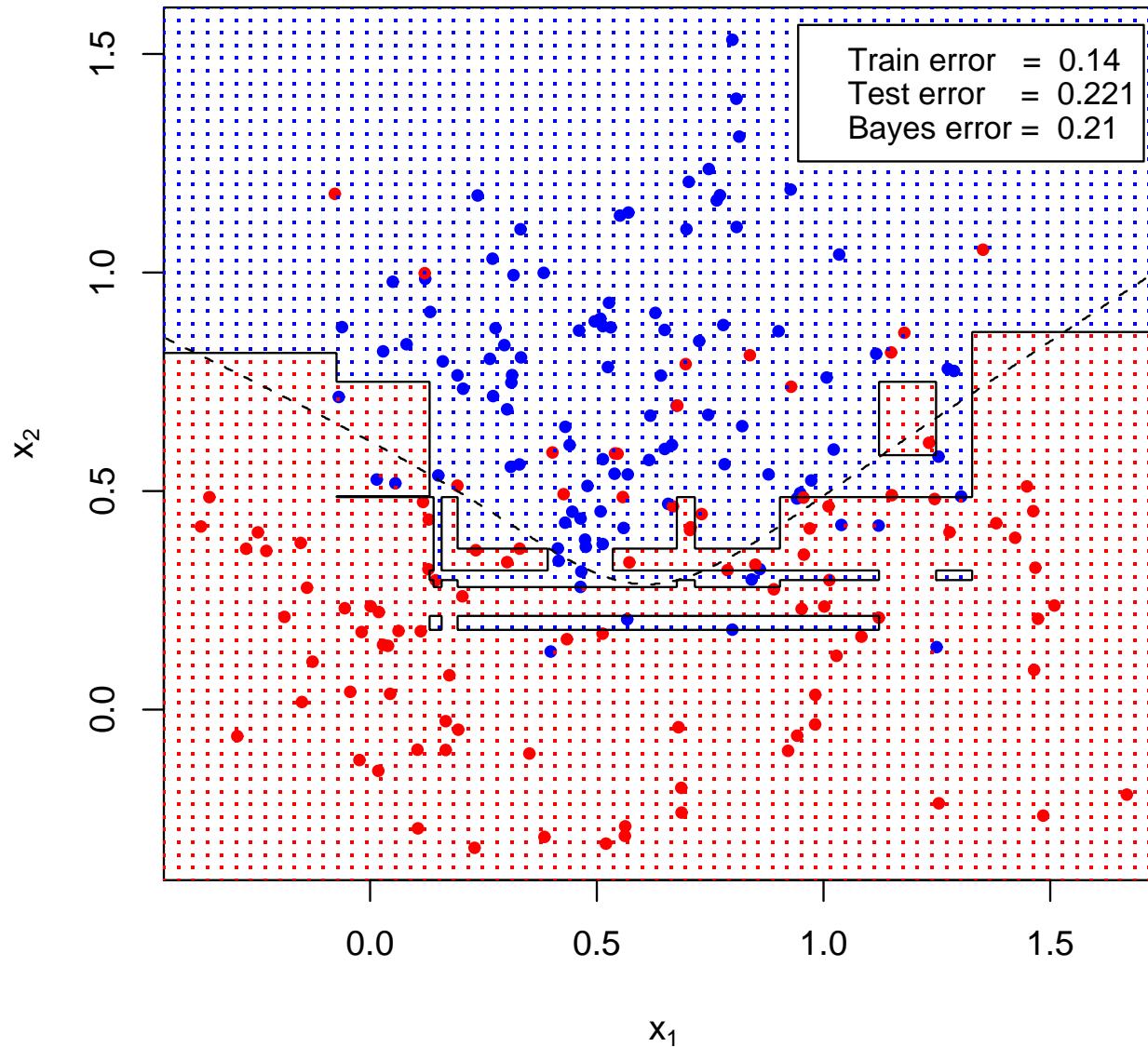
$$M = 20$$



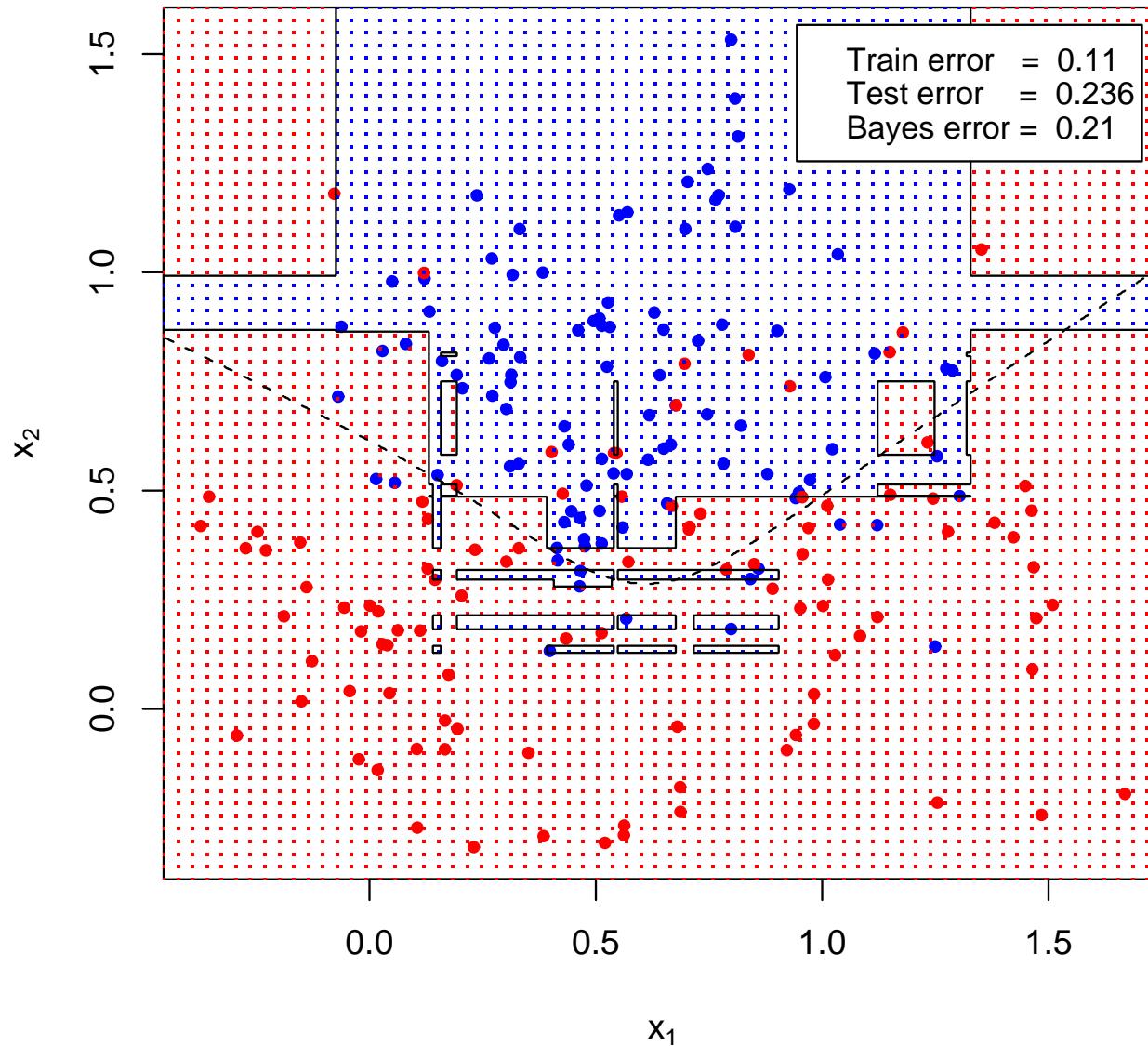
$M = 30$



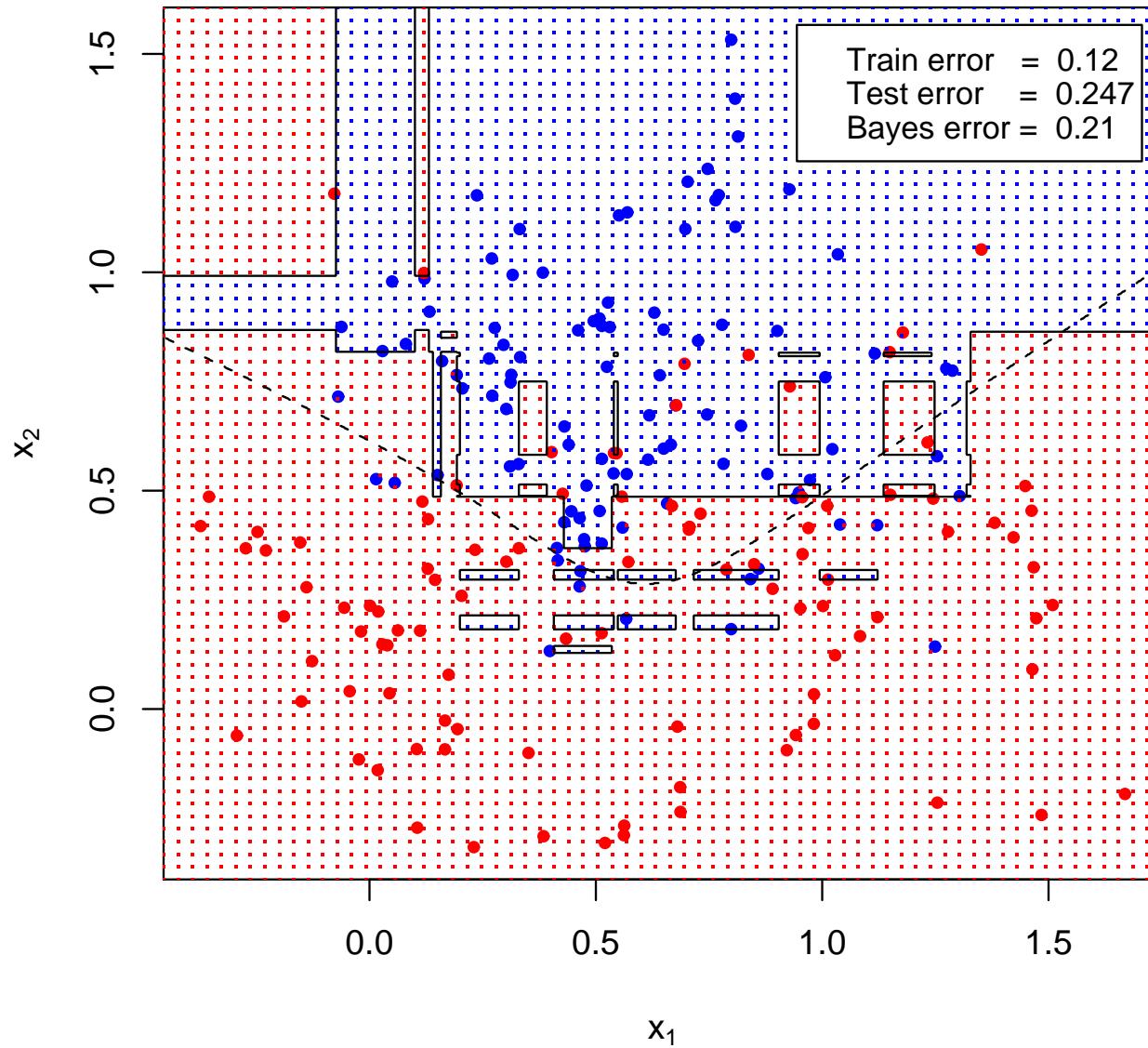
$M = 40$



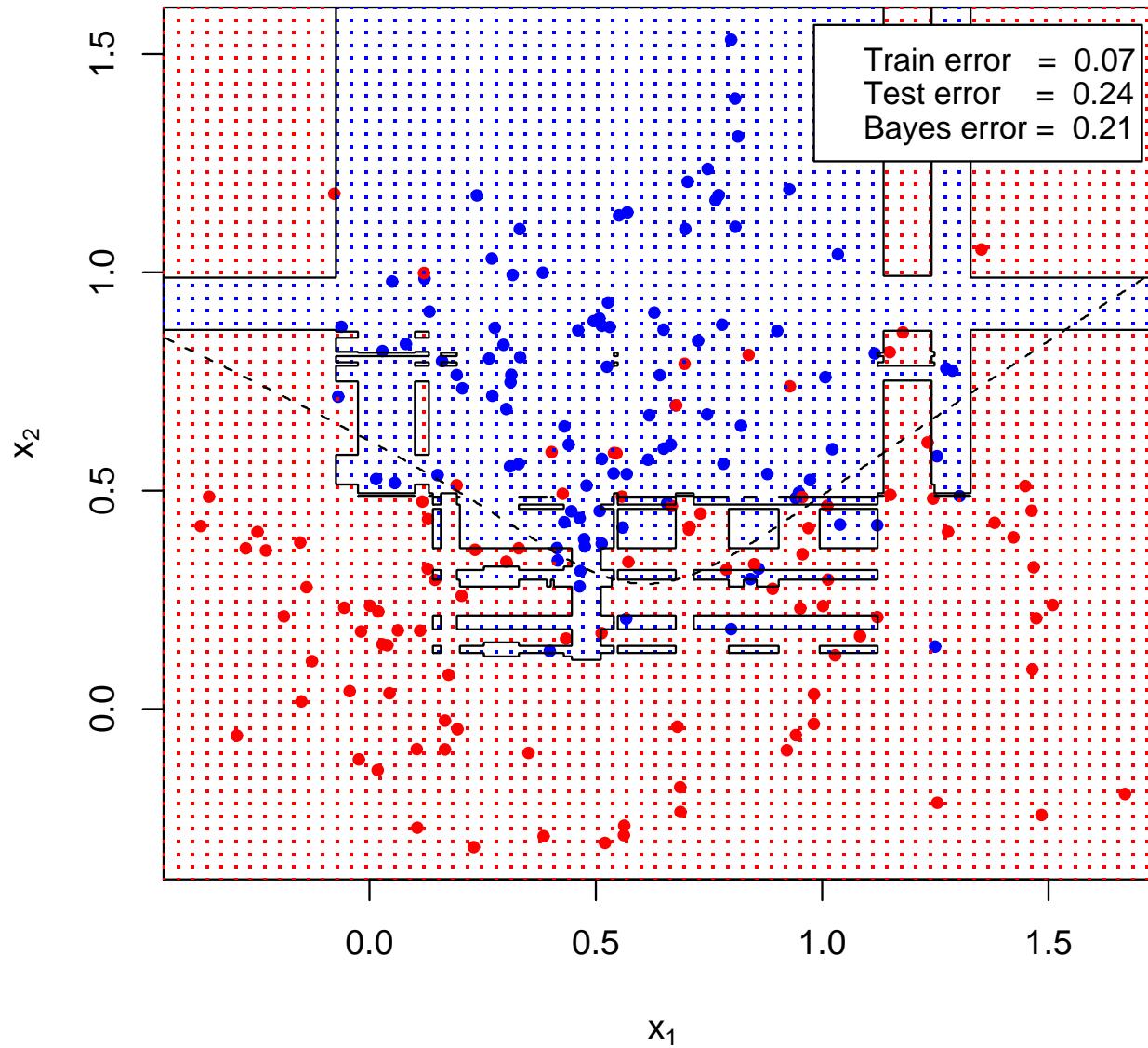
$M = 50$



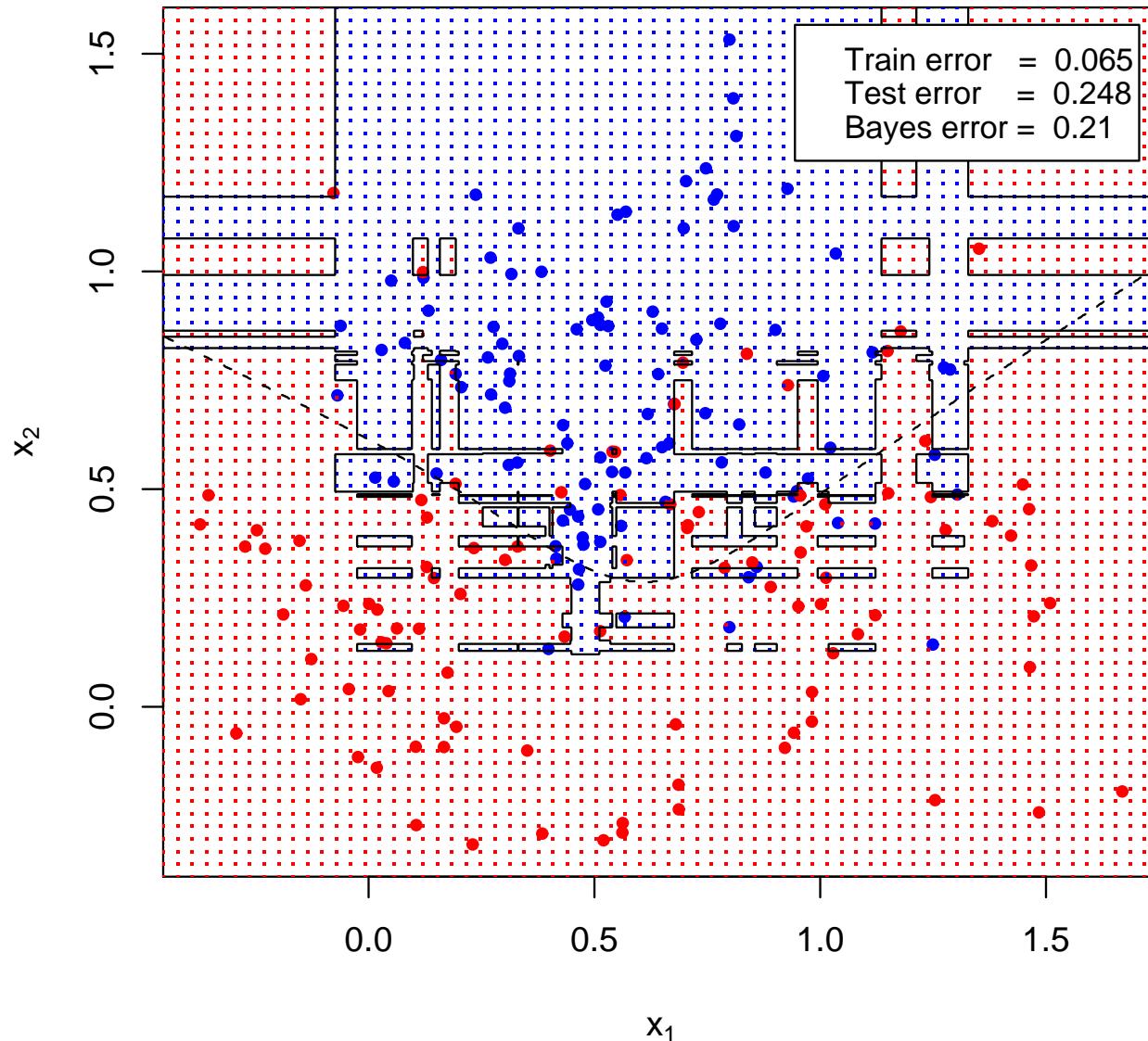
$M = 60$



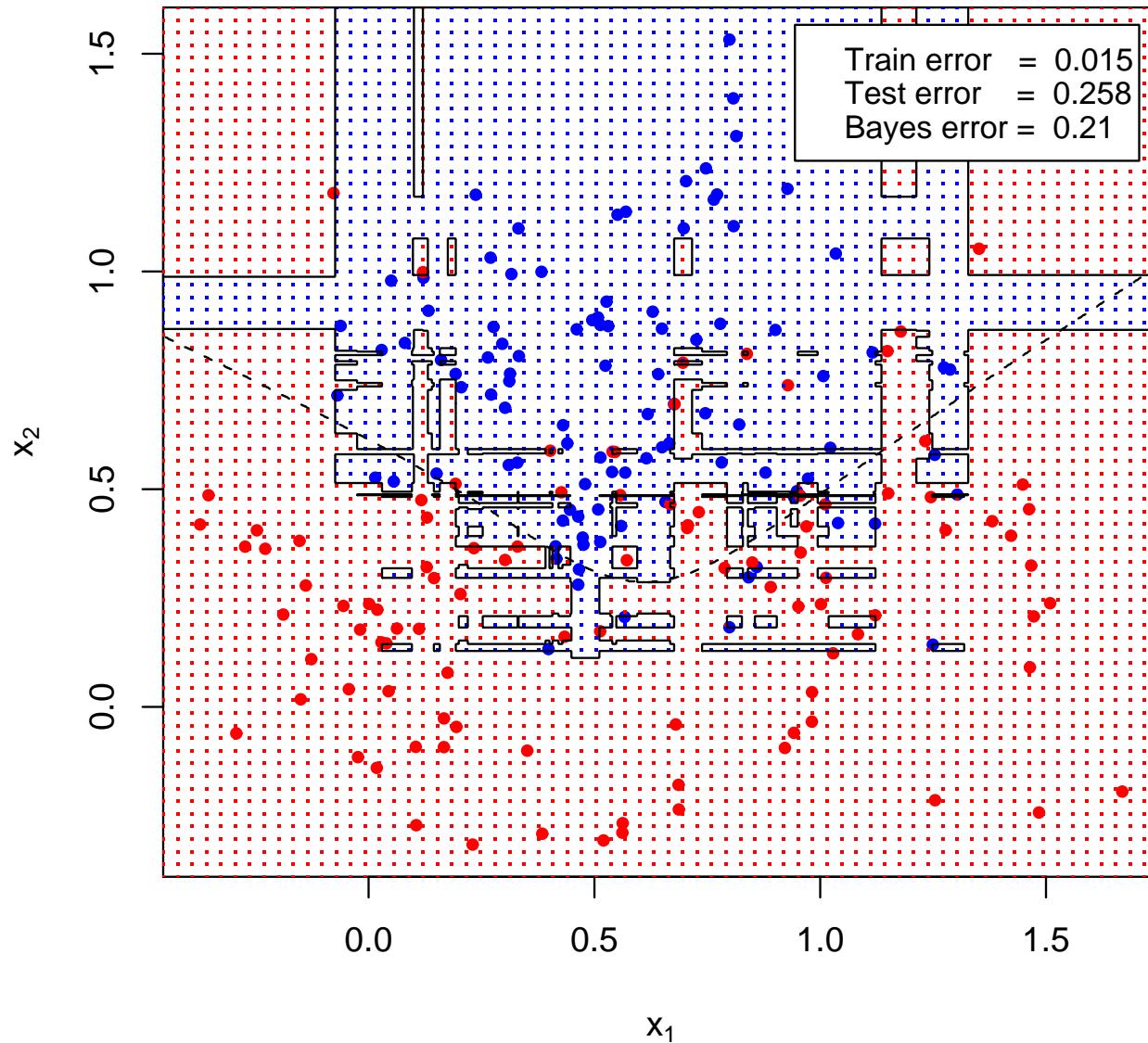
$M = 100$



$M = 150$



$M = 200$



## Пример. Классификации рукописных цифр

Выборка размера 1934 разбита случайно поровну на обучающую и тестовую.

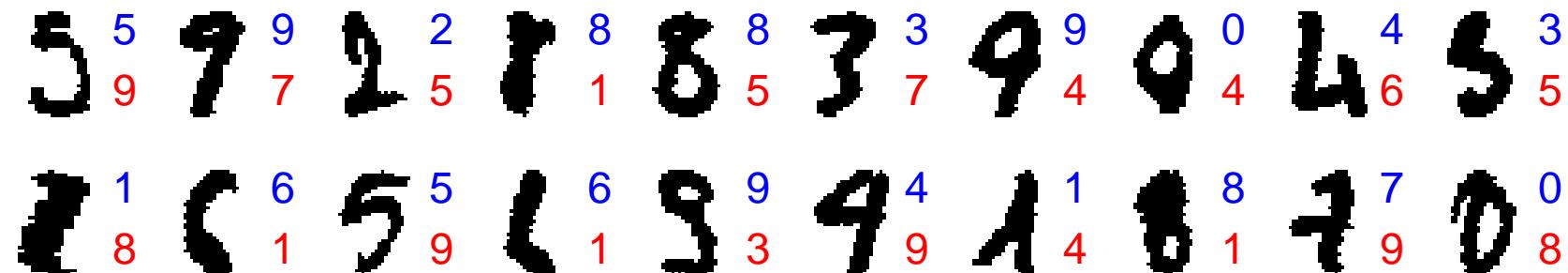
В качестве слабых классификаторов использовались деревья решений, высоты  $\leq 10$ .

M	<i>Ошибка</i>	
	<i>на обучающей выборке</i>	<i>на тестовой выборке</i>
5	0.070	0.121
10	0.017	0.074
20	0.001	0.056
30	0	0.048
40	0	<b>0.044</b>

<i>Алгоритм</i>	<i>Ошибка на обучающей выборке на тестовой выборке</i>	
Машина опорных векторов	0 %	2.1 %
Метод ближайшего соседа	0 %	3.1 %
Нейронная сеть	0 %	4.7 %
AdaBoost	0 %	4.8 %

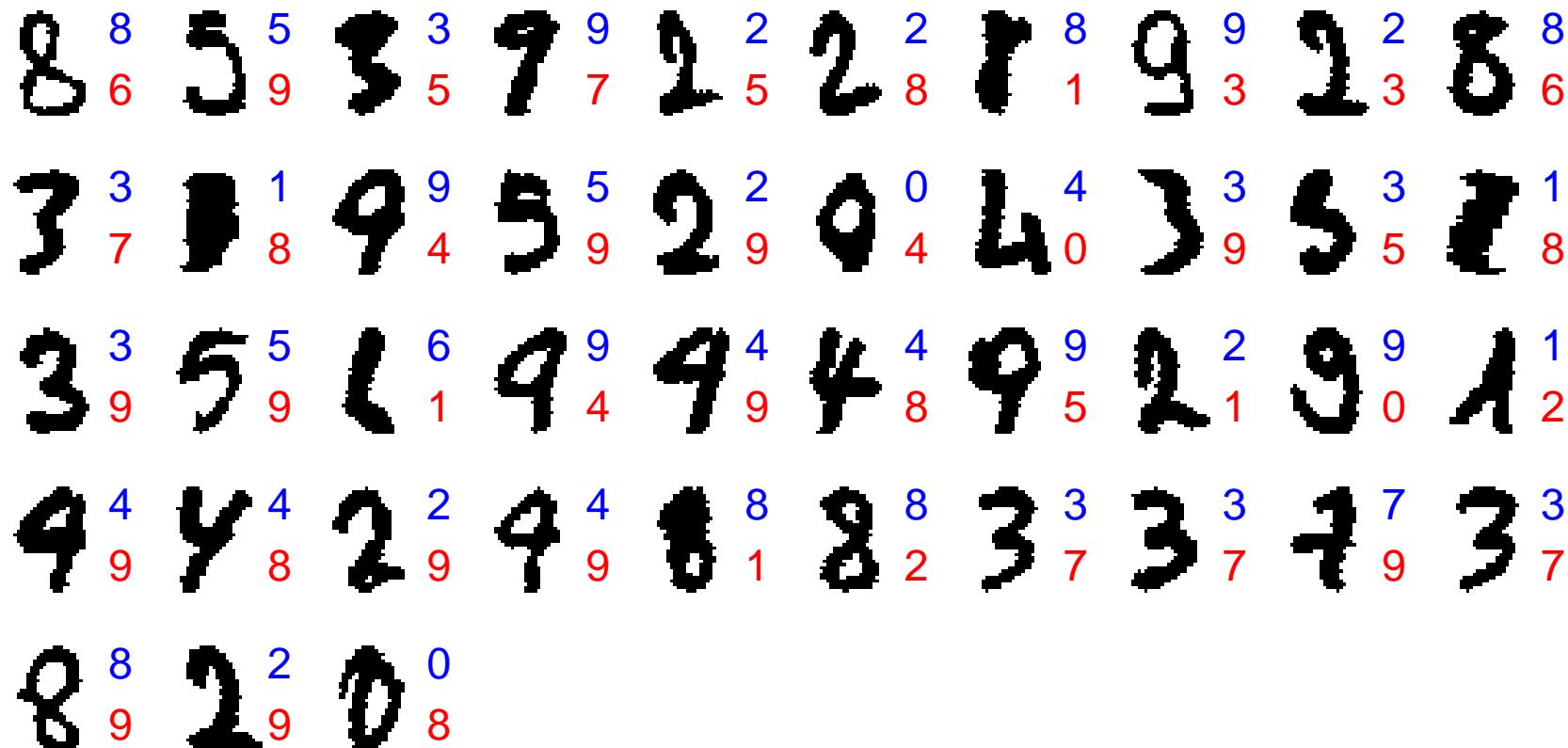
Все случаи неправильной классификации алгоритмом SVM.

Красная цифра — ответ классификатора, синяя — верный ответ.



Все случаи неправильной классификации алгоритмом AdaBoost ( $M = 40$ ).

Красная цифра — ответ классификатора, синяя — верный ответ.



Алгоритм AdaBoost и его варианты применяется с большим успехом в различных прикладных задачах.

Например, прорыв в решении задачи детектирования лица на изображении был осуществлен путем комбинирования каскадного алгоритма детектирования с AdaBoost [Viola, Jones, 2001].

## **Градиентный бустинг деревьев решений**

GBT – Gradient Boosting Trees [Friedman, 1999]

– другой подход к реализации идеи бустинга:

использование идей метода градиентного спуска в оптимизации

## 2.3.2. Баггинг

*Bagging* – от bootstrap aggregation [Breiman, 1994]

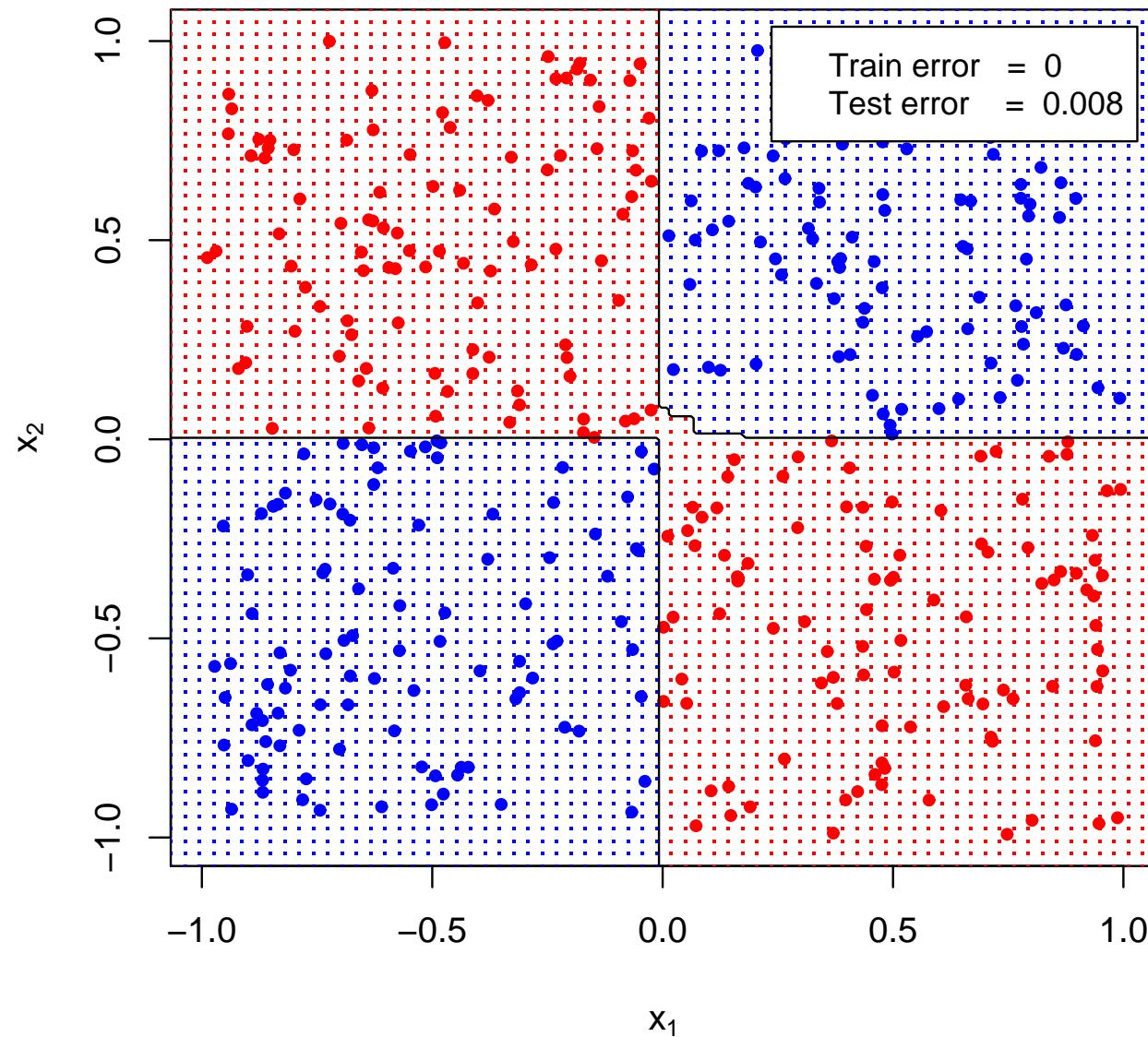
Классификатор  $f_m$  обучается на bootstrap–выборке ( $m = 1, 2, \dots, M$ ).

Финальный классификатор  $f$  – функция голосования:

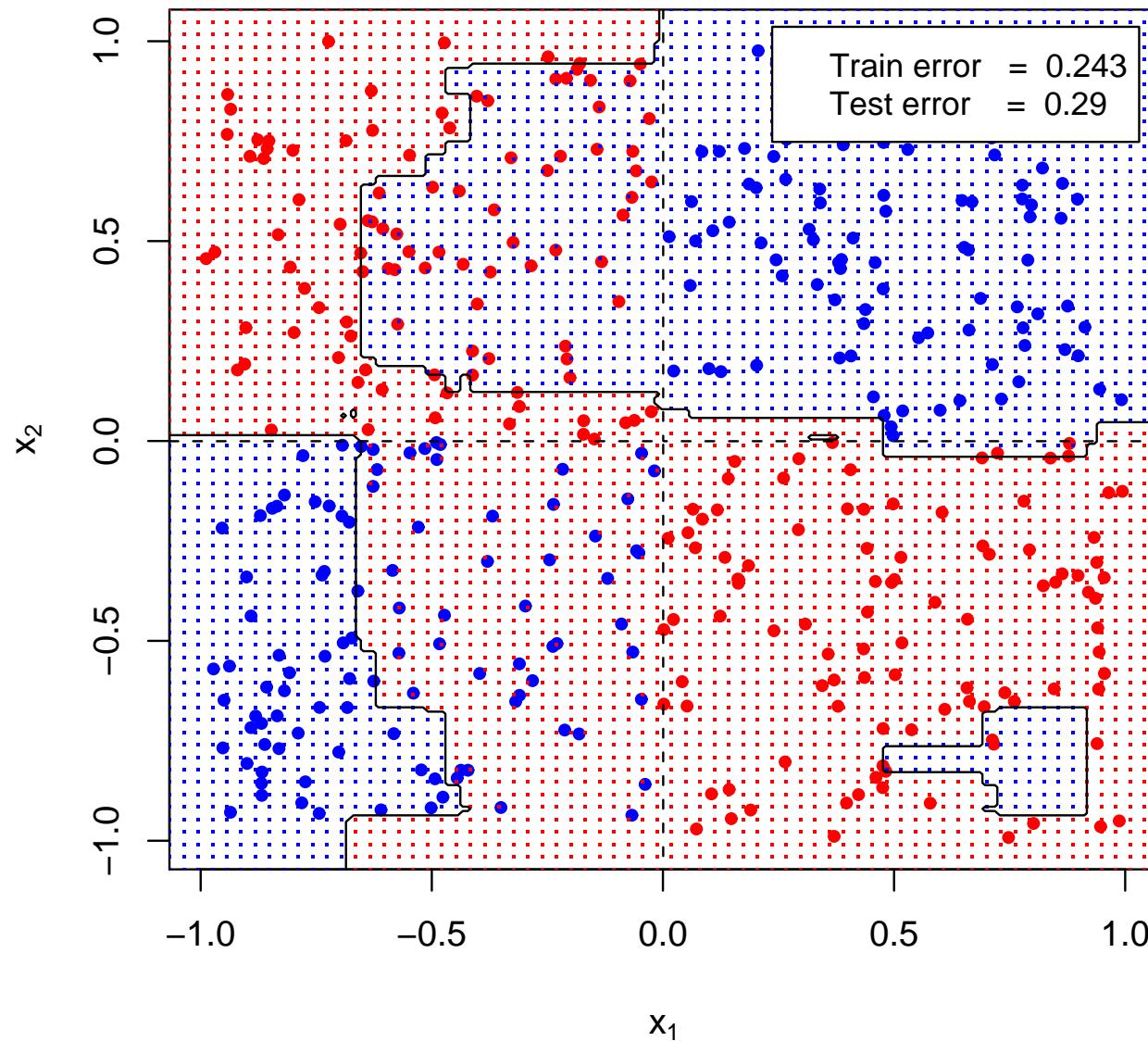
$$f(x) = \max_{k=1, \dots, K} \sum_{m=1}^M I(f_m(x) = k).$$

Позволяет бороться с неустойчивостью классификаторов  $f_m$ .

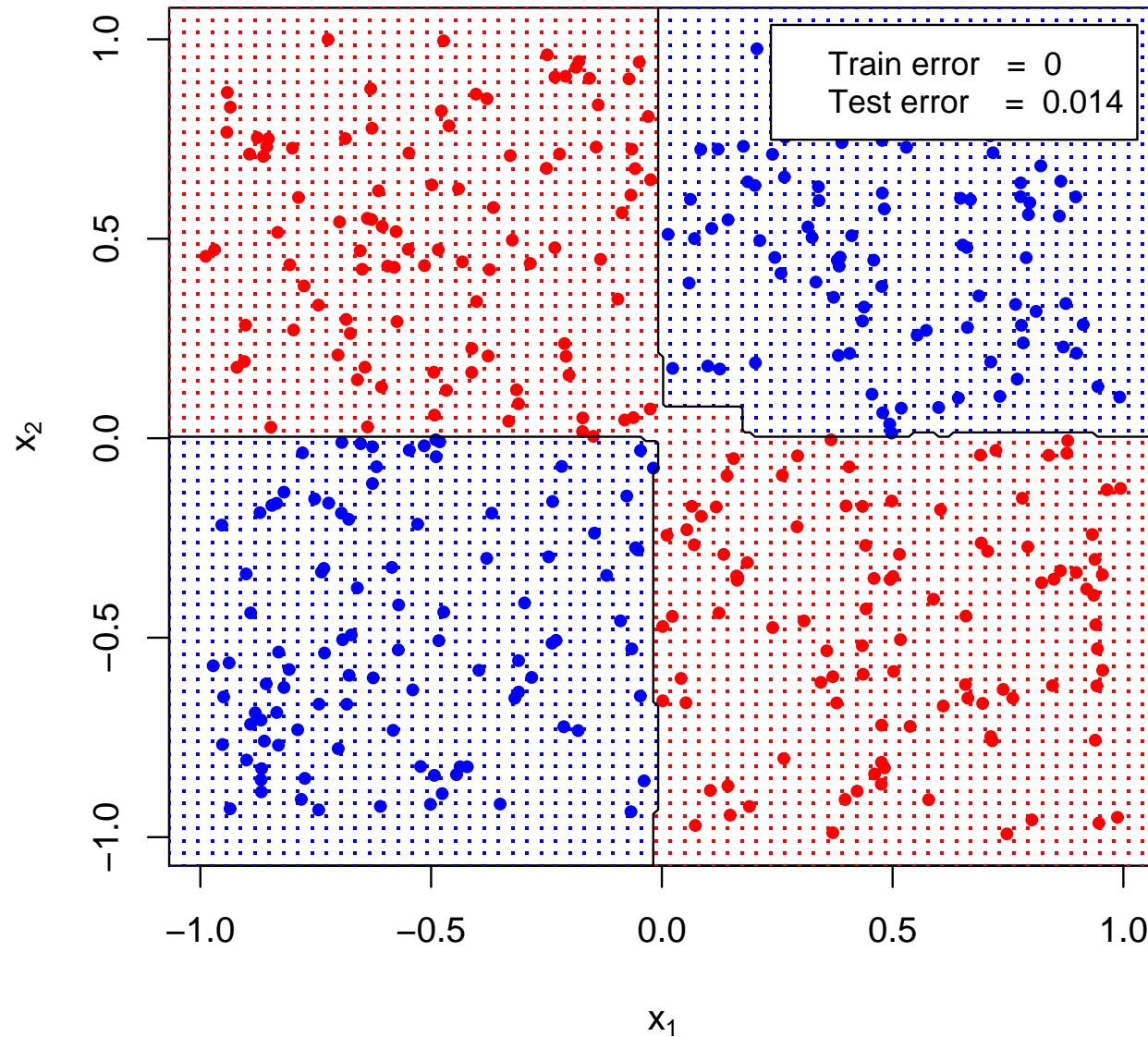
## AdaBoost — деревья решений высоты 2



## Bagging – деревья решений высоты 2



## Bagging – деревья решений высоты 3



## Случайный лес

Random forests [Breiman, 2001]

Ансамбль параллельно обучаемых независимых деревьев решений.

Независимое построение определенного количества деревьев:

Генерация случайной подвыборки из обучающей выборки (50–70% от размера всей обучающей выборки) и построение дерева решений по данной подвыборке (в каждом новом узле дерева переменная для разбиения выбирается не из всех признаков, а из случайно выбранного их подмножества небольшой мощности).

## 2.4. Эксперименты

Данные: UCI Machine Learning Repository <http://archive.ics.uci.edu/ml/>

Software: OpenCV Library <http://opencv.willowgarage.com>

(Реализация GBT – П.Н.Дружков, ННГУ)

Эксперимент – П.Н. Дружков

Задачи классификации

10-CV ошибка

Задача	$N$	$d$ (колич.+ном.)	$K$	GBT	DTree	RF	ExtRF	SVM
Agaricus lepiota	8124	22 (0 + 22)	2	0	0.00	0	0	0
Liver disorders	345	6 (6 + 0)	2	0.25	0.31	0.22	0.25	0.28
Car evaluation	1728	6 (0 + 6)	4	0	0.051	0.036	0.039	0.050

GBT – Gradient Boosting Trees – градиентный бустинг деревьев решений,

DTree – Decision Tree – деревья решений,

RF – Random Forests – случайные леса,

ExtRF – Extremely Random Forests – экстремально случайные леса,

SVM – Support Vector Machine – машина опорных векторов

## Задачи восстановления регрессии. Средняя абсолютная 10-CV ошибка

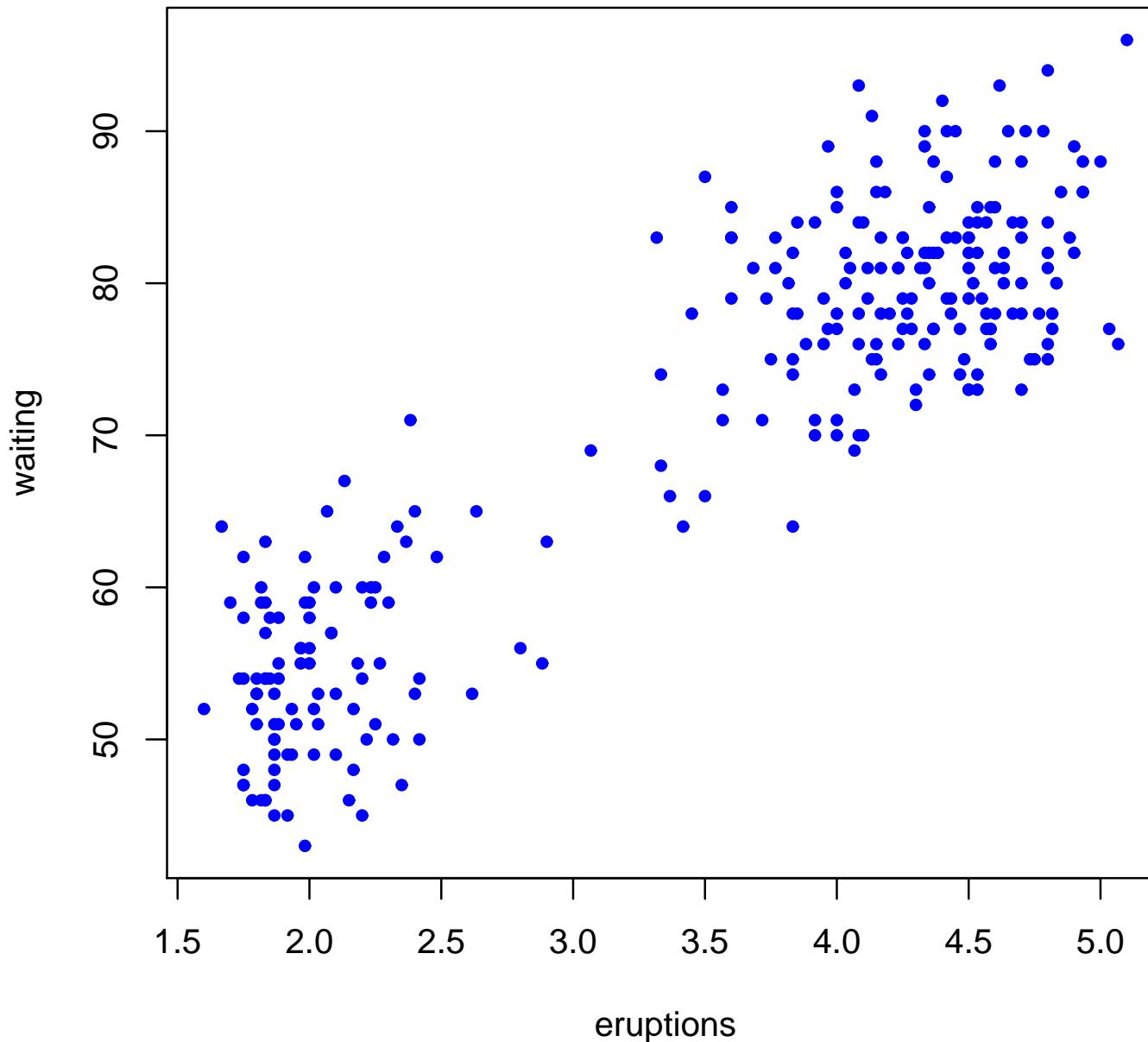
<i>Задача</i>	<i>N</i>	<i>d</i> (колич.+ном.)	GBT	DTree	RF	ExtRF	SVM
Auto-mpg	398	7 (4 + 3)	2.00	2.24	<b>1.88</b>	2.15	2.98
Computer hardware	209	8 (7 + 1)	12.61	15.62	11.62	<b>9.63</b>	37.00
Concrete slump	103	9 (9 + 0)	2.26	2.92	2.60	2.36	<b>1.77</b>
Forestfires	517	12 (10 + 2)	18.74	17.26	17.79	<b>16.64</b>	<b>12.90</b>
Boston housing	506	13 (13 + 0)	<b>2.03</b>	2.60	2.13	2.20	4.05
Import-85	201	25 (14 + 11)	1305	1649	<b>1290</b>	1487	1787
Servo	167	4 (0 + 4)	<b>0.238</b>	0.258	0.247	0.420	0.655
Abalone	4177	8 (7 + 1)	<b>1.470</b>	1.603	1.492	1.498	2.091

### **3. Обучение без учителя: кластеризация**

#### **Пример 1. Извержения гейзера**

Рассмотрим данные о времени между извержениями и длительностью извержения гейзера Old Faithful geyser in Yellowstone National Park, Wyoming, USA (*A. Azzalini, A.W.Bowman A look at some data on the Old Faithful geyser // Applied Statistics. 1990, 39. P. 357–365.*)

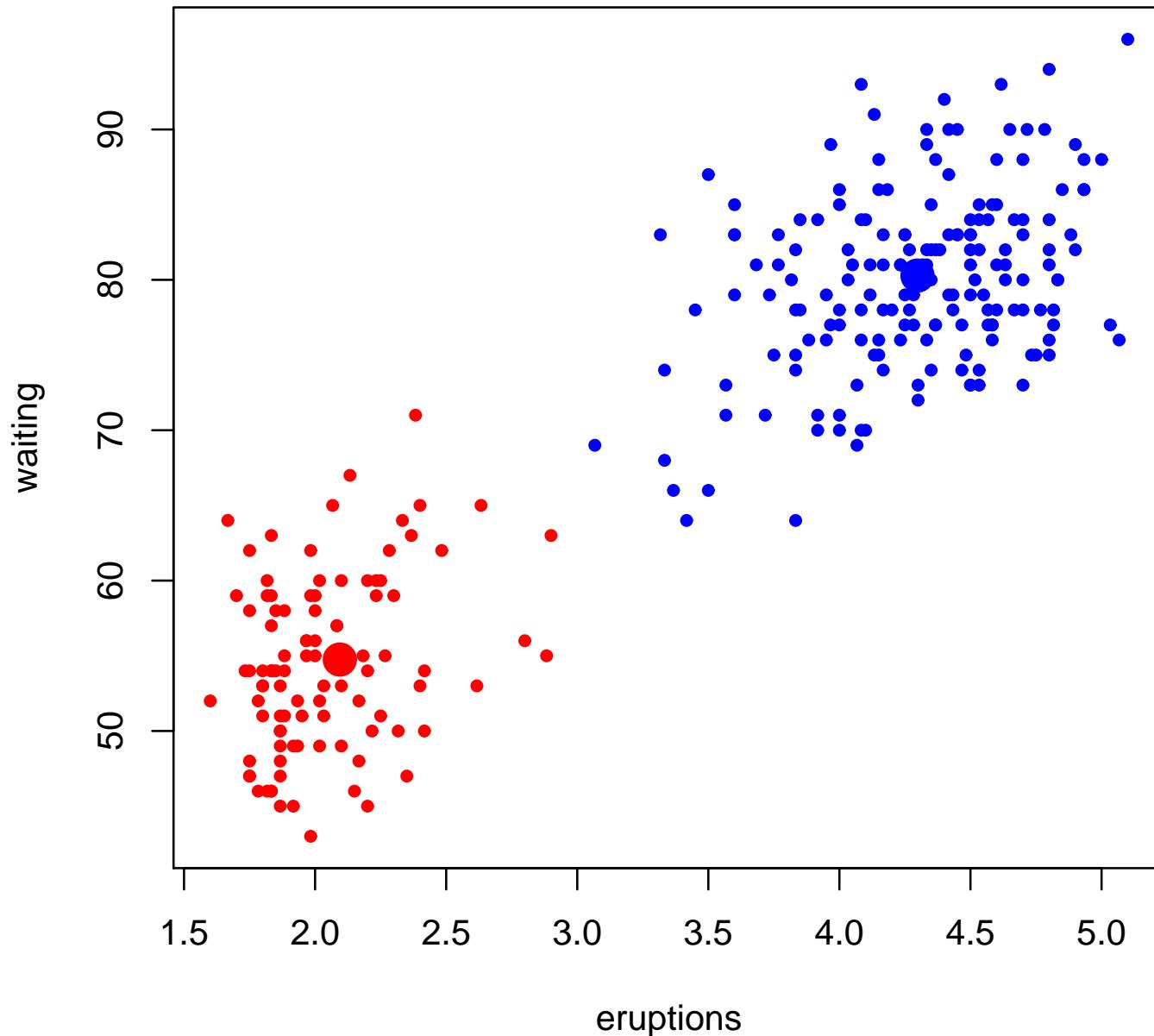
Диаграмма, представляющая данные о времени извержения и промежутках между извержениями гейзера.



Мы видим, что точки группируются в два кластера.

В одном кластере находятся точки, соответствующие извержениям с малой длительностью и малым временем ожидания.

В другом — с большой длительностью и большим временем ожидания.



## **Пример 2. Анализ данных, полученных с биочипов**

*Биочип, или микроэррэй, (biochip, microarray) — это миниатюрный прибор, измеряющий уровень экспрессии генов в имеющемся материале.*

*Экспрессия — это процесс перезаписи информации с гена на РНК, а затем на белок.*

*Количество и даже свойства получаемого белка зависят не только от гена, но также и от различных внешних факторов (например, от введенного лекарства).*

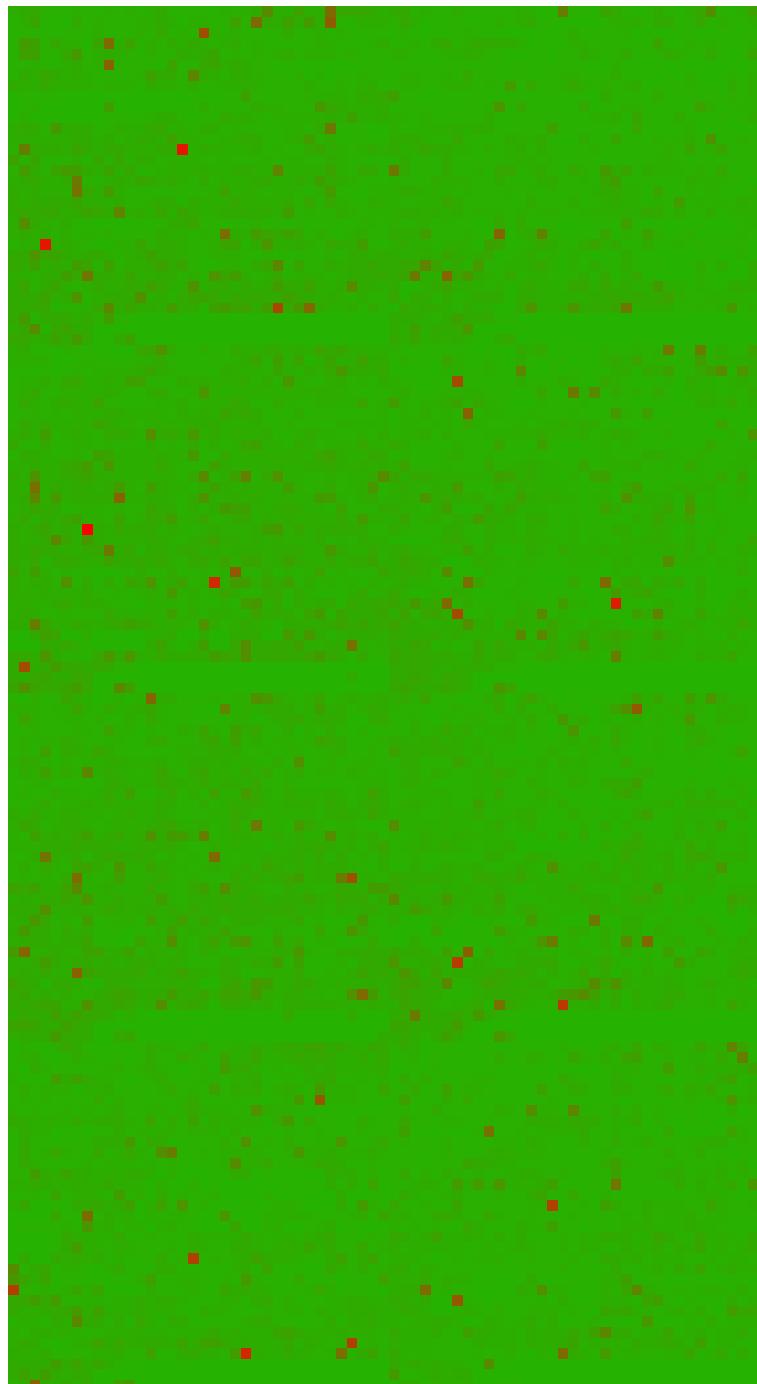
*Таким образом, уровень экспрессии — это мера количества генерируемого белка (и скорости его генерирования).*

*На биочип кроме исследуемого материала помещается также «контрольный» генетический материал.*

*Положительные значения (красный цвет) — увеличение уровня экспрессии по сравнению с контрольным.*

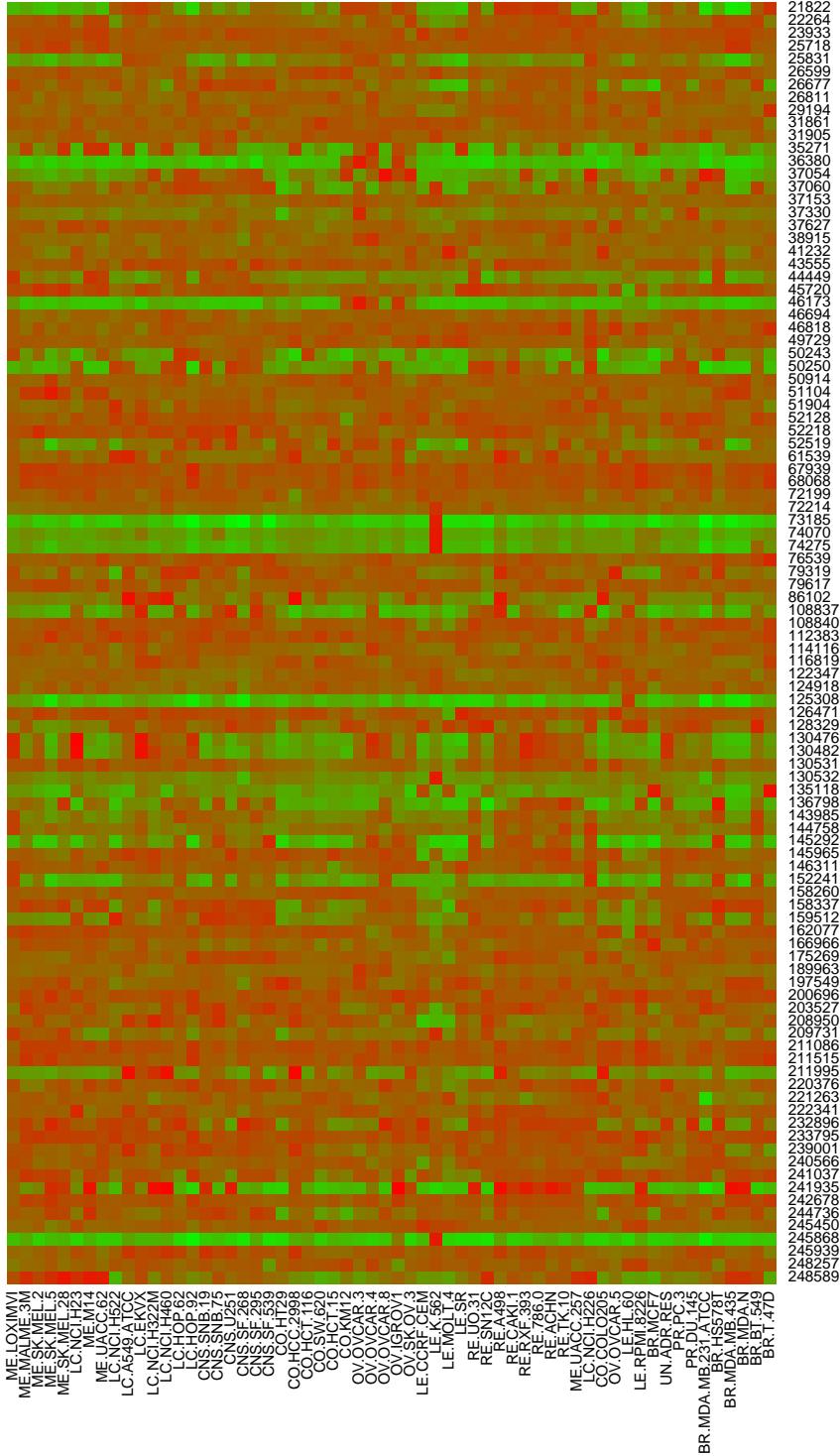
*Отрицательные значения (зеленый цвет) — уменьшение.*

Условное изображение биочипа. Каждая точка на рисунке соответствует определенному гену. Всего анализируется  $132 \times 72 = 9504$  гена. Brown, V.M., Ossadtchi, A., Khan, A.H., Yee, S., Lacan, G., Melega, W.P., Cherry, S.R., Leahy, R.M., and Smith, D.J.; Multiplex three dimensional brain gene expression mapping in a mouse model of Parkinson's disease; Genome Research 12(6): 868-884 (2002).



Данные для 60 экспериментов с биочипом

<http://discover.nci.nih.gov/datasetsNature2000.jsp> Строки соответствуют генам, столбцы — экспериментам. Приведены только первые 100 строк (из общего числа 1375). Строки, содержащие отсутствующие значения, исключены.



Поставим следующие задачи:

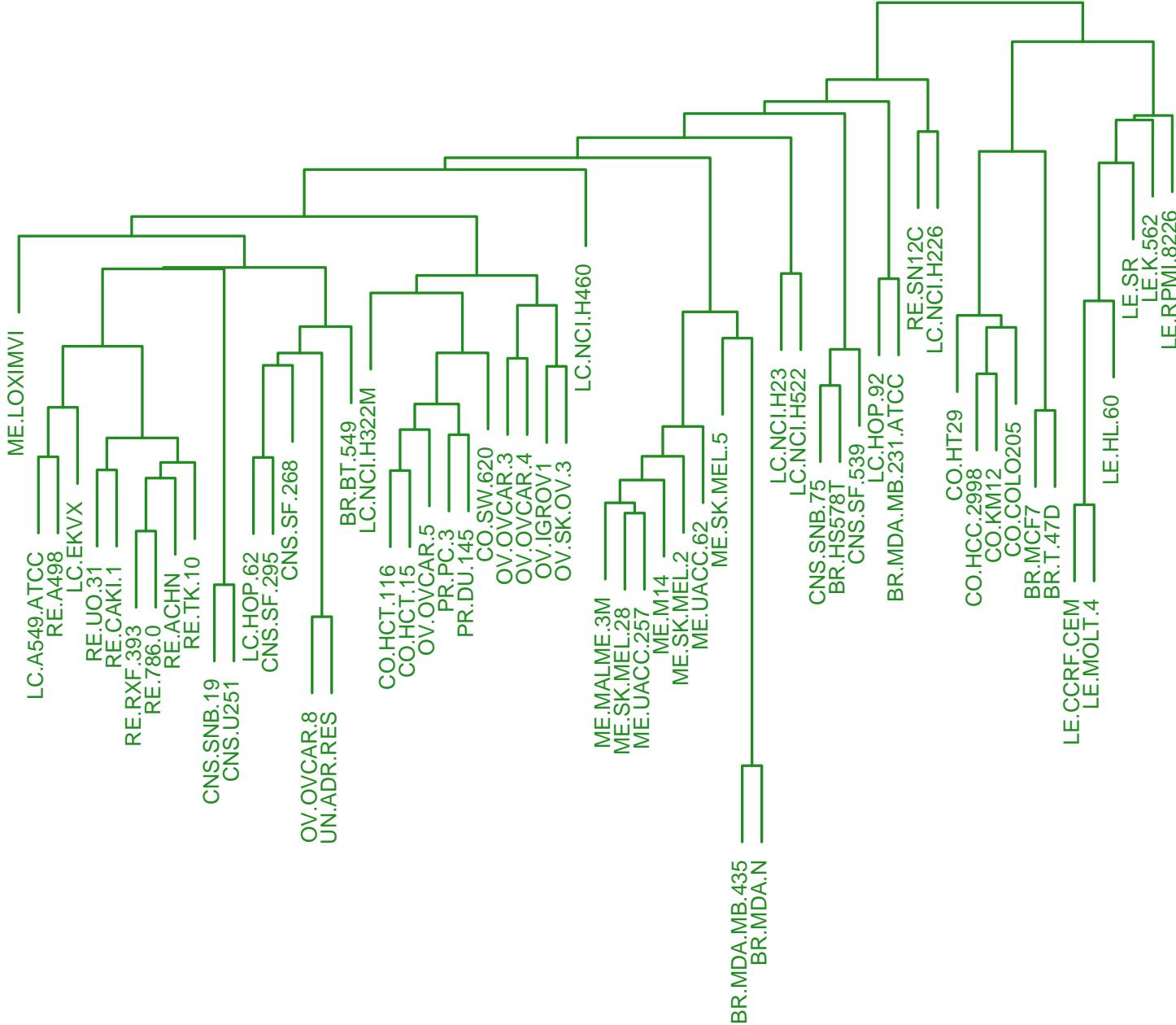
- (а) Найти гены, показавшие высокую экспрессию, в заданных экспериментах.  
т.е. найти наиболее красные клетки в заданных столбцах.
- (б) Разбить гены на группы в зависимости от влияния на них экспериментов. Гены, реагирующие «почти одинаковым» образом в «большом» числе экспериментов, должны попасть в одну группу. Гены, реагирующие по-разному, должны находиться в разных группах.  
т.е. разбить строки на группы (кластеры) «похожих» между собой строк
- (в) Разбить эксперименты на группы в зависимости от их влияния на гены. Эксперименты, в которых одинаковые гены реагировали «сходным» образом должны оказаться в одной группе. Эксперименты, в которых гены реагировали «различно», должны находиться в разных группах.  
т.е. разбить столбцы на группы (кластеры) «похожих» между собой строк

Задачи (б) и (в) — это задачи кластерного анализа.

Таксономия 60 клеток на основе анализа уровня экспрессии их генов.

<http://discover.nci.nih.gov/datasetsNature2000.jsp>

60 прецедентов, 1375 признаков

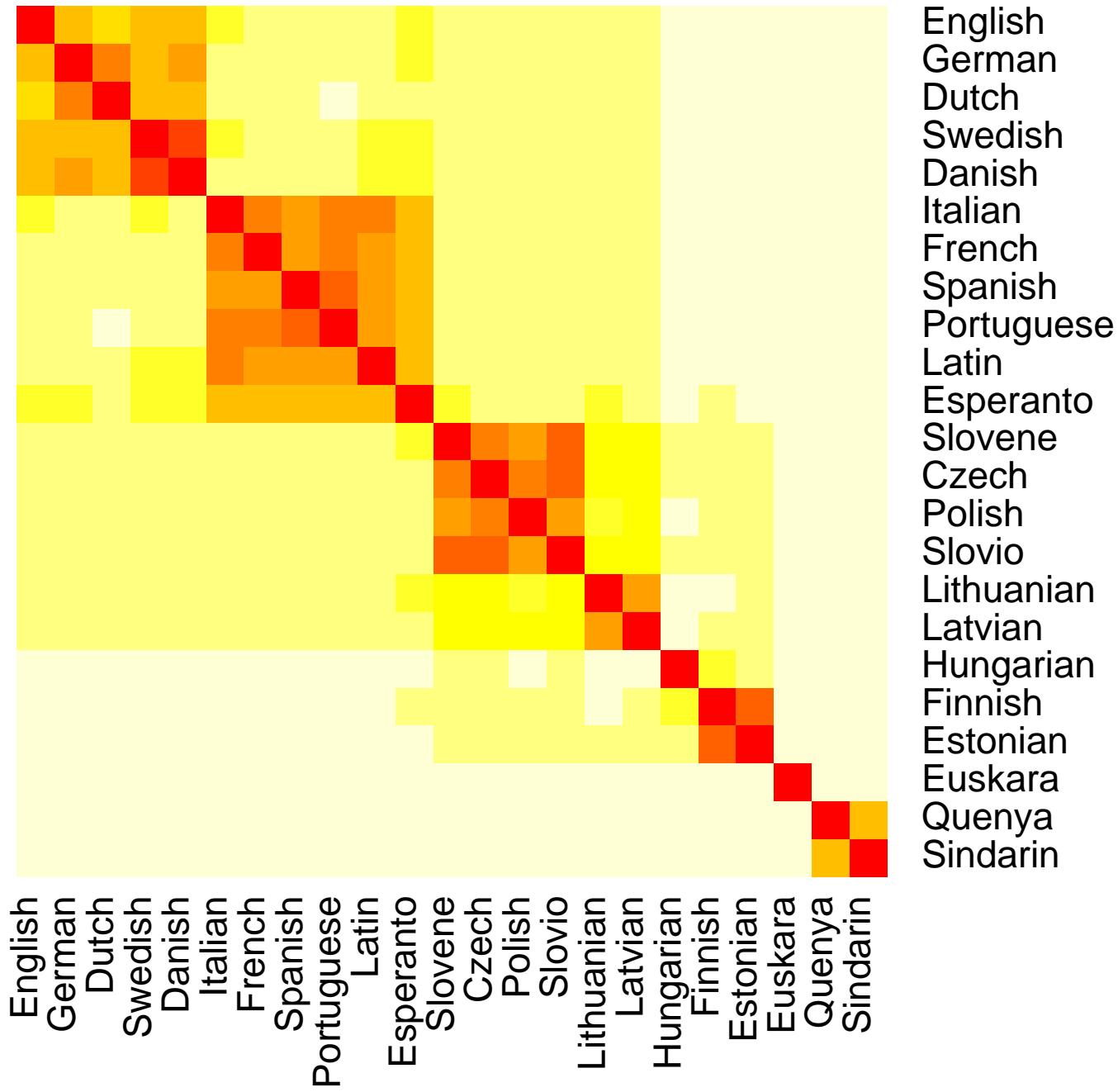


## **Пример 3. Списки Сводеша и таксономия языков**

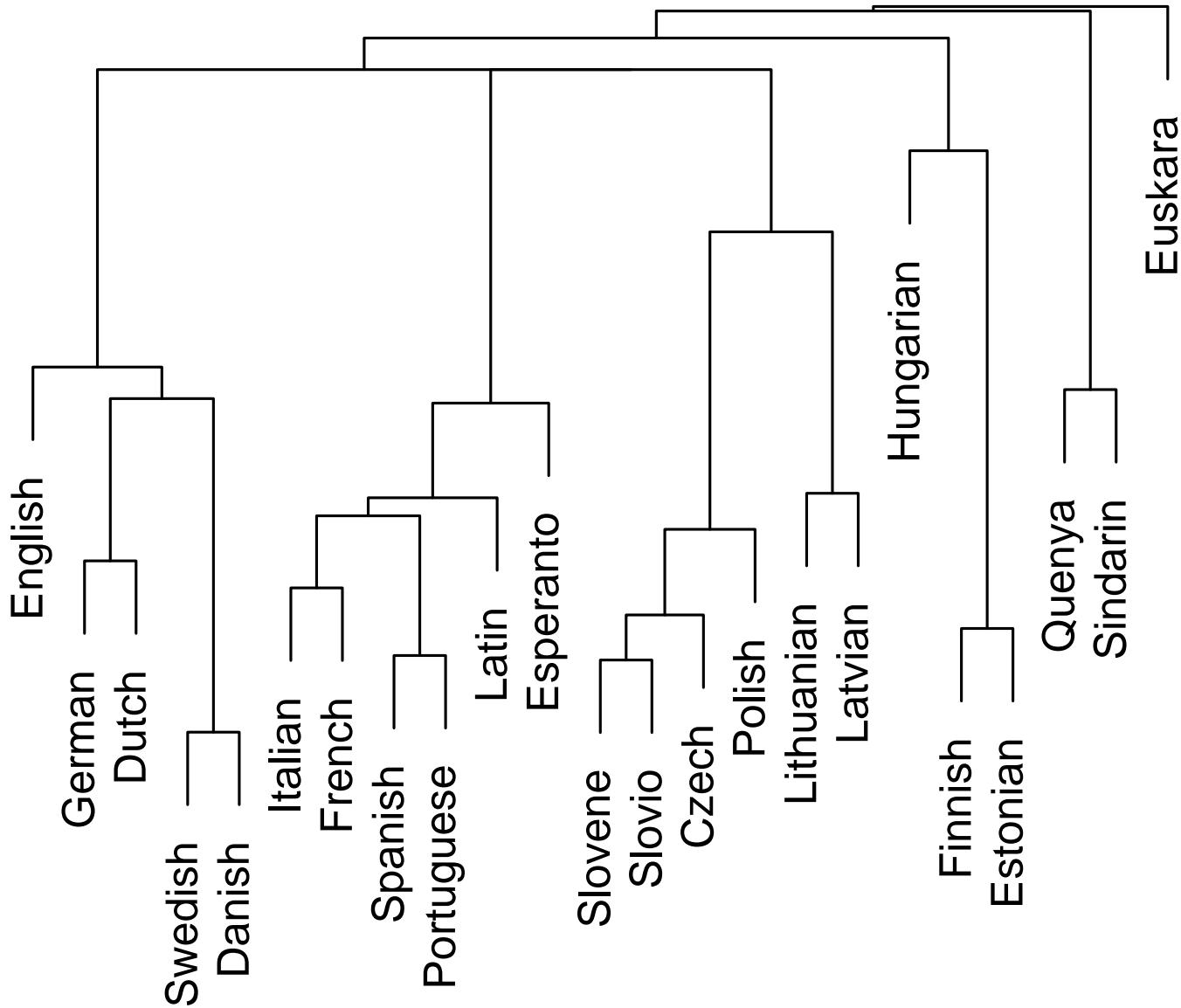
Список Сводеша (Swadesh) — список из 207 слов «базового» словаря (местоимения, числительные 1–5, глаголы, обозначающие простые действия и т. п.)

№	Русский	Английский	Немецкий	Итальянский	Французский	Чешский
1	я	I	ich	io	je	já
2	ты	you	du	tu	tu	ty
3	он	he	er	lui	il	on
4	мы	we	wir	noi	nous	my
5	вы	you	ihr	voi	vous	vy
6	они	they	sie	loro	ils	oni
7	этот	this	dieses	questo	ceci	tento
8	тот	that	jenes	quello	cela	tamten
9	здесь	here	hier	qui	ici	zde
10	там	there	dort	lá	lá	tam
11	кто	who	wer	chi	qui	kdo
12	что	what	was	che	quoi	co
13	где	where	wo	dove	où	kde
14	когда	when	wann	quando	quand	kdy
15	как	how	wie	come	comment	jak
16	не	not	nicht	non	ne... pas	ne
.....						
205	если	if	wenn	se	si	jestlize
206	потому что	because	weil	perché	parce que	protoze
207	имя	name	Name	nome	nom	jméno

Матрица сходства между некоторыми языками, построенная на основе списков Сводеша.



Дерево иерархической кластеризации для 23 языков, построенное на основе списков Сводеша.



## 4. Выводы

Что мы узнали?

- Узнали, что машинное обучение — это круто
- Узнали, что такое обучение с учителем (задача классификации и задача восстановления регрессии) и обучение без учителя (кластеризация)
- Познакомились с идеей метода SVM (машина опорных векторов)
- Узнали, что такое деревья решений
- Познакомились с двумя принципами комбинирования слабых классификаторов (бустинг и баггинг)
- Узнали, что такое переобучение; осознали необходимость борьбы с ним
- Чтобы заниматься машинным обучением, надо хорошо знать математику

КОНЕЦ