Введение в машинное обучение и анализ данных

Фонарев Александр

http://newo.su

Спецкурс ЛКШ 2012
Сборка от 09.08.2012

Спасибо за помощь

Потапенко Анне, Ромову Петру, Евдокимовой Валерии, Иванову Олегу, Борисову Михаилу

Лекция 1

Часть 1

Чем занимаются в анализе данных
и машинном обучении?

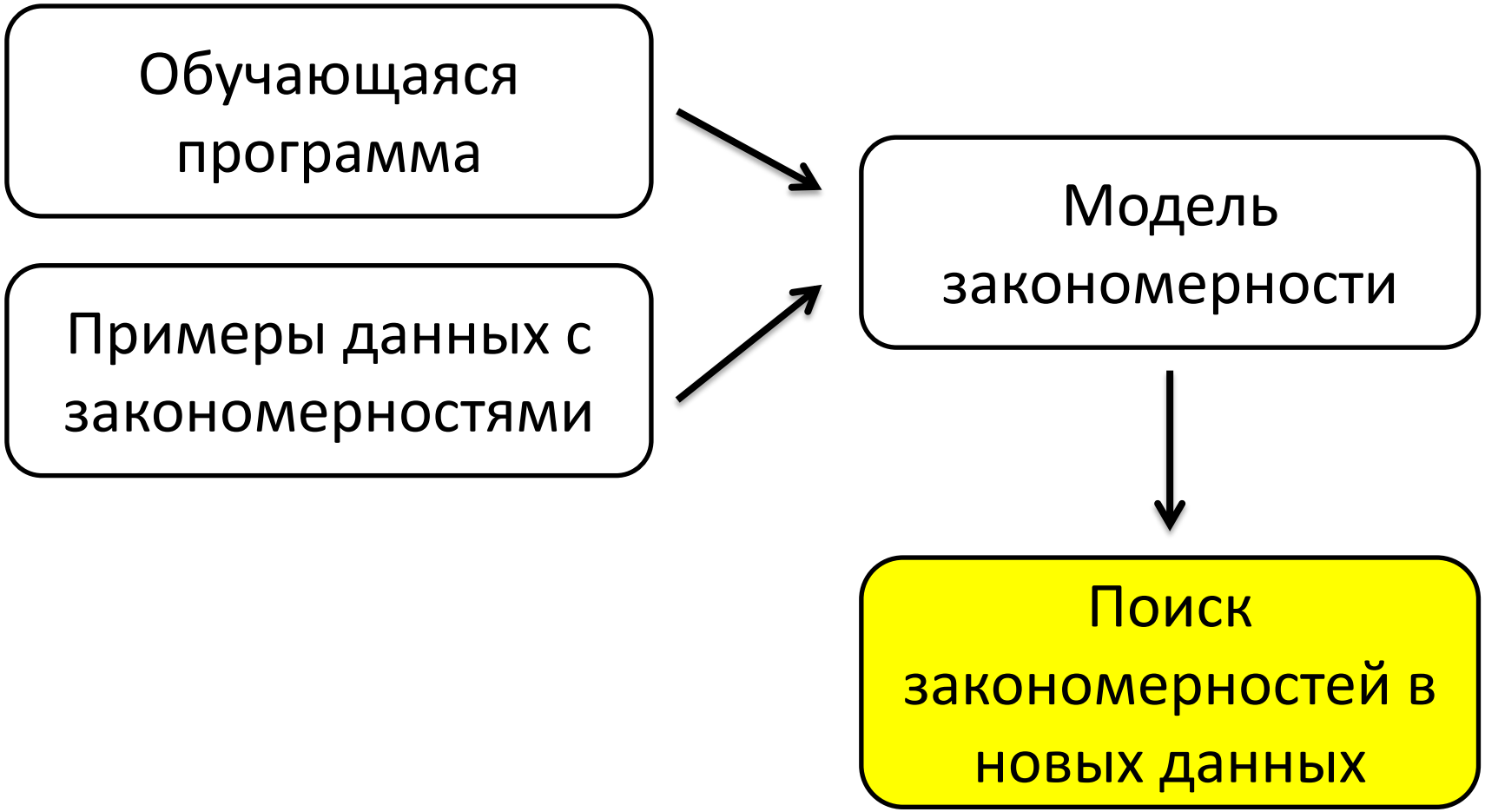
Что такое «Анализ Данных»?

- Сложно устроенные данные
- Большие объемы данных
- Надо найти или проверить закономерности в данных
- А что такое закономерность и как их искать?

Закономерности в данных

- Поиск подстроки в строке – тоже поиск закономерности
- Что делать, если мы не умеем хорошо описать закономерность?
- Почему человек понимает закономерность?

Суть машинного обучения

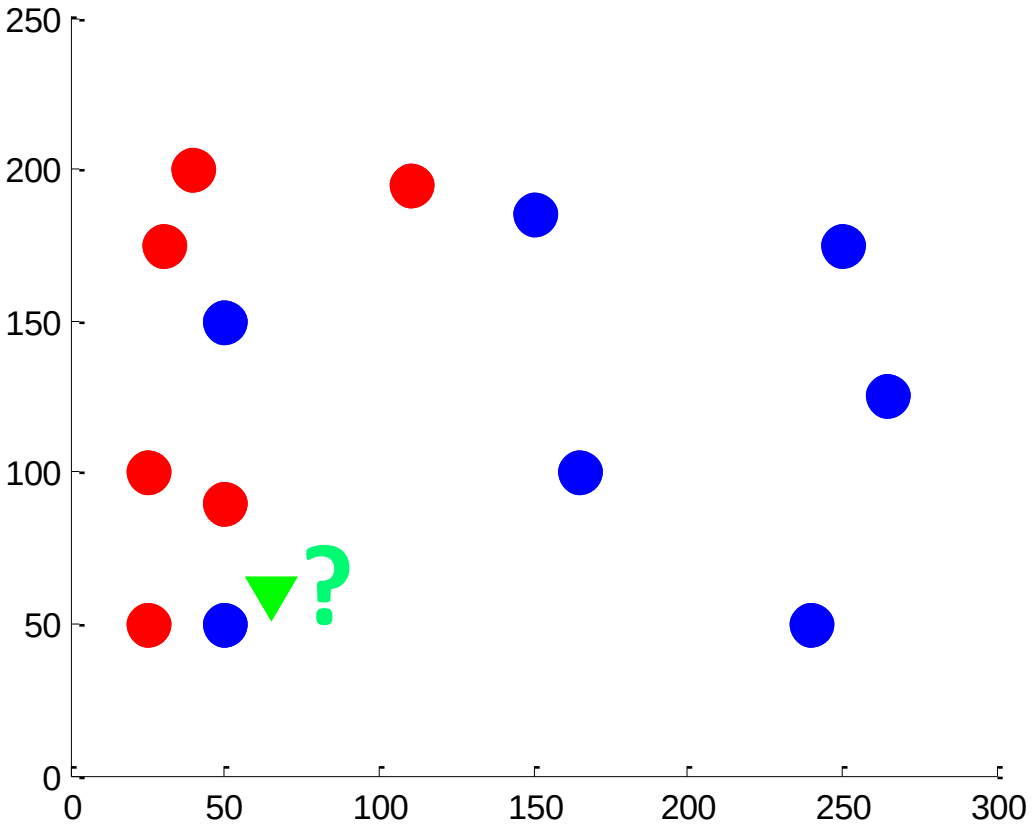


Часть 2

Простейшие задачи классификации

Простая задача

- Синий или красный новый объект?

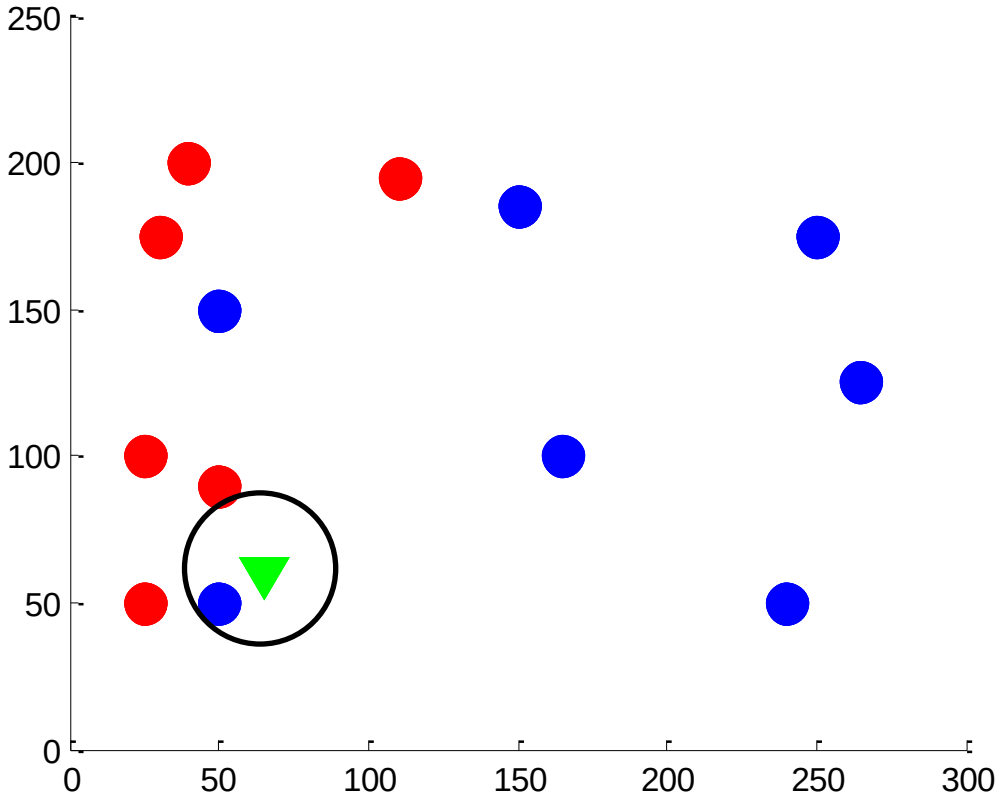


Как решать?

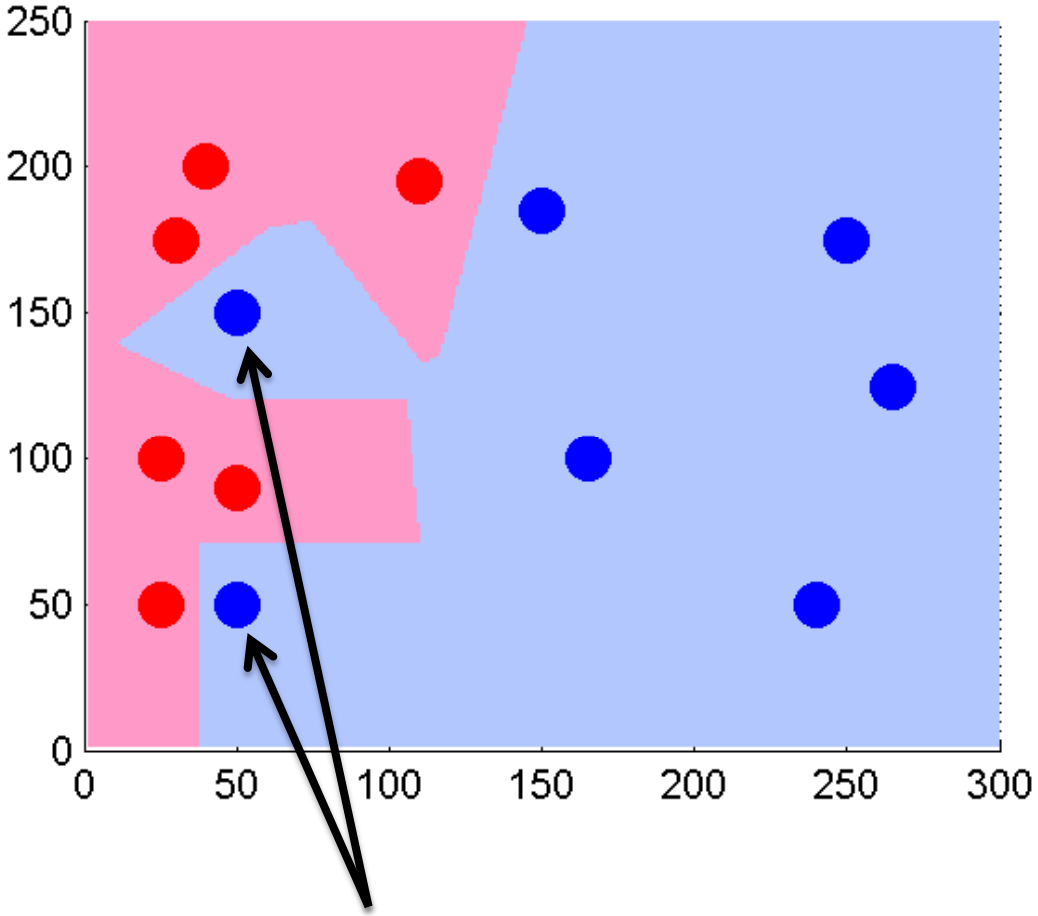
- Точного и правильного решения здесь нет
- Пытаемся решить логично с интуитивной точки зрения

Ближайший сосед

- Пусть новый объект принадлежит к тому же классу, что и его ближайший сосед



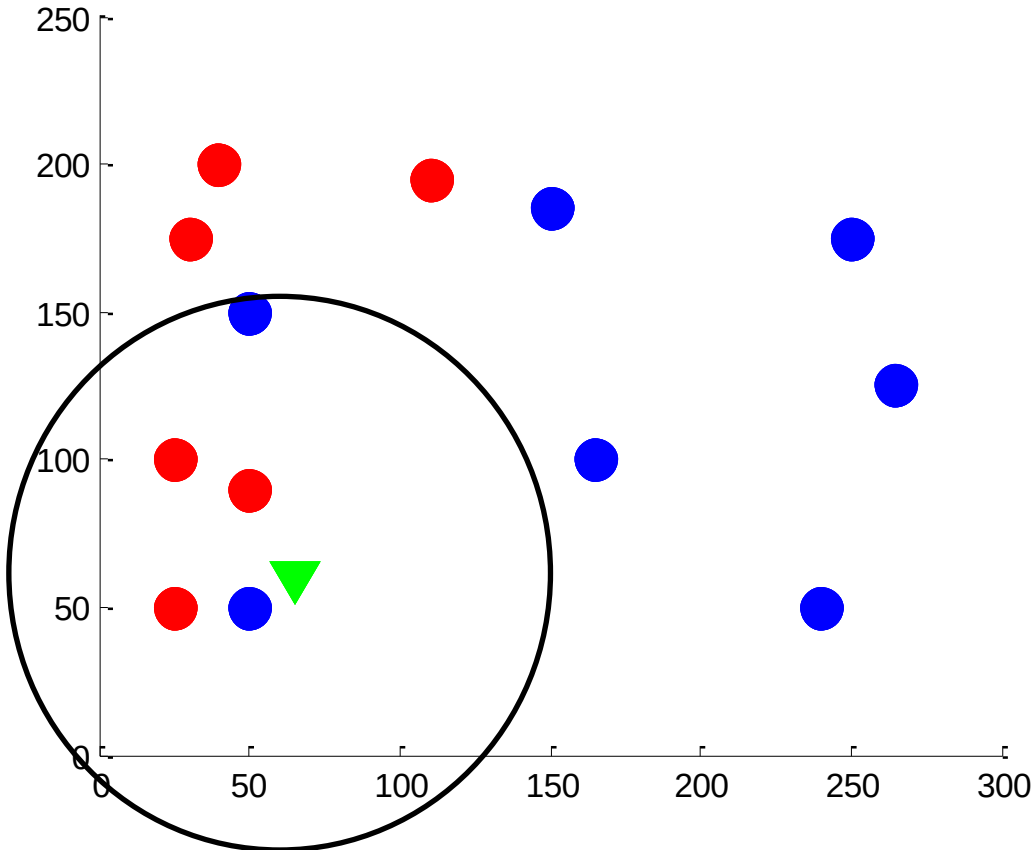
Граница разделения классов



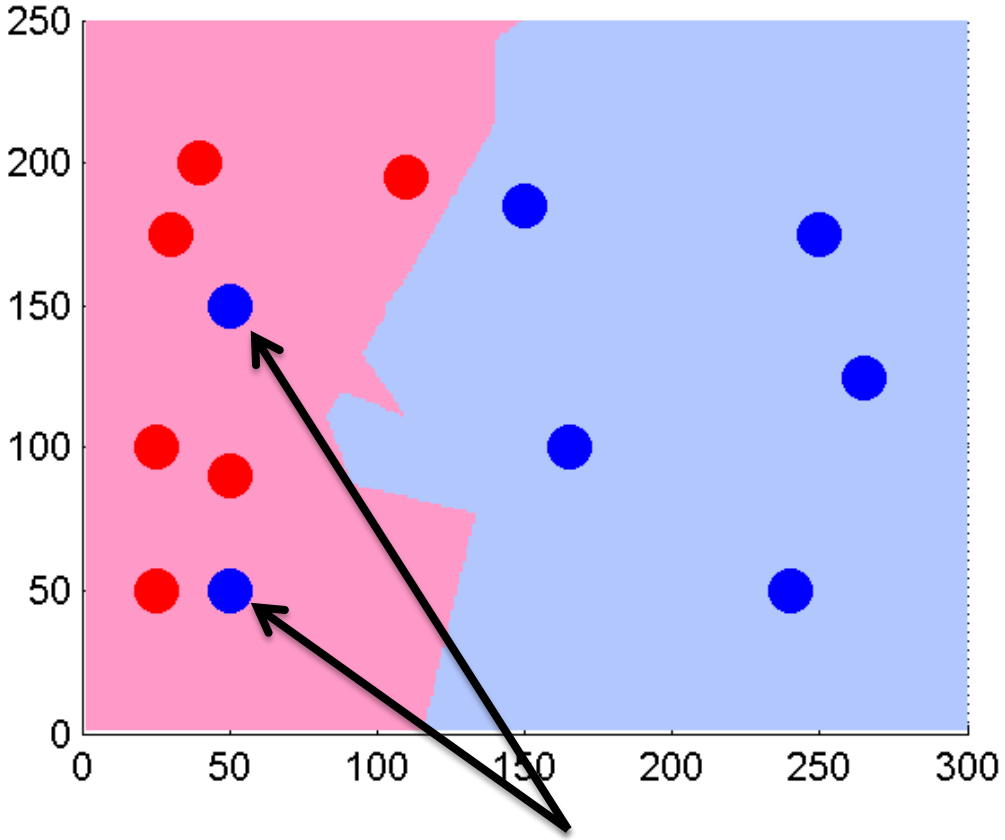
Возможно, шумовые объекты

Несколько ближайших соседей

- Новый объект принадлежит тому же классу, что и большинство из k его соседей

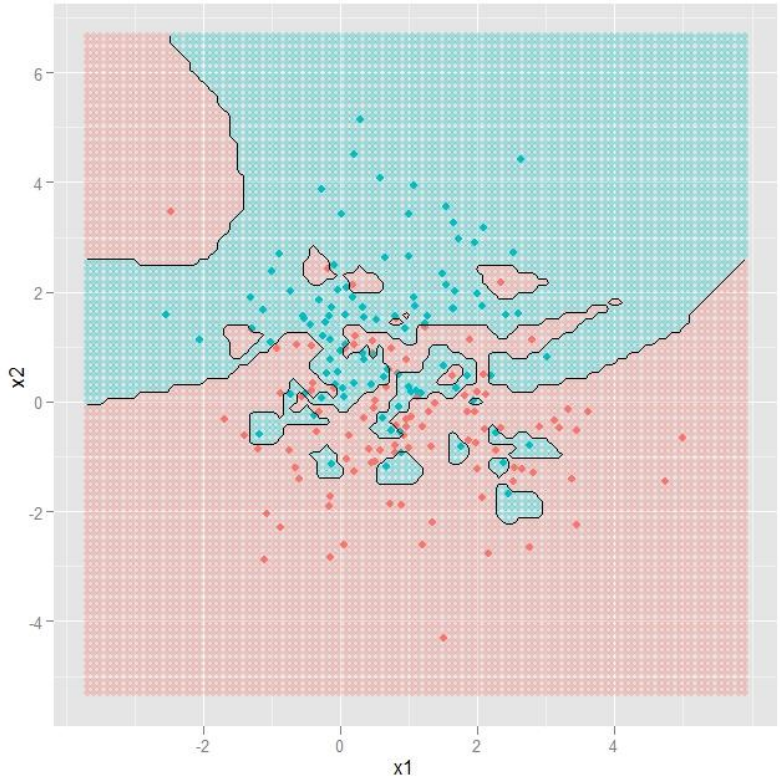


Граница разделения классов для k=5

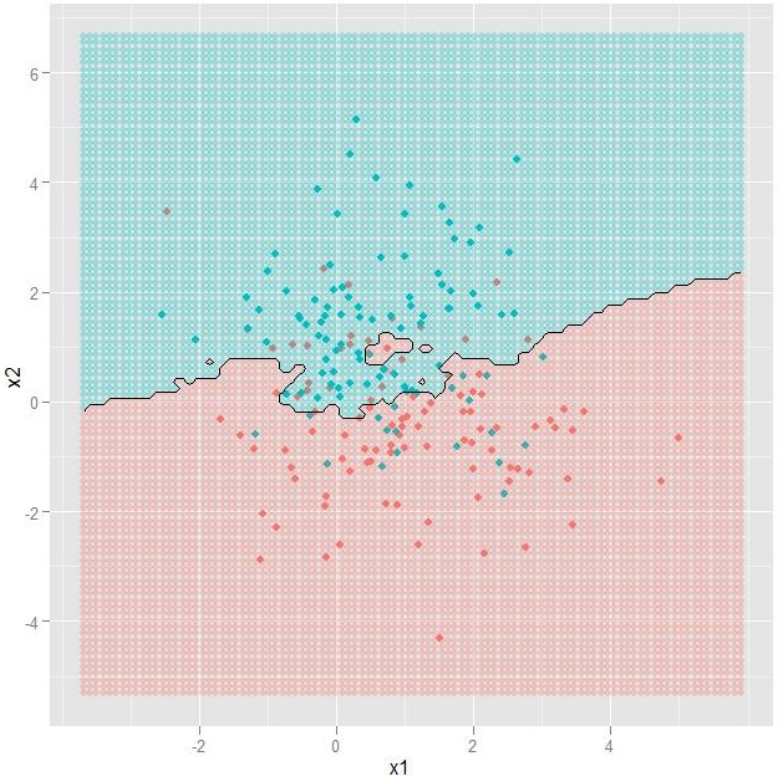


Оказывается, алгоритм дает ошибку на обучающей выборке! А это и не плохо.

А если объектов больше?



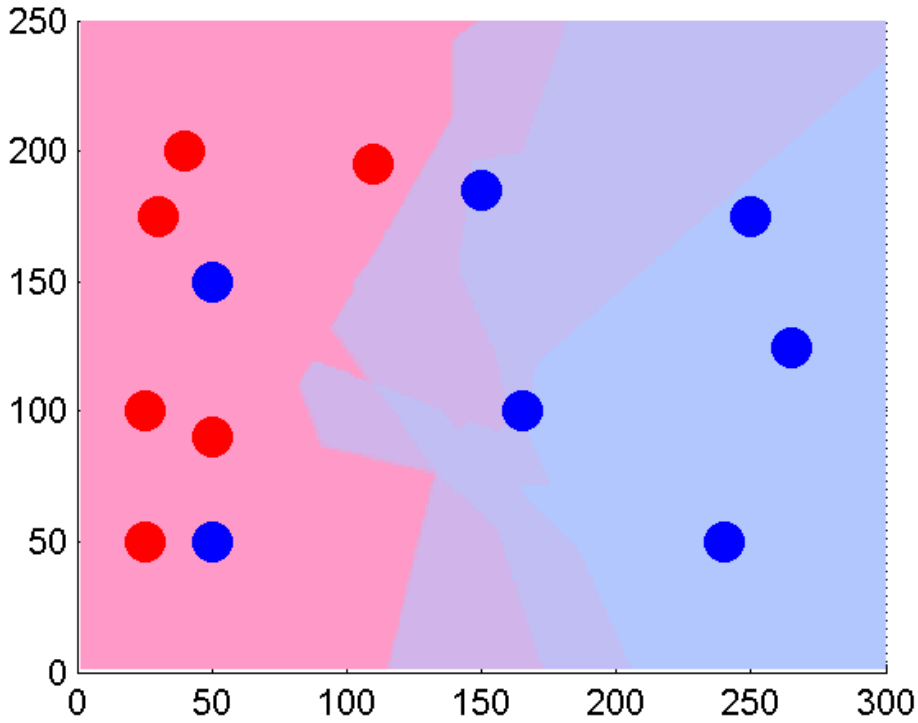
K=1



K=15

Нечеткая граница для k=5

- Полутона означают, что примерно половина соседей одного класса и половина другого



Часть 3

Основные понятия машинного обучения

Формальная постановка задачи

Какой алгоритм выбрать?

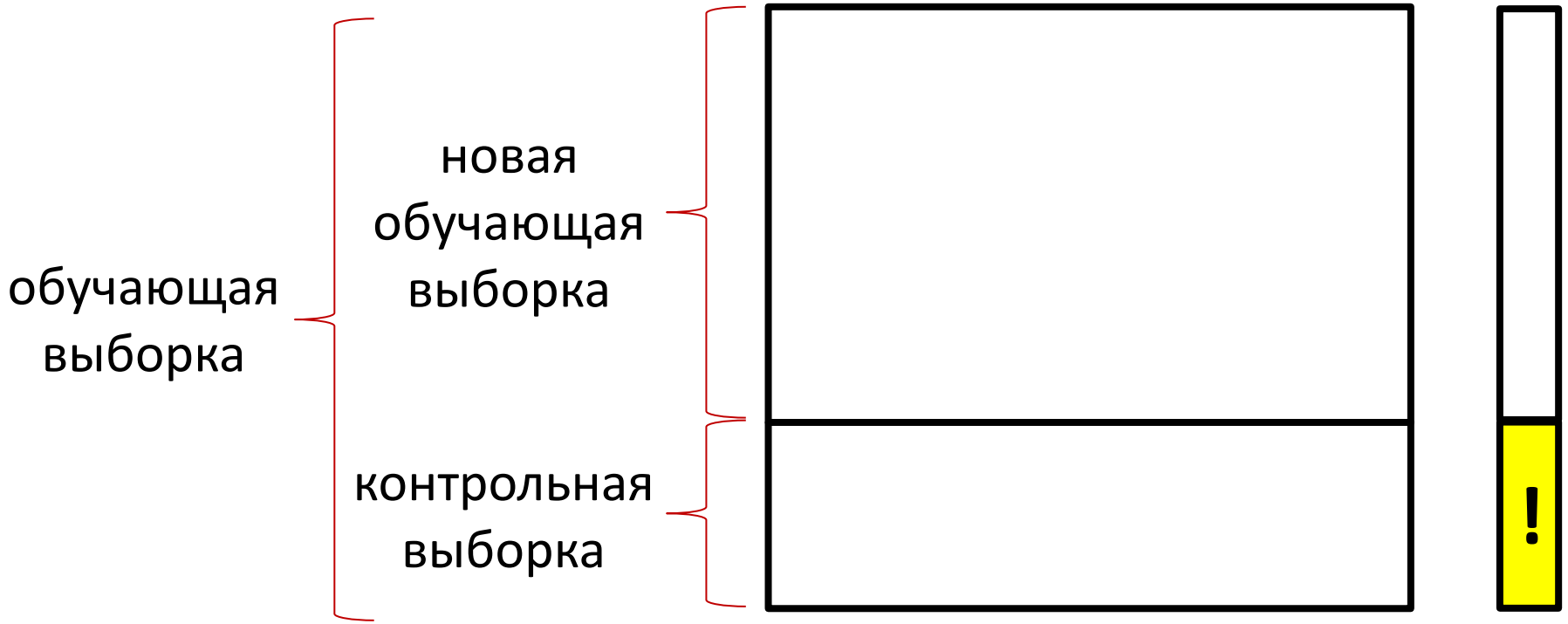
- Можно придумать много разных алгоритмов
- Качество – это доля правильных ответов на новых данных, т.е.

$$\frac{\text{количество объектов с правильными ответами}}{\text{количество новых объектов}}$$

- Как понять, сильно ли он ошибается, если нам не известны правильные ответы новых объектов?

Разбиение на контроль

- Используем имеющиеся данные из обучающей выборки. Разобъем обучение на две части.
- На одной мы будем обучаться, а на второй проверять, сколько ошибок выдал алгоритм

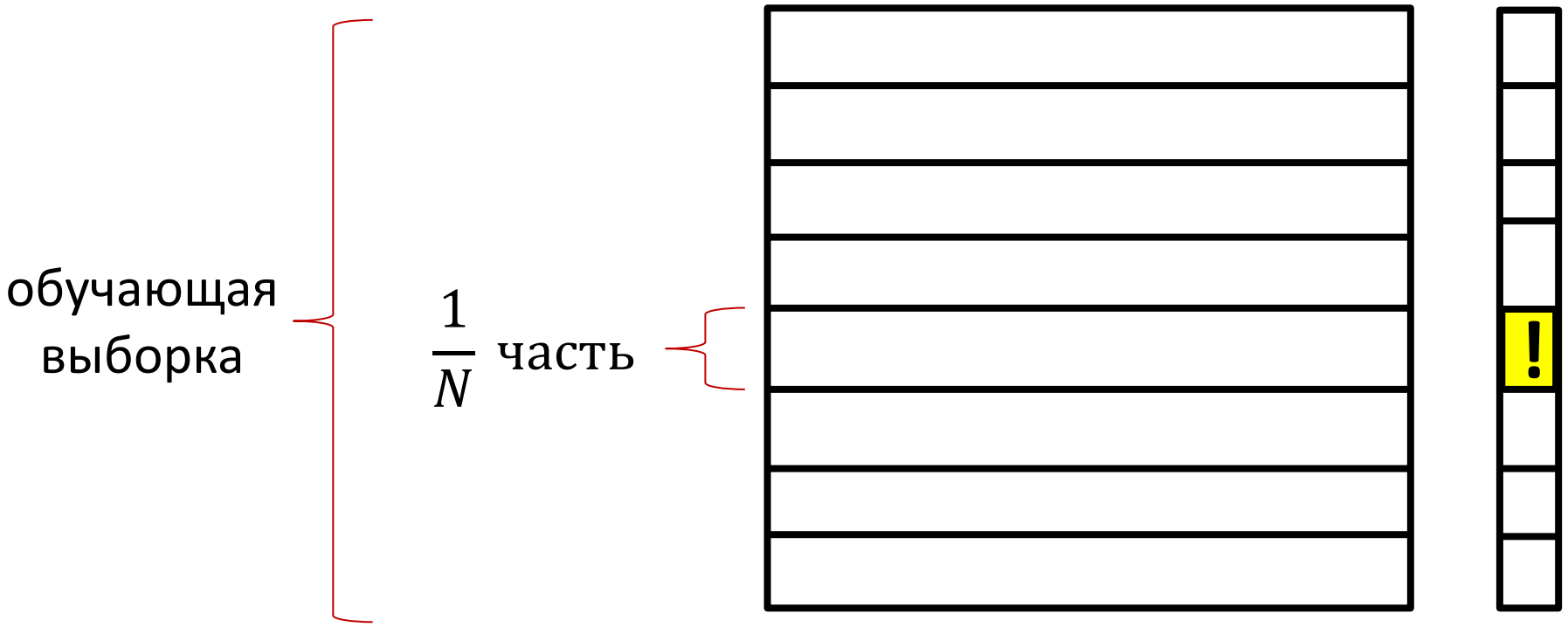


Недостатки разбиения на контроль

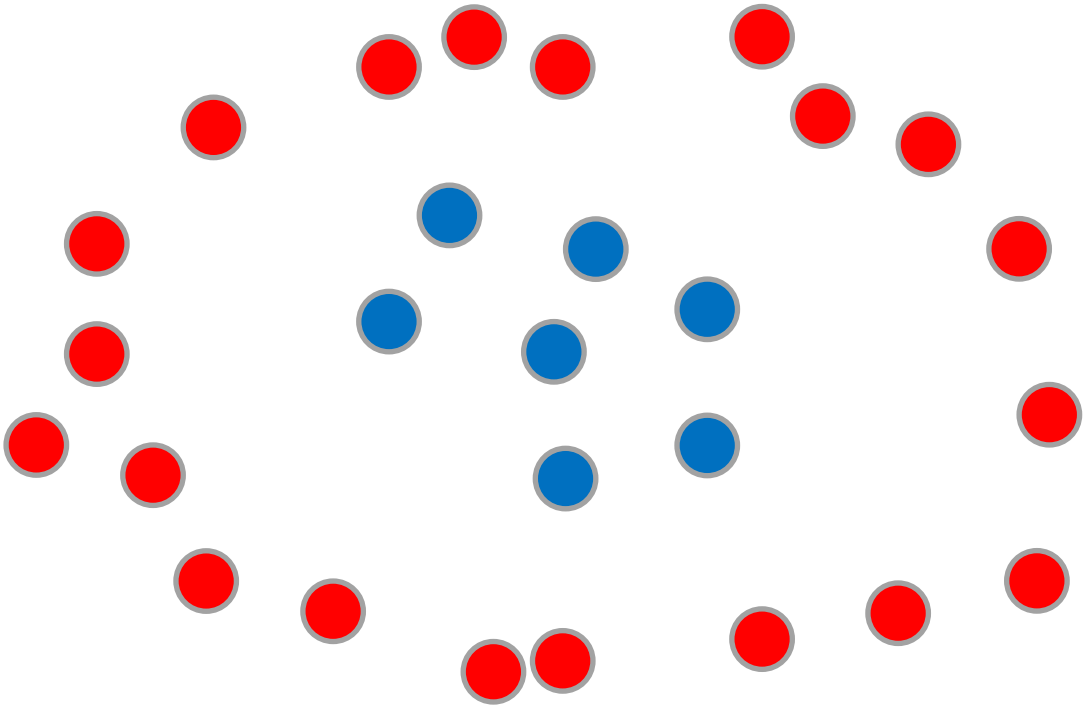
- Обучаемся не на всех данных, т.е. классификация получается хуже
- Проверяем качество только на малой части данных
- Как бы проверить качество на всех данных?

Скользящий контроль

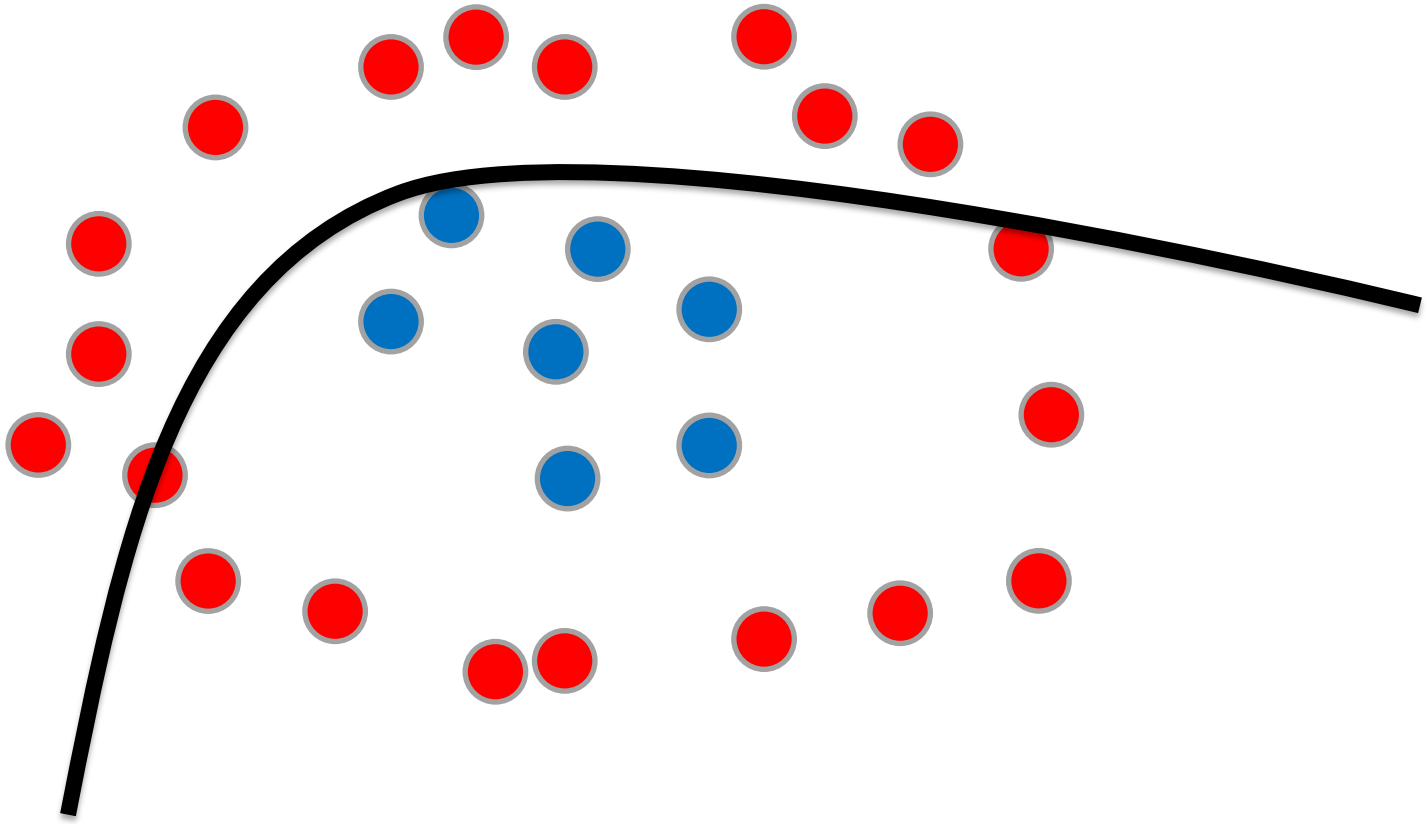
- Разбиваем обучающую выборку на N равных частей
- Поочередной выбрасываем каждую из частей, обучаемся на остальных и оцениваем качество
- Усредняем



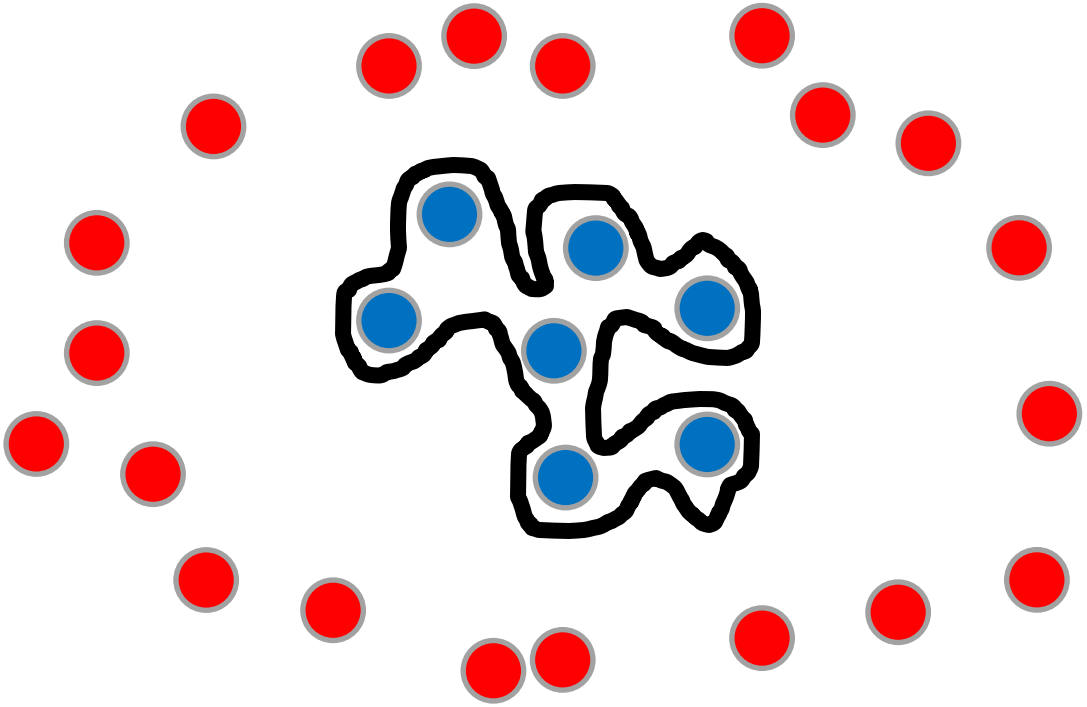
Как лучше выбрать границу?



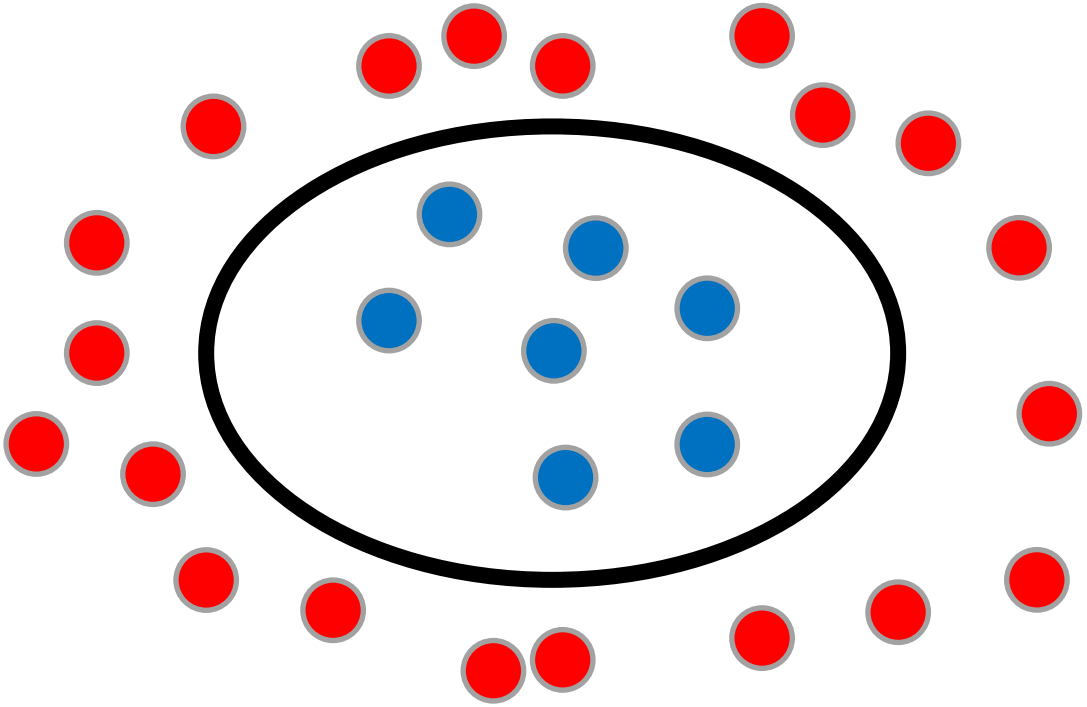
Недообученная модель



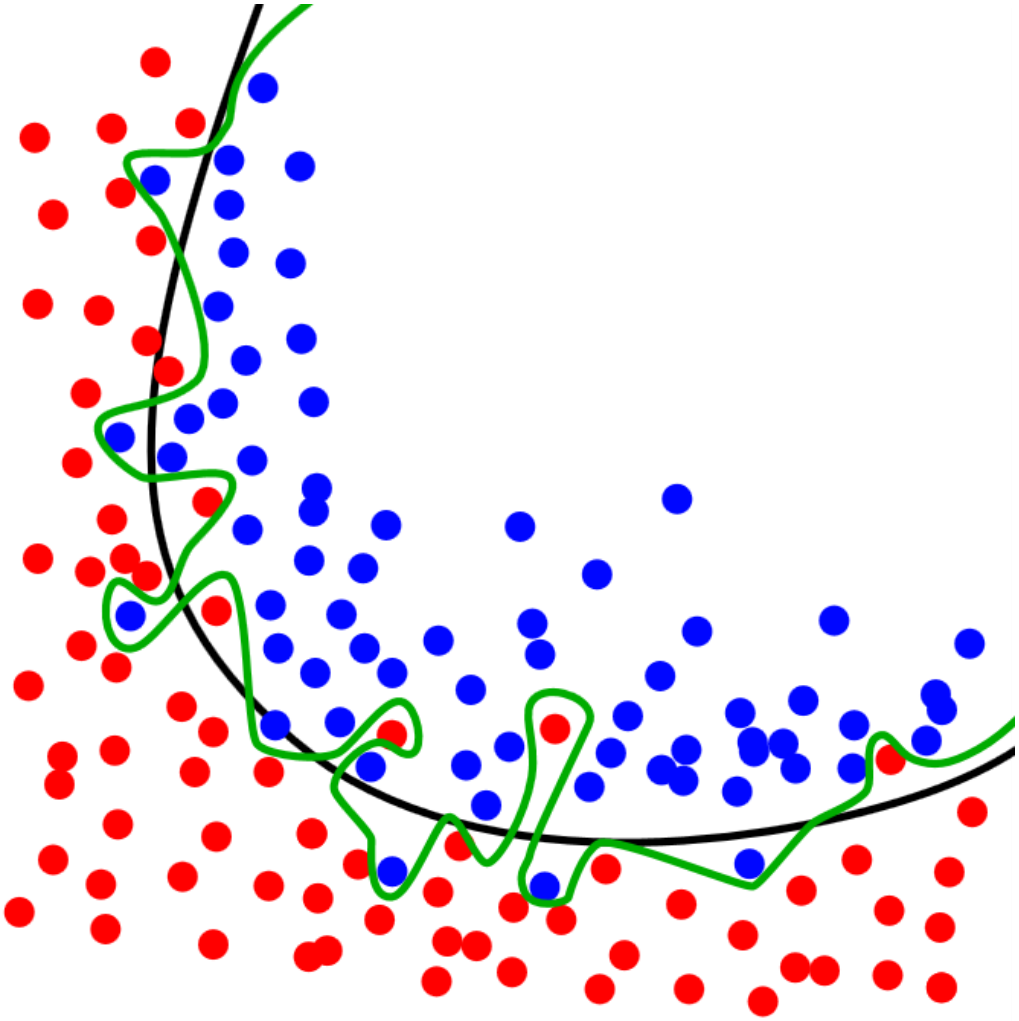
Переобученная модель



Оптимальная модель



Переобучение



Синонимы

- Распознавание, предсказание, прогнозирование
- Обучающая выборка, тренировочный набор объектов, наблюдение
- Тестовая выборка, контрольная выборка, валидационная выборка, скрытая выборка
- Классы, метки классов
- Скользящий контроль, кроссвалидация

Часть 4

Классификация ирисов

Классификация ирисов

- Решим реальную задачу
- Три биологических вида цветков:

Ирис щетинистый
(Iris setosa)

Ирис виргинский
(Iris virginica)

Ирис
разноцветный
(Iris versicolor)

Обучающая выборка

- Дана база из 150 конкретных цветков
- Каждого вида ровно 50 цветков (треть)
- Даны значения четырех параметров каждого цветка (в сантиметрах):
  - Длина чашелистика
  - Ширина чашелистика
  - Длина лепестка
  - Ширина лепестка
- Необходимо научиться классифицировать новые ирисы в один из трех видов

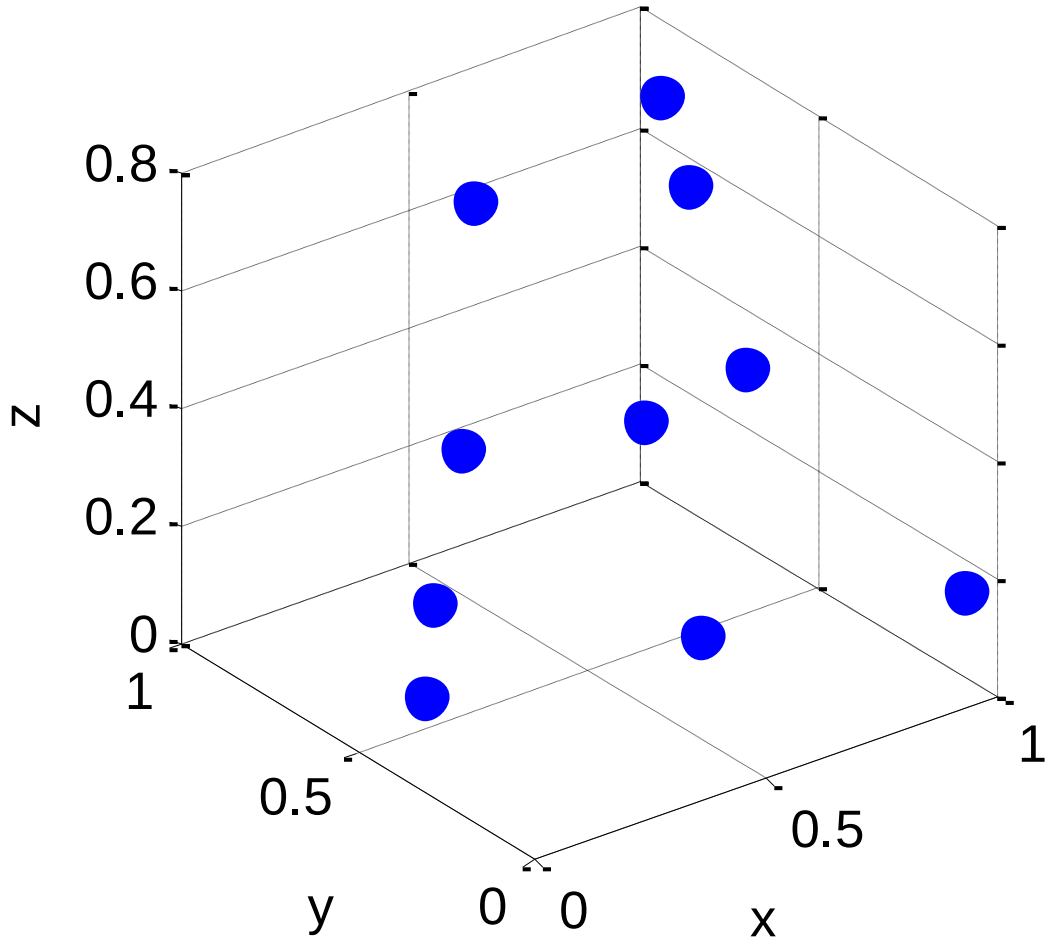
Обучающая выборка

- Произвольный кусок таблицы с данными:

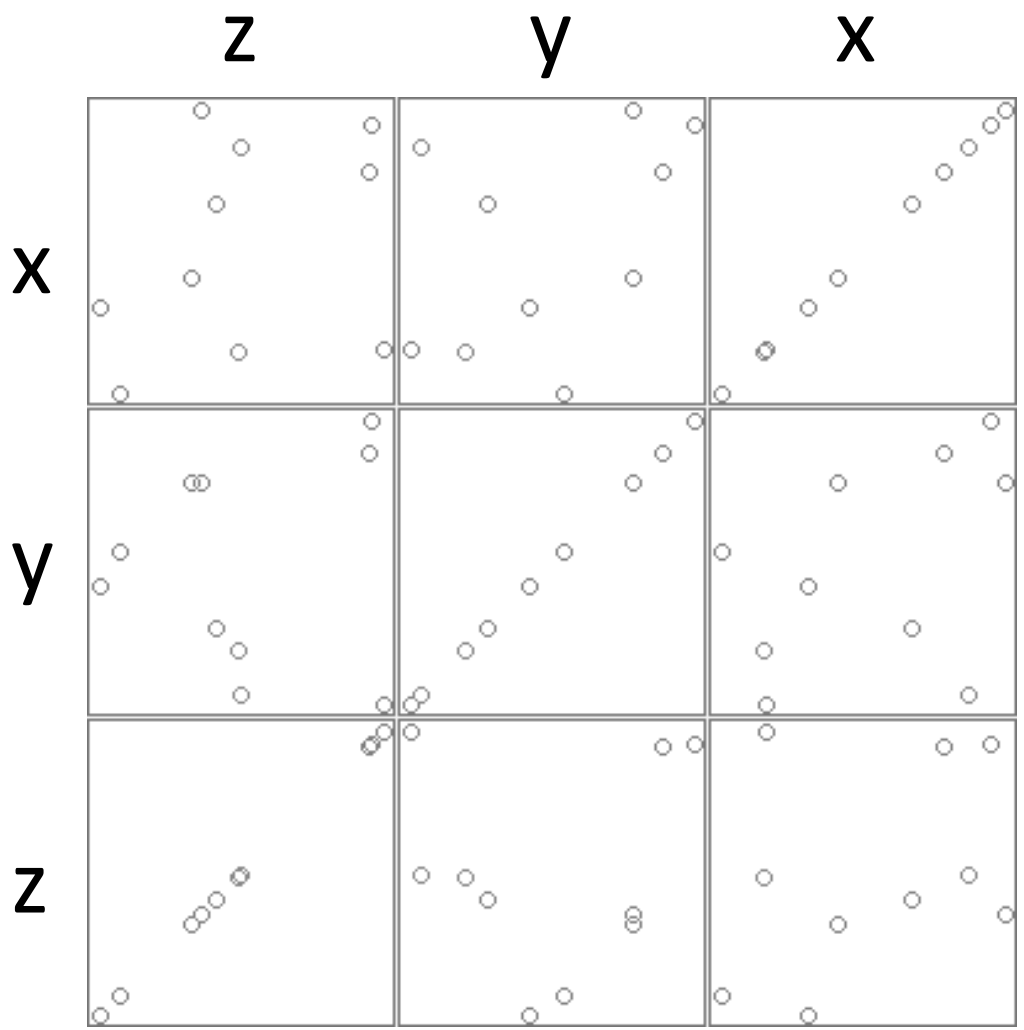
Длина чашелистика	Ширина чашелистика	Длина лепестка	Ширина лепестка	Вид ириса
7.7	3.8	6.7	2.2	**виргинский**
4.9	3.0	1.4	0.2	**щетинистый**
5.9	3.0	4.2	1.5	**разноцветный**
6.0	2.2	4.0	1.0	**разноцветный**
6.3	3.3	6.0	2.5	**виргинский**
5.2	3.5	1.5	0.2	**щетинистый**
5.7	2.9	4.2	1.3	**разноцветный**

Лирическое отступление

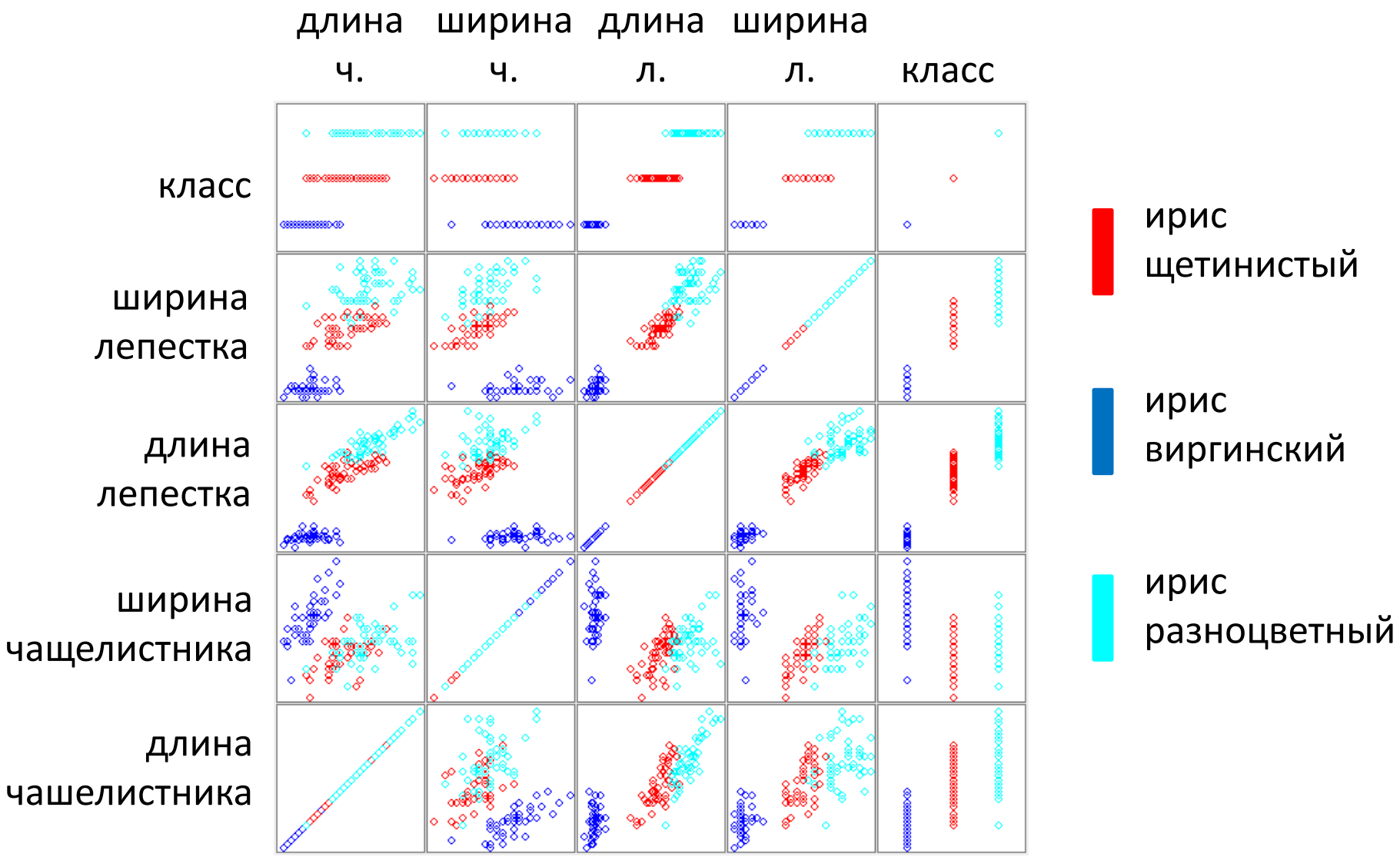
- Пусть объекты – точки в трехмерном пространстве



Двумерные проекции трехмерных данных



Как выглядят данные об ирисах?



Начнем решать задачу

- Опять воспользуемся методом ближайшего соседа
- Вопрос: как считать расстояние, ведь пространство четырехмерное?

Подсчет расстояния между точками a и b

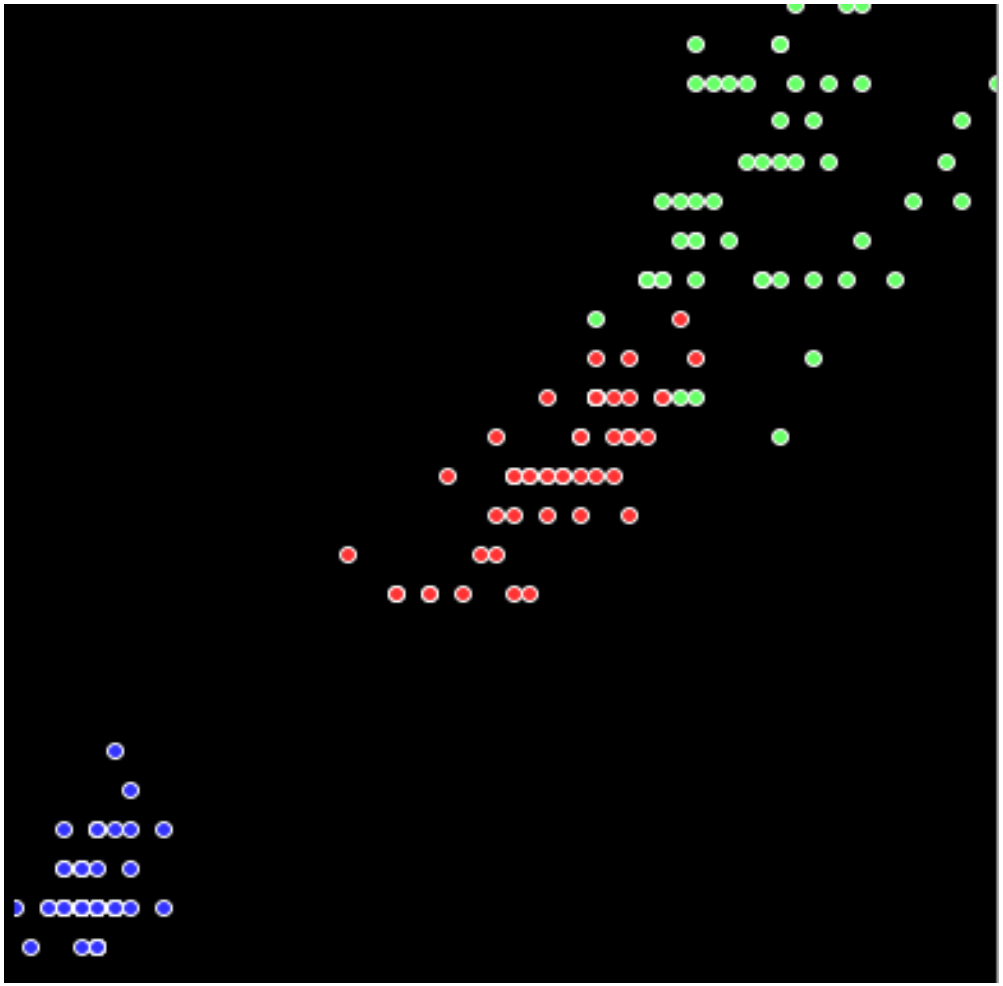
- Двумерный случай (на плоскости):

$$\sqrt{(a_x - b_x)^2 + (a_y - b_y)^2}$$

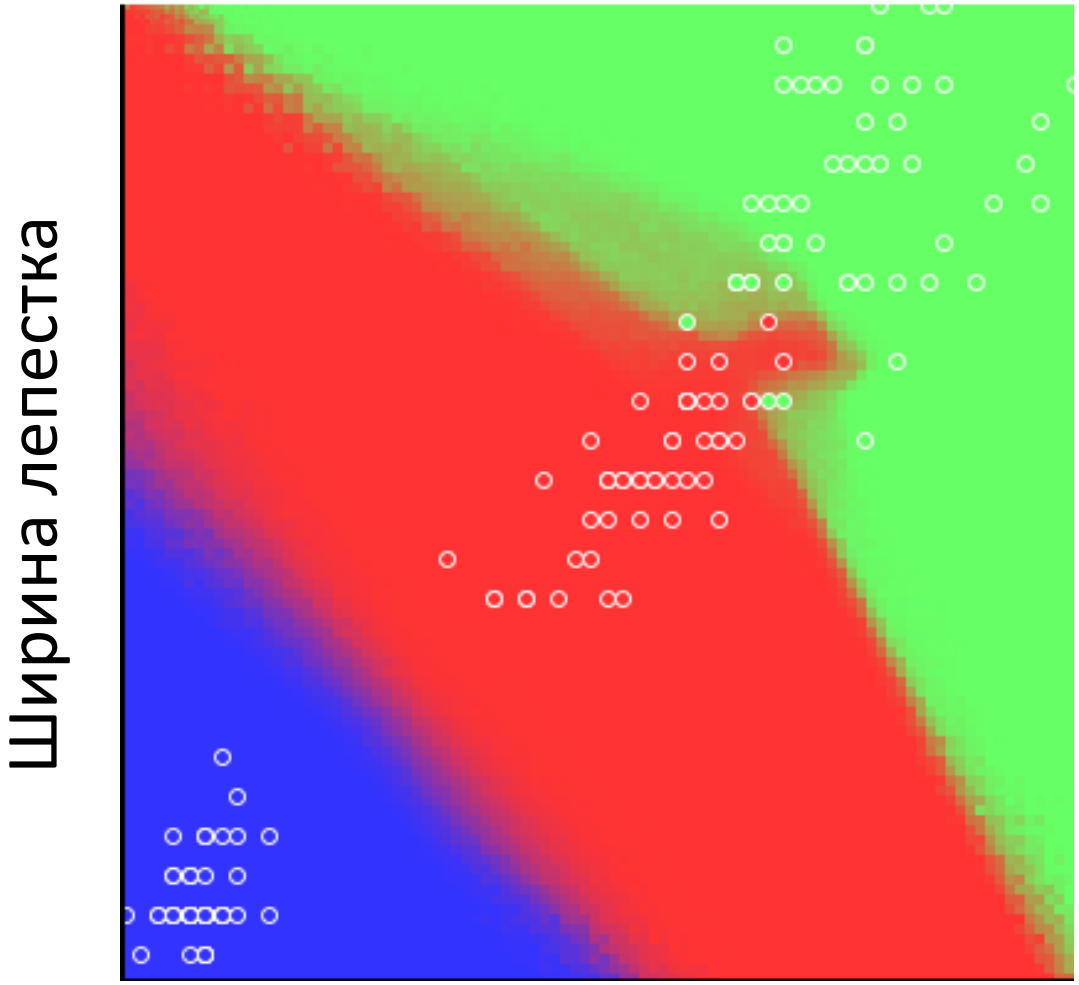
- Четырехмерный случай (аналогично):

$$\sqrt{(a_1 - b_1)^2 + (a_2 - b_2)^2 + (a_3 - b_3)^2 + (a_4 - b_4)^2}$$

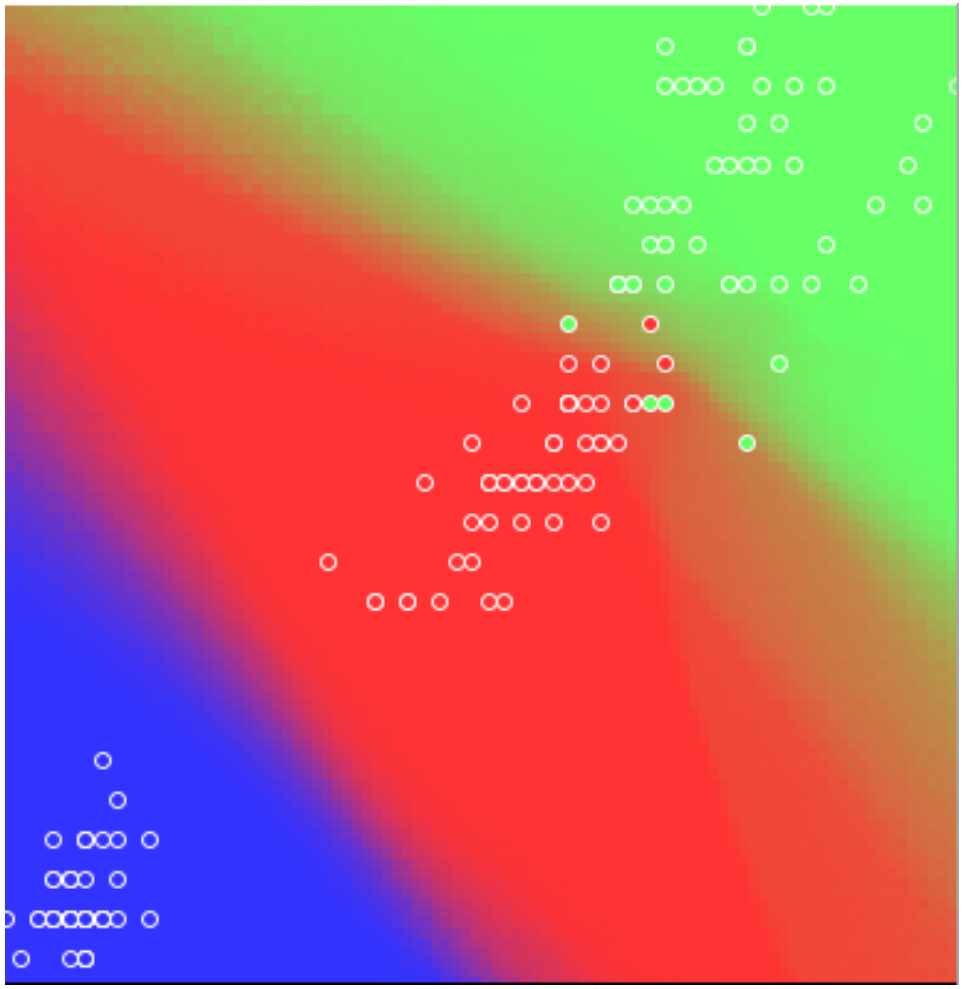
В разрезе по двум признакам



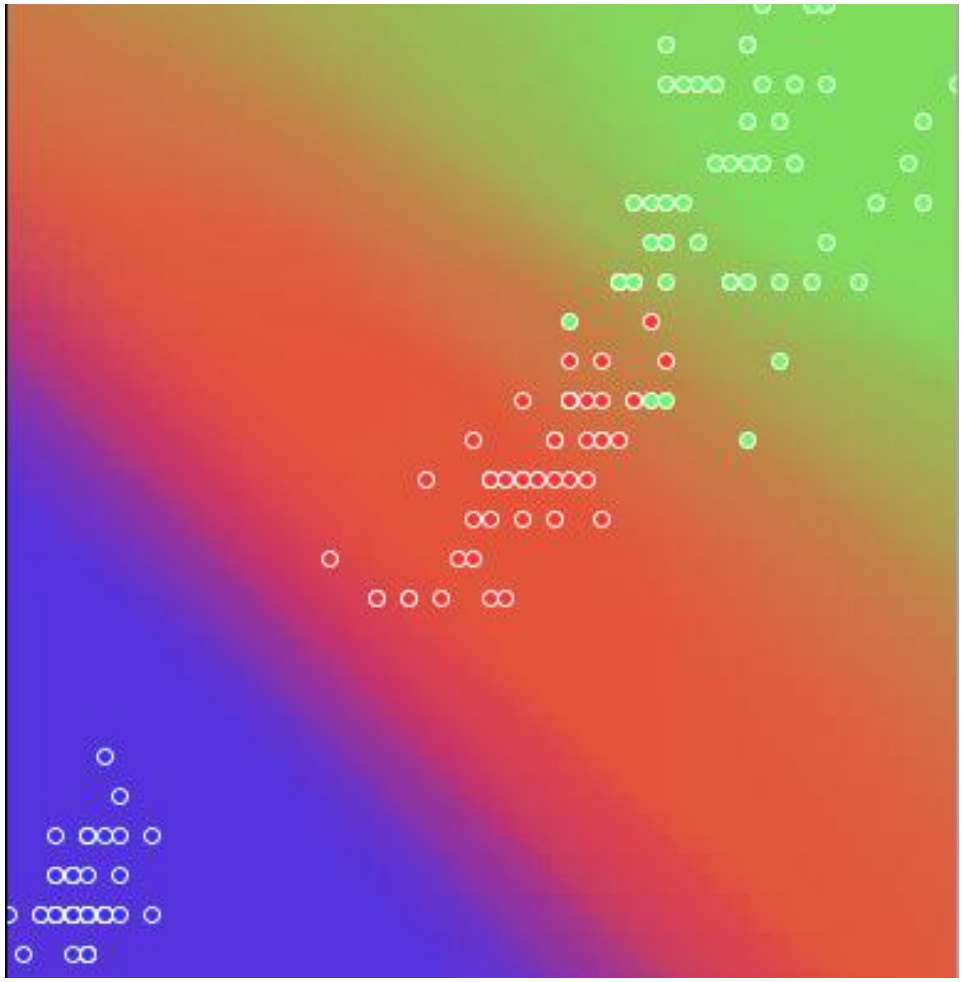
Граница разделения классов при k=1



Граница разделения классов при k=10

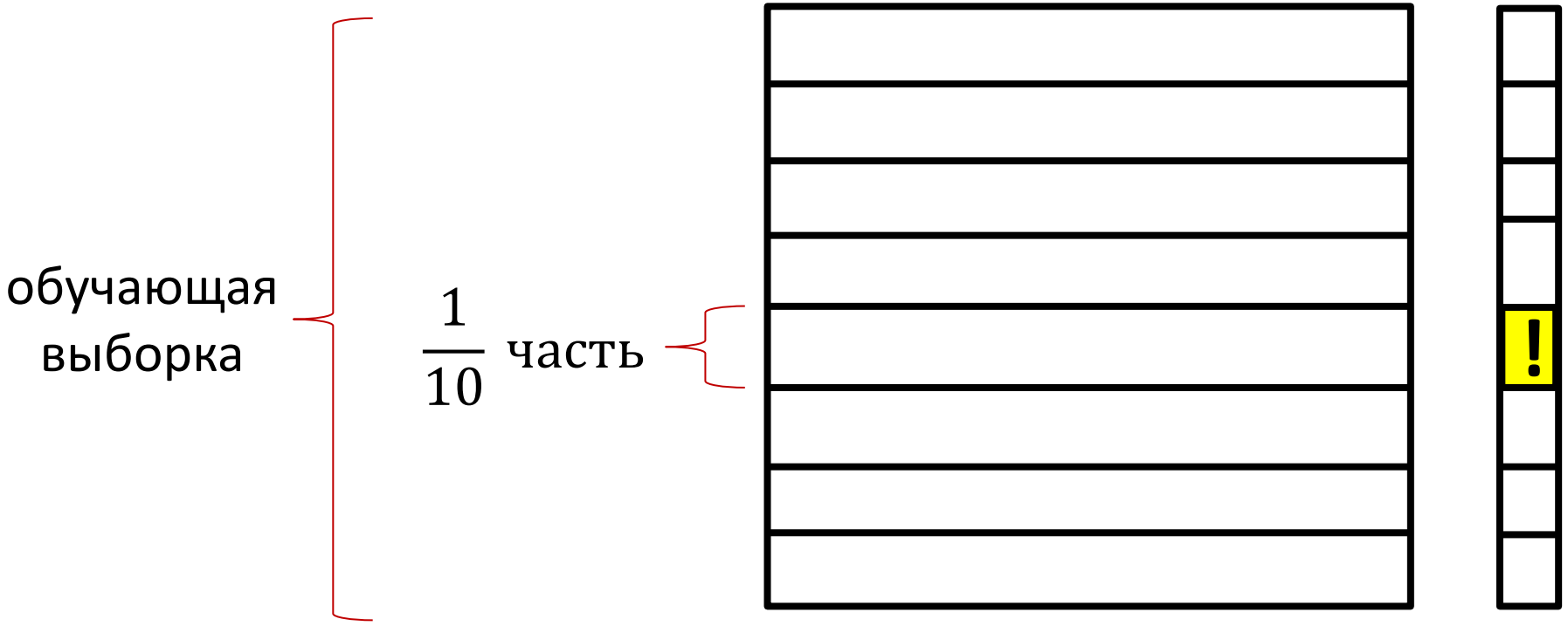


Граница разделения классов при k=60



Какое k выбрать? Скользящий контроль!

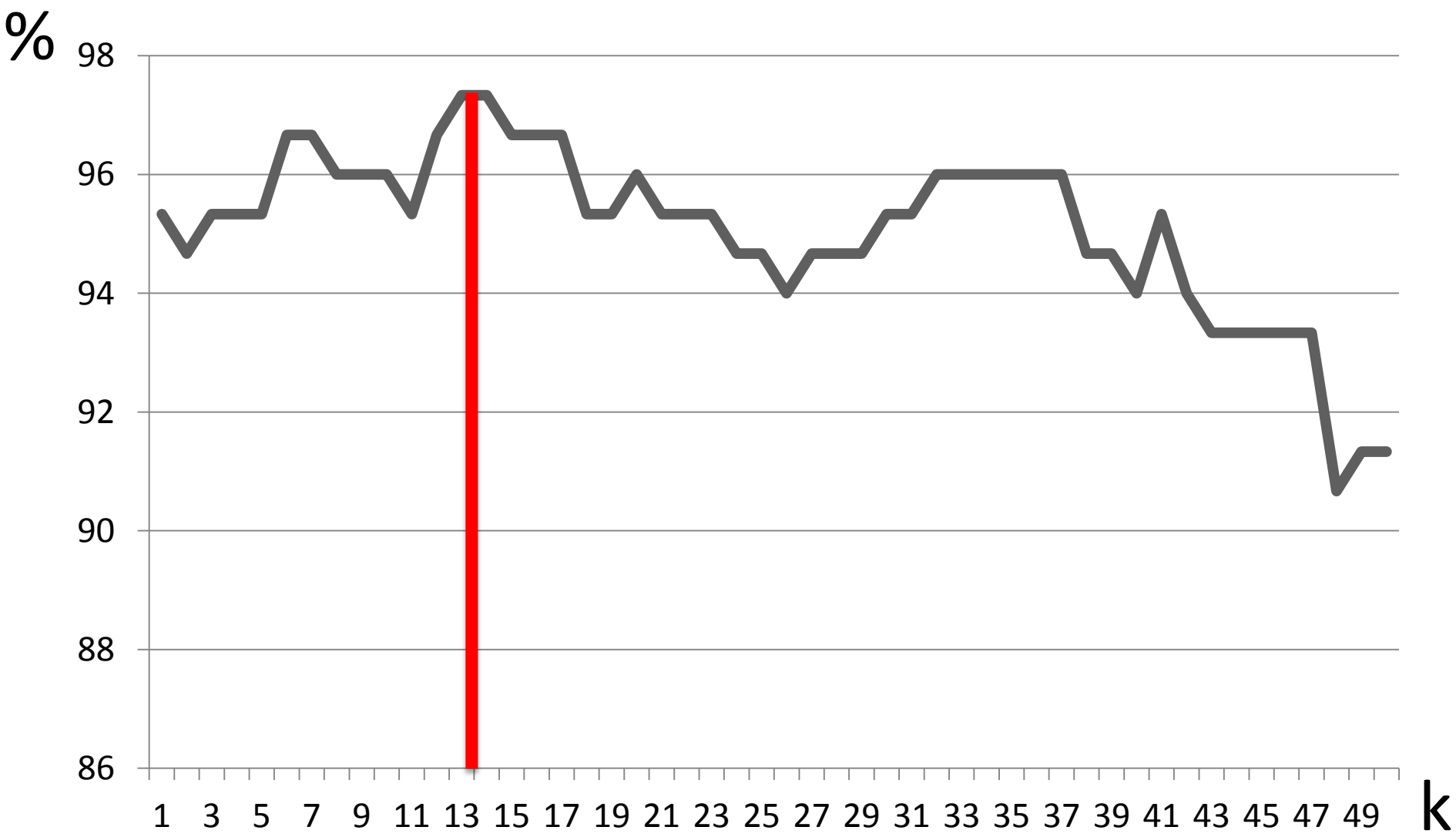
- Произвольно разбиваем обучающую выборку на 10 равных частей
- Поочередно выбрасываем каждую из частей, обучаемся на остальных и оцениваем качество
- Усредняем



Качество при разных k

- При k=1 качество составило 95.3333 %
- При k=2 качество составило 94.6667 %
- При k=3 качество составило 95.3333 %
- Посчитаем для всех k

Качество обучения в зависимости от k



Как точнее узнавать оптимальное значение k?

- Видно что график скачет, почему?
- Точно узнать k тяжело
- Проводить кроссвалидацию много раз, а затем усреднять

Еще раз про выбор расстояния

Несколько мыслей:

- Можно добавить признакам веса
  $\sqrt{\mathbf{10}(a_1 - b_1)^2 + (a_2 - b_2)^2 + (a_3 - b_3)^2 + (a_4 - b_4)^2}$
- Можно вообще считать расстояния вообще по-другому: $|a_1 - b_1| + |a_2 - b_2| + |a_3 - b_3| + |a_4 - b_4|$
- Как раз это и основная проблема в методе ближайших соседей – надо знать, как удачно считать расстояние в данной задаче

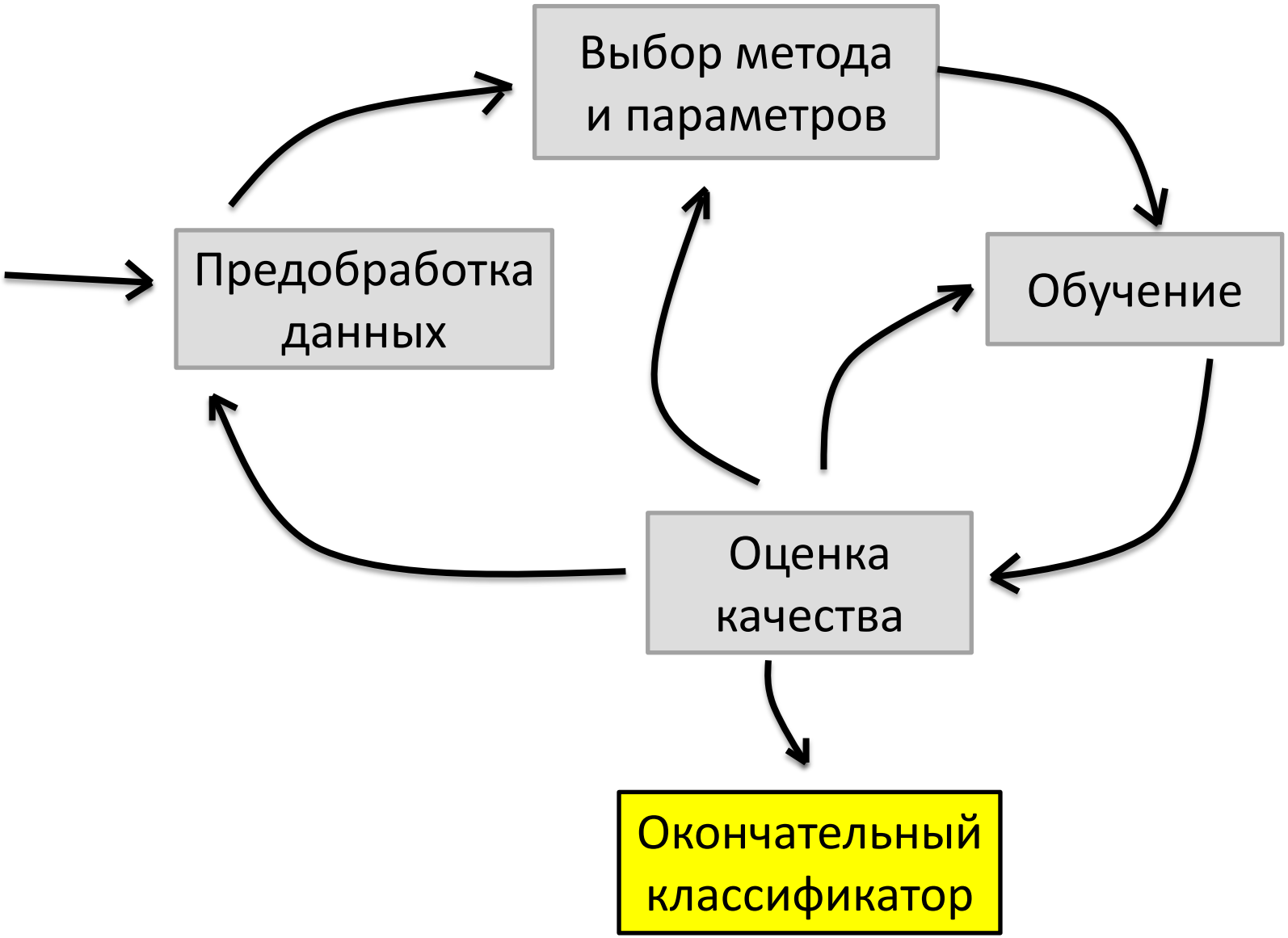
Еще раз про метод ближайших соседей

- Похожие объекты рядом
- Сложность: O(NM), N – количество объектов в обучении, M – количество новых объектов, O(1) – подсчет одного расстояния
- Структуры данных: kd-tree, R-tree, Ball-tree
- Нужно знать, как считать расстояние между объектами
- Есть один параметр – число соседей k

Параметры модели

- Количество параметров у алгоритма бывает куда больше
- Не всегда удается «тупо» перебрать все значения параметров у модели
- Придумываются разные методы нахождения параметров

Цикл решения задачи



Часть 5

Практика решения реальных задач

Про терминологию

- Интеллектуальный **анализ данных** (Data Mining)
- **Машинное обучение** (Machine Learning, Statistical Learning)
- **Прикладная статистика** (Applied Statistics)
- **Факторный анализ** (Factor Analysis)
- **Глобальная оптимизация** (Global Optimization)
- **Искусственный Интеллект** (Artiffitial Intellengence)

Соревнования по анализу данных

- Сайты
  - Kaggle.com
  - Tunedit.org
  - Яндекс Интернет-Математика
- Кем проводятся
  - Компаниями
  - Работодателями
  - Университетами

Отличия от олимпиадного программирования

- Дается одна задача, а не несколько
- Решаются значительно дольше (недели, месяцы, годы)
- Не существует точного и правильного решения, проводится много экспериметнтов, чтобы понять, какое решение выбрать
- Идет борьба за сущие проценты качества
- Не важен язык, скорость работы и ресурсы; важен только результат
- В одиночку или командами

На чем пишут алгоритмы обучения?

- Готовые наборы методов машинного обучения (для общего понимания, какой метод лучше)
  - Weka
  - RapidMiner
  - Orange
- Интерпретируемые языки (для экспериментов и выбора алгоритма)
  - Matlab (Octave – бесплатная версия)
  - Python (+ библиотеки NumPy, SciPy, и т.п.)
  - R
- Более низкоуровневые языки (для скорости работы, когда уже ясно, какой алгоритм будет использоваться)
  - C
  - C++

Примеры реальных задач

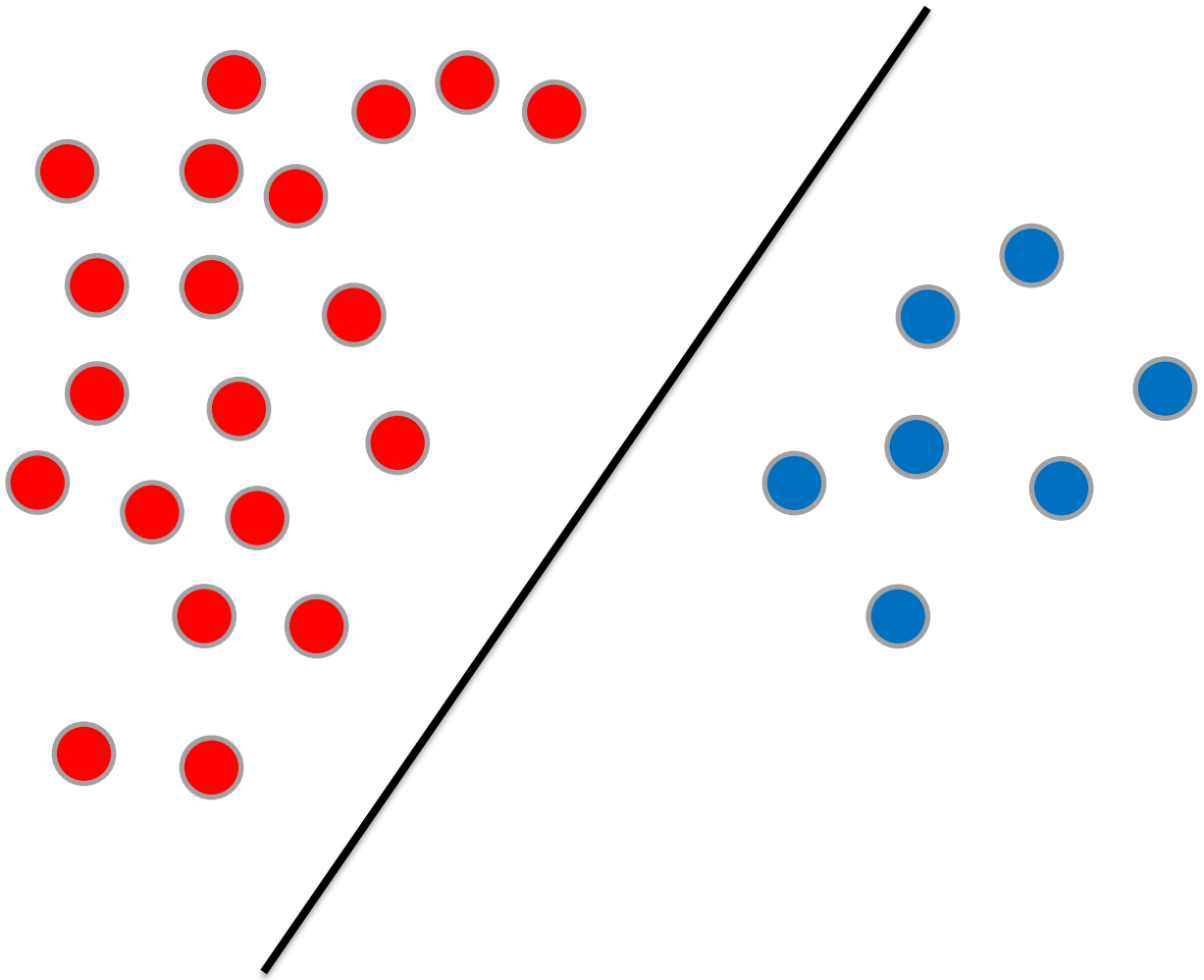
- Геологические данные – ищем золото
- Компьютерное зрение – распознавание чего-то на картинках
- Военная оборона – птица или ракета?
- Рекомендательные системы на сайтах
- Медицина – наличие болезни по симптомам
- Прогнозирование пробок
- Распознавание сигналов головного мозга
- Сайт научных статей - категоризация текстов
- Кредитный скоринг – надежность клиентов банка
- И еще очень много чего...

Лекция 2

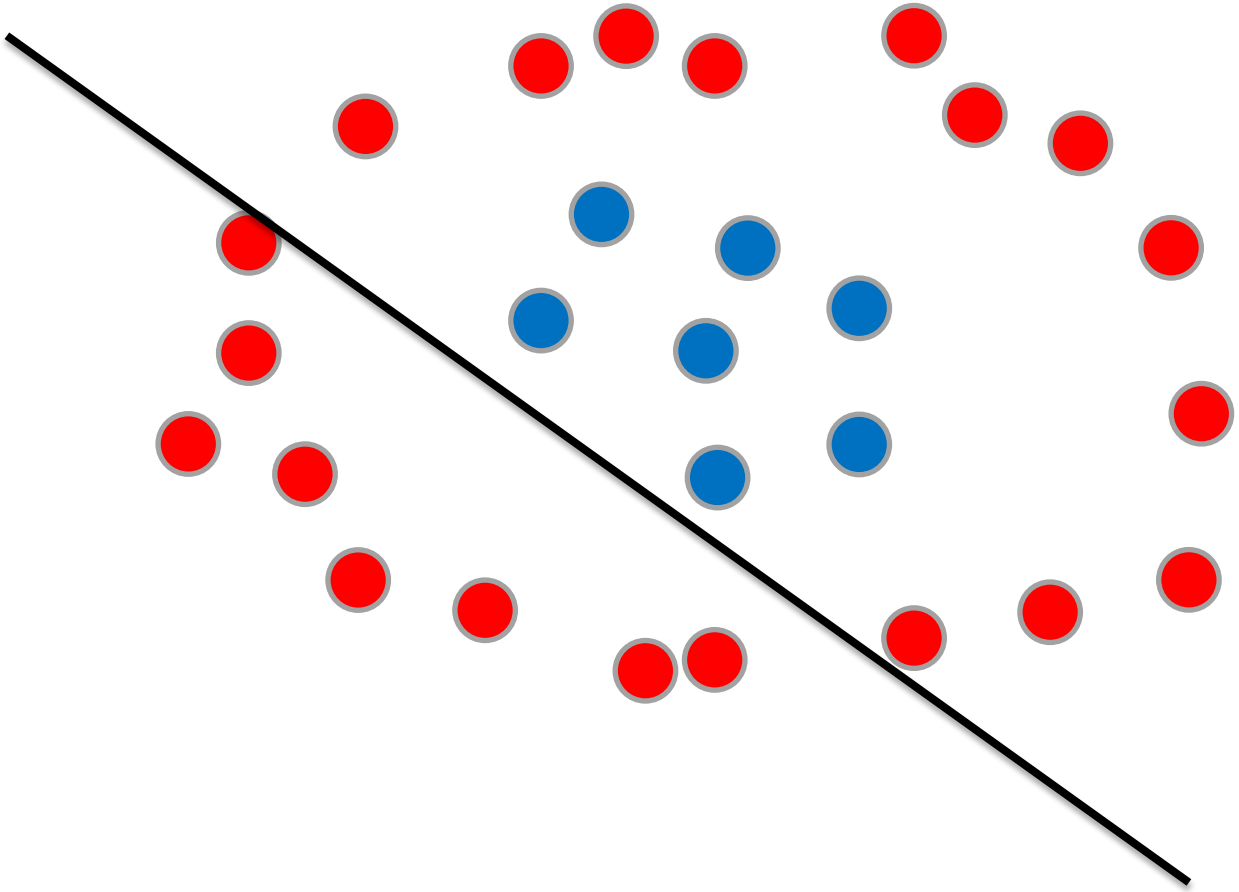
Часть 6

Линейные классификаторы

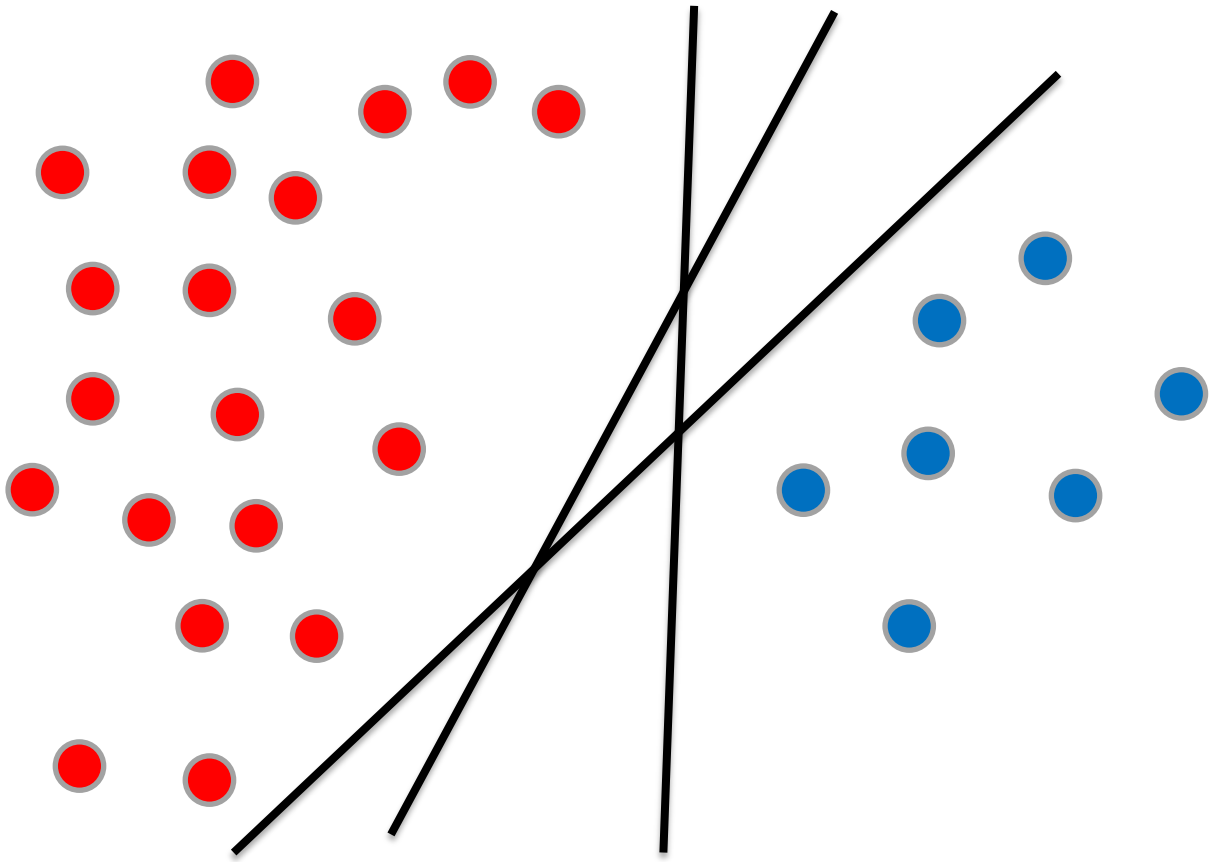
Пусть граница – прямая



Иногда прямая плохо помогает

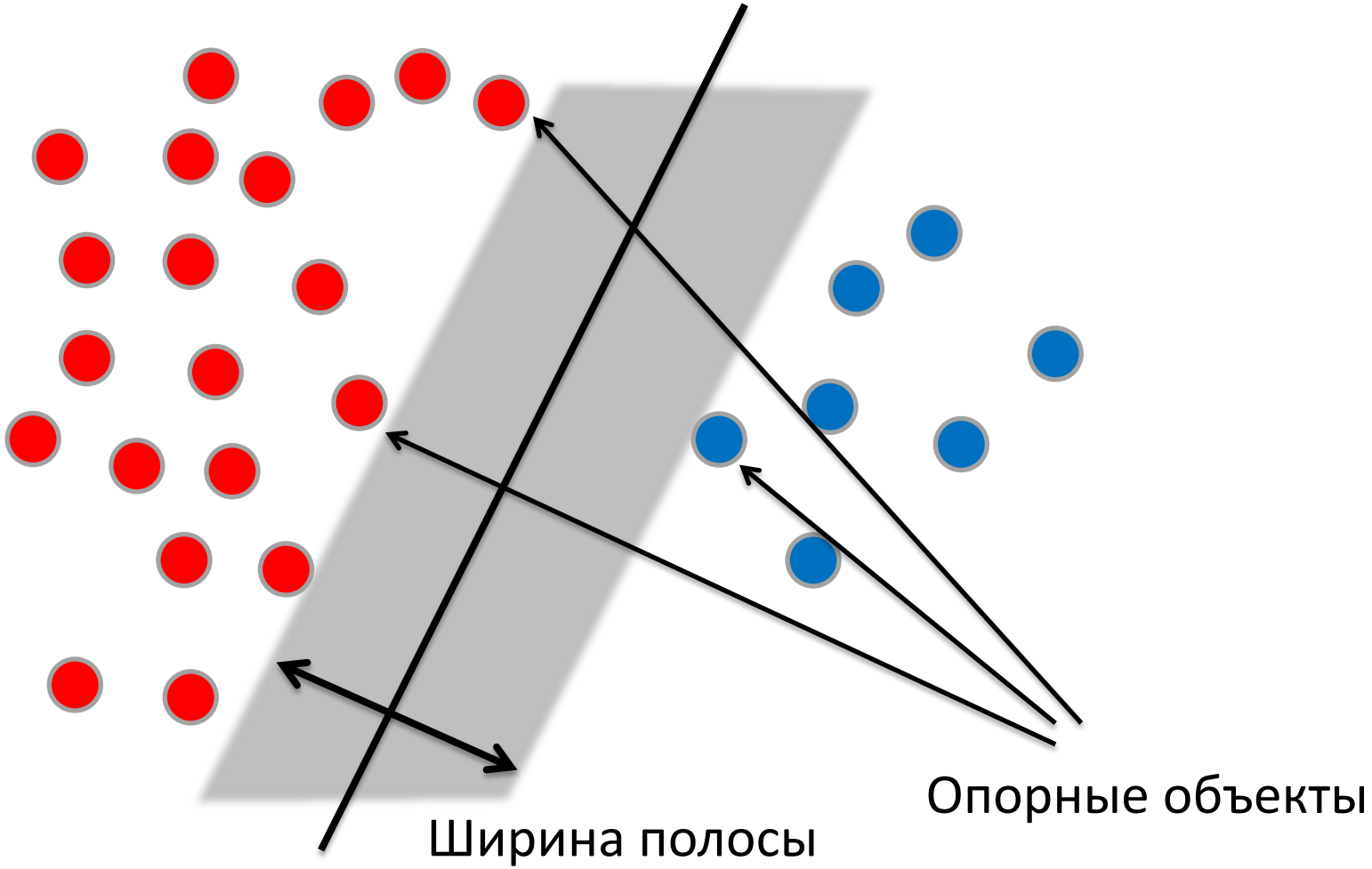


Есть много способов провести прямую

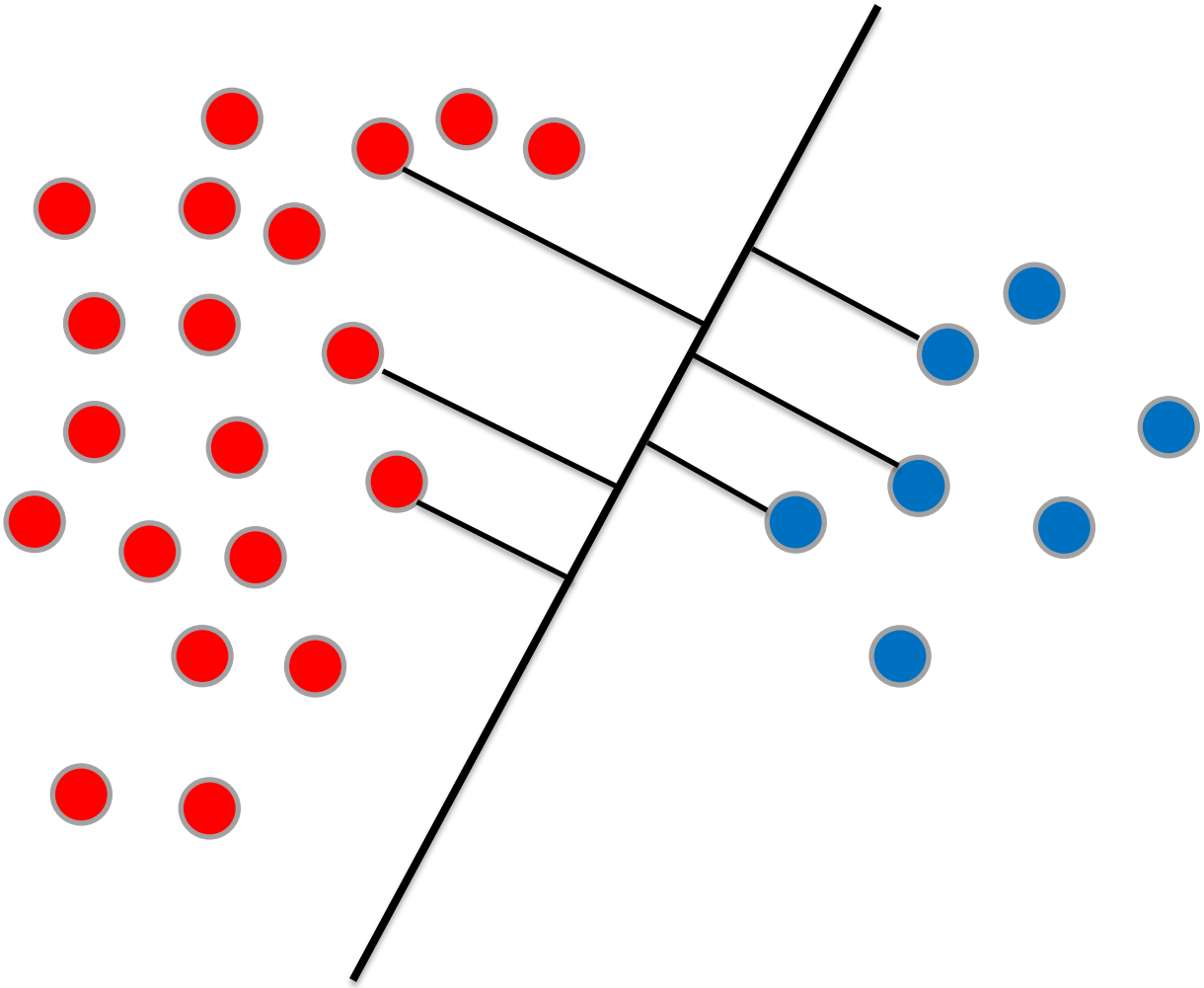


Какая прямая лучше?

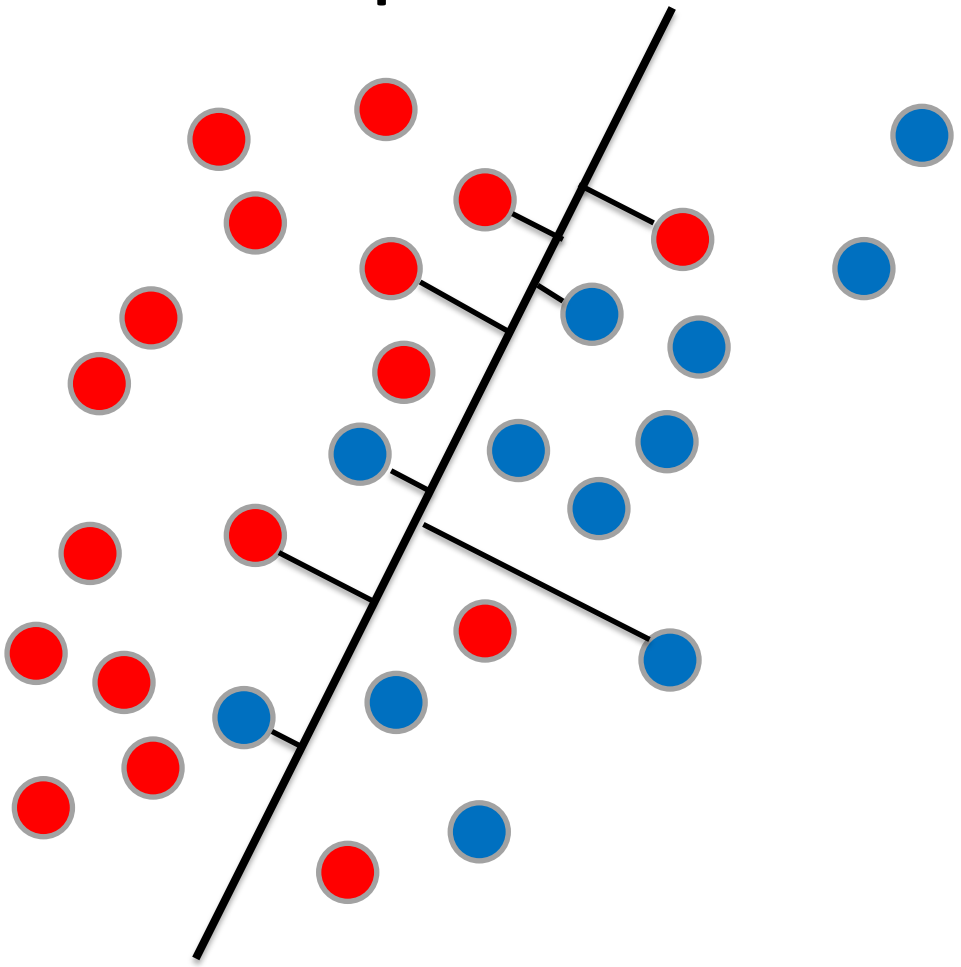
Идея: строить самую широкую полосу, и ее центр – нужная прямая



Другая идея: отступы должна быть максимальны

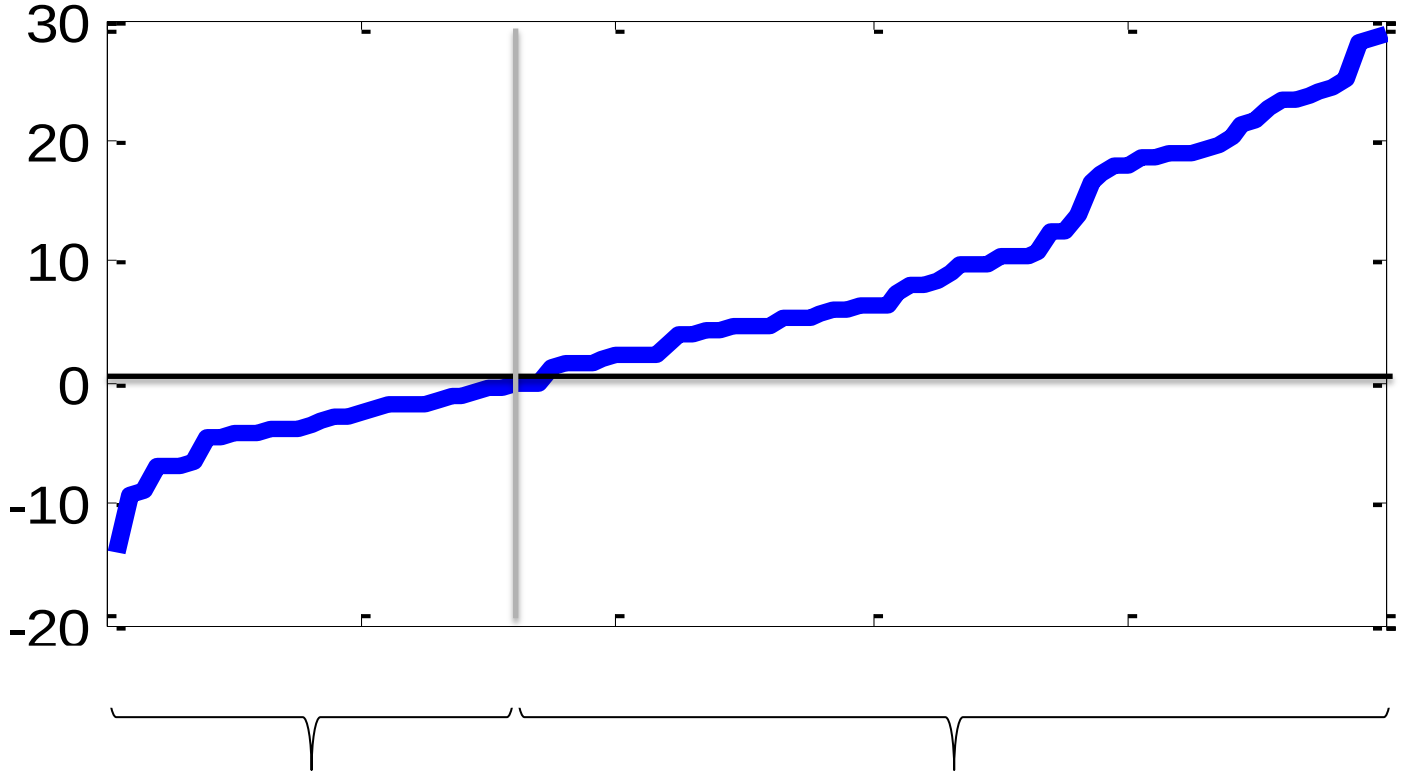


Если объекты расположены хуже



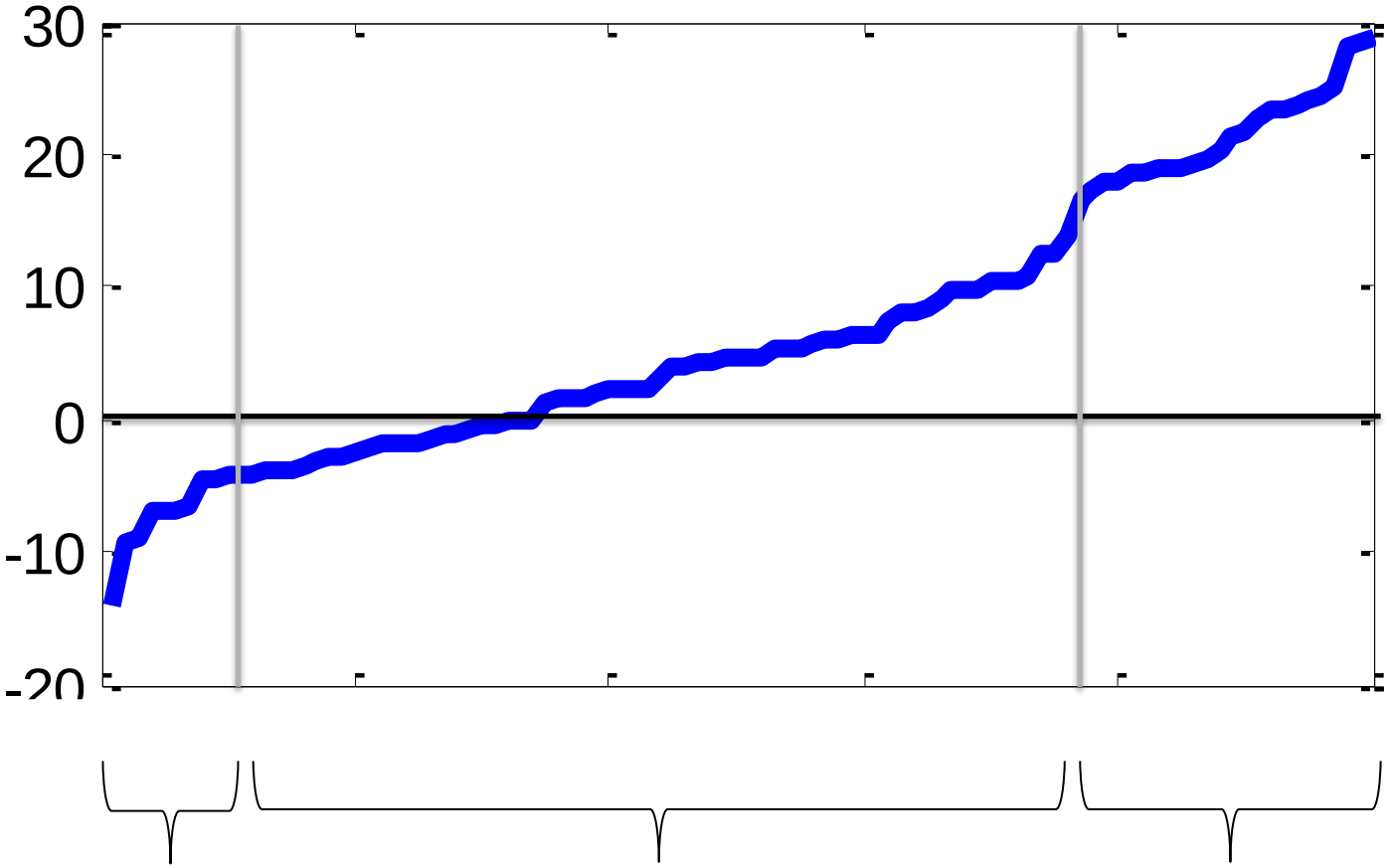
Отступы могут быть отрицательными! А это и не плохо :)

Отсортируем значения отступов



Важность

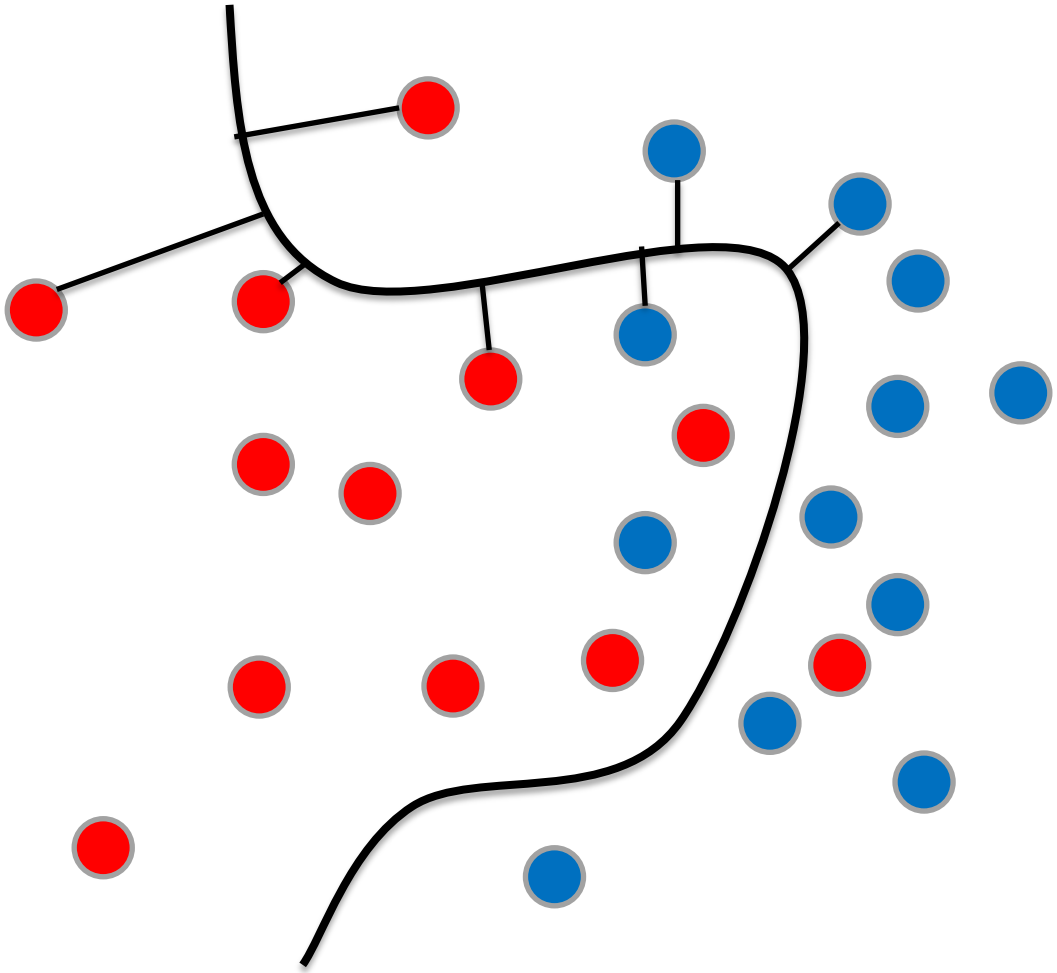
Отсортируем значения отступов



Метод опорных векторов (SVM)

- Основан на принципе максимизации отступов
- Математически точно вычисляется
- Использует только часть объектов, т.н. «опорные векторы»

Максимизация отступов для сложной границы



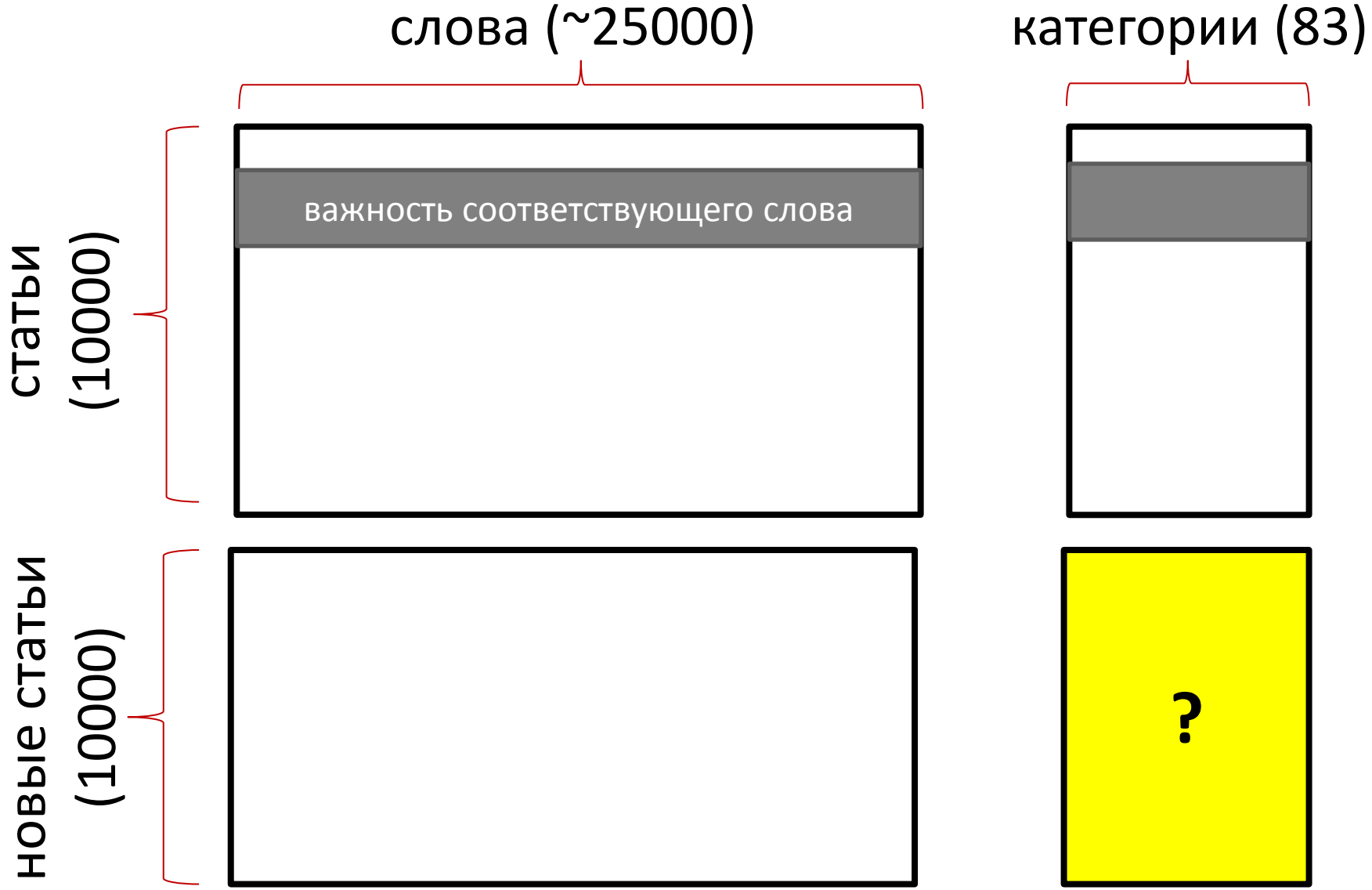
Гиперплоскости в многомерных пространствах

- В двумерном случае – прямая, в трехмерном – плоскость, дальше – гиперплоскость
- Главное, что она линейна и делит все пространство на два полупространства
- Метод опорных векторов умеет строить гиперплоскости

Часть 7

Категоризация текстов

Обучающая выборка



Начнем решать

- Будем рассматривать каждую из 83 категорий отдельно, т.е. по сути решим 83 отдельных задачи классификации
- Размерность пространства значительно больше количества объектов, т.е. разделяем лининейно
- Используем метод опорных векторов!

Результат

- Оказывается, даже такое элементарное, грубое и общеизвестное решение приводит к высоким позициям в таблице результатов соревнования

Еще несколько мыслей:

- Часто полезно отбросить частые и редкие слова
- Разумно объединять признаки
- Популярные методы в задачах про тексты:
  - ближайший сосед
  - линейные алгоритмы
  - вероятностные соображения

Часть 8

Распознавание пешеходов на изображениях

Учимся распознавать пешеходов

Исходная постановка задачи

- Дан набор фотографий (250 штук)
- Для каждой фотографии известны координаты кусочков, в которых содержатся пешеходы
- Надо распознать пешеходов на новых изображениях
- Обнаружение считается верным, если оно совпадает с настоящим хотя бы на 50 процентов площади

Небольшое упрощение

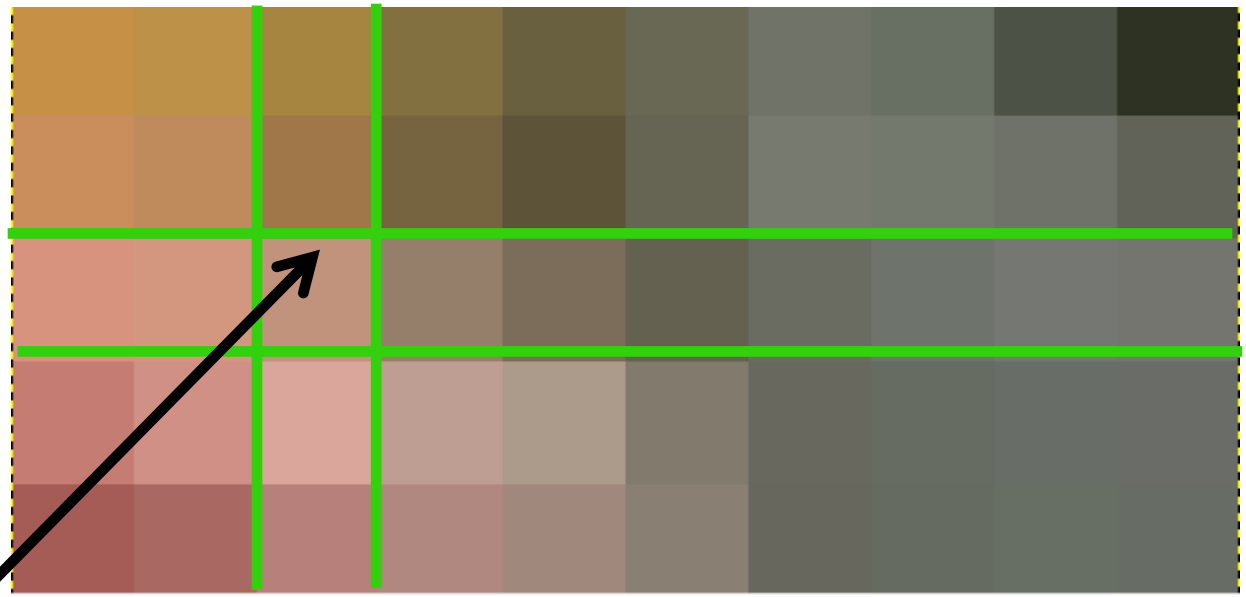
- Все фотографии имеют высоту 200 пикселей
- Все пешеходы имеют размер 200 на 80 пикселей

Приведем задачу к стандартному виду

- Объекты: картинки размера 200*80
- Два класса: пешеход и фон (не пешеход)
- Обучающая выборка: картинки с пешеходами и картинки с фоном, вырезанные из предоставленных фотографий
- Признаки объектов – их пока нет! Как по картинке получить характеризующий её набор чисел?

Что такое изображение?

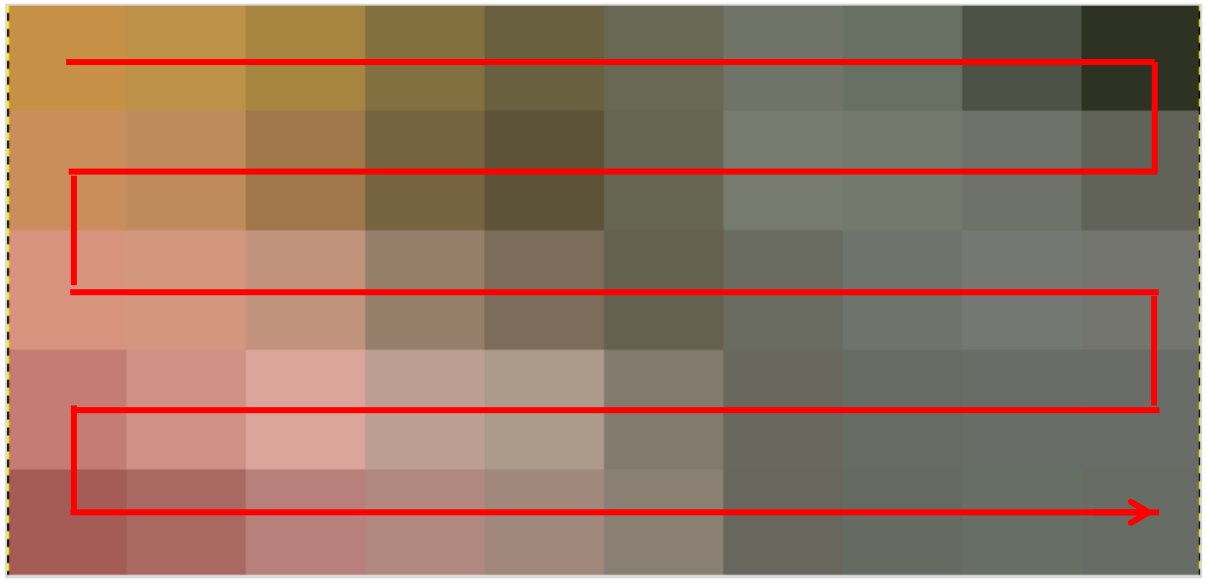
- Матрица пикселей
- Каждый пиксель имеет свой цвет (число)



Число, кодирующее цвет

Первая идея:

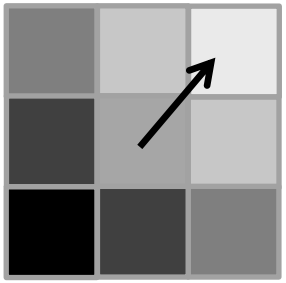
- развернуть матрицу пикселей в одну строчку и использовать это как набор признаков



Получится плохо. Почему?

- Цвета мало о чем говорят – важнее граница!
- Человеческий мозг в основном анализирует именно её.

Вектор изменения яркости



- В каждой точке картинки можем найти направление, в котором быстрее всего увеличивается яркость
- Чтобы посчитать вектор для конкретного пикселя, потребуются его соседи

Будем использовать такие векторы в качестве признаков

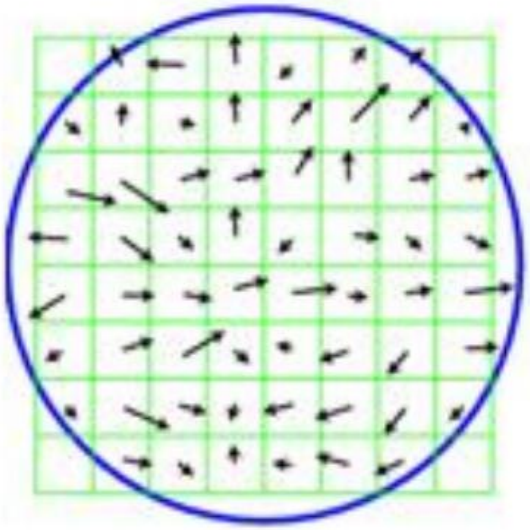
- В каждом пикселе находим вектор изменения яркости (направление и длина)
- Получается матрица из направлений (т.е. чисел от 0 до 2π) и длин
- Разворачиваем матрицу, как мы это уже делали раньше

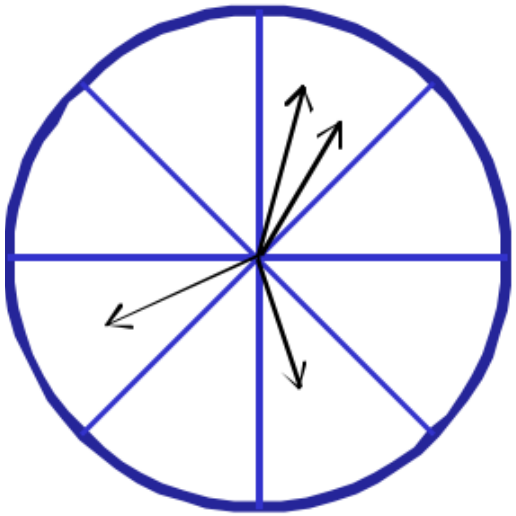
Все опять плохо!
В чем дело?

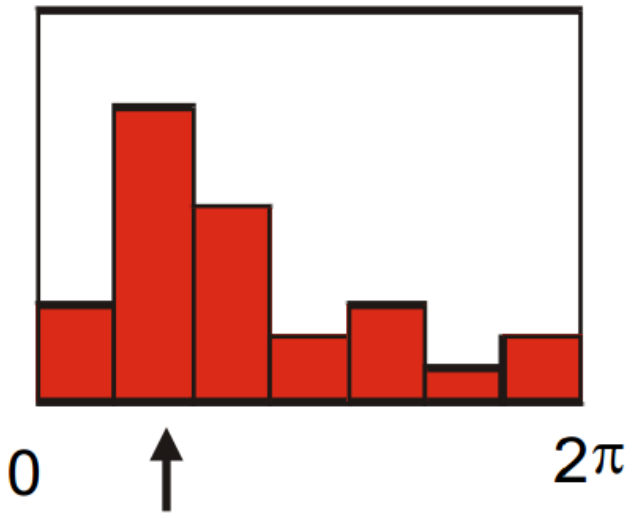
- Значения признаков сильно изменятся, если пешеход сдвинется хотя бы чуть-чуть
- Мы получим огромное количество признаков (2*число пикселей)
- Как учесть векторы изменения яркости в пикселях хитрее?

Идея!

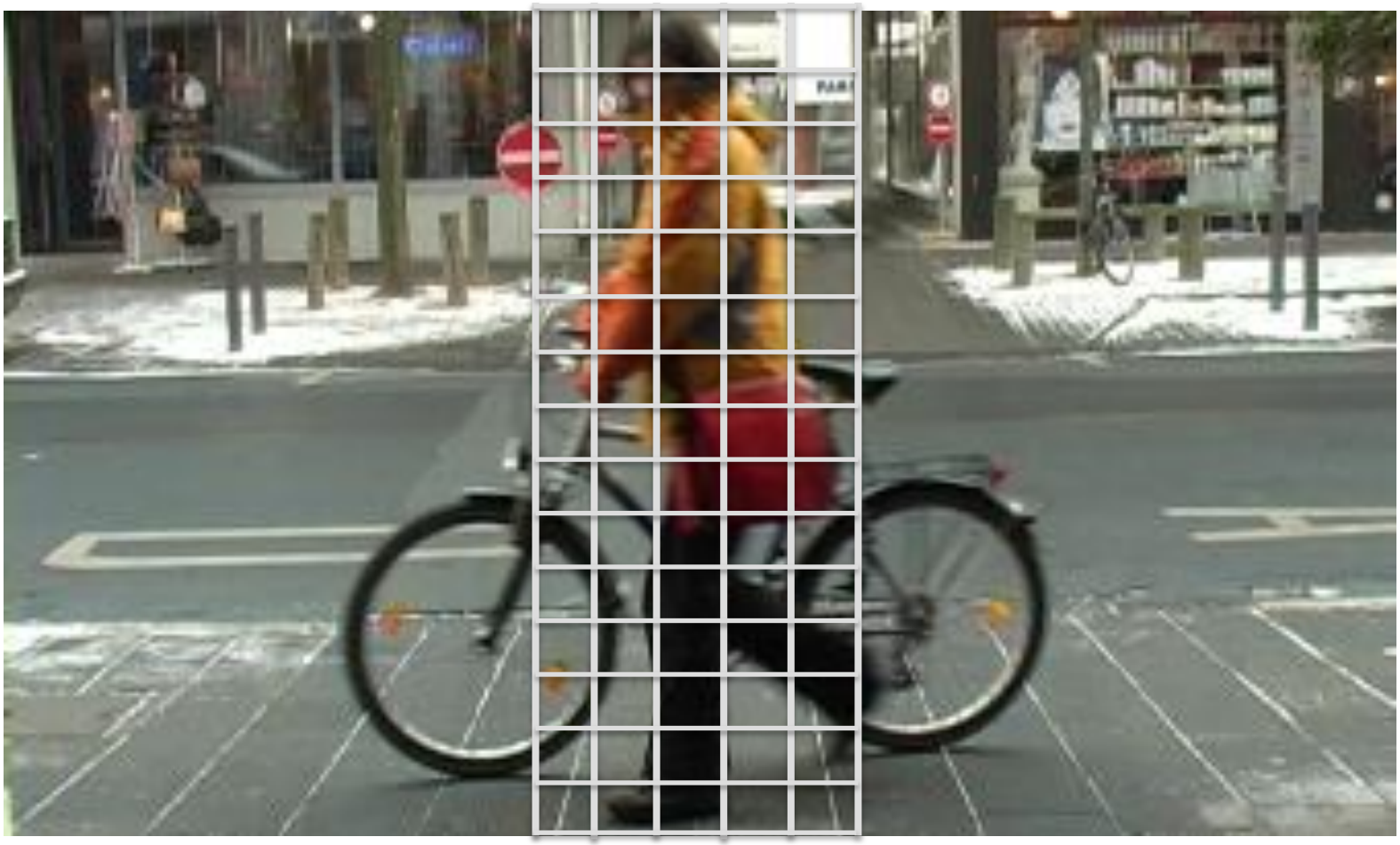
- Делим промежуток от 0 до 2π на несколько участков (пусть 8)
- В каждой точке определяем, какому участку принадлежит направление, и увеличиваем нужный столбик. Получается гистограмма.







Итак:



Мы научились искать признаковое описание картинки:

- Делим картинку на много небольших квадратов (N штук)
- Для каждого квадрата строим гистограмму направлений – это 8 чисел
- Разворачиваем все в строчку – получаем 8*N чисел

Преимущества метода:

- Признаковое описание не изменится, если пешеход чуть-чуть сдвинется или повернется
- Так как производится усреднение по нескольким пикселям, сама собой будет примерно учитываться длина вектора
- Признаков становится значительно меньше

Вернемся к задаче. Схема решения:

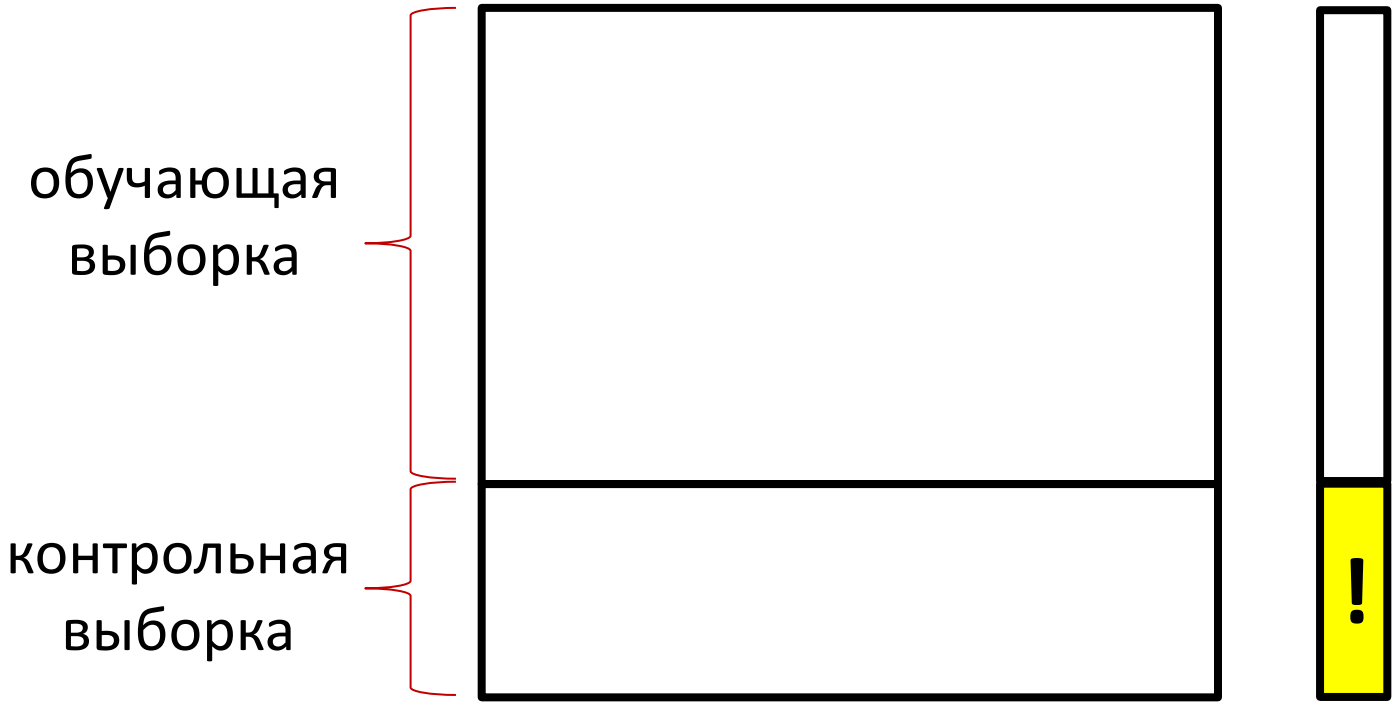
- Предобработка данных: научиться для картинки 200*80 определять признаки, т.е. характеризующий её набор чисел
- Задача классификации: построить алгоритм, относящий каждую картинку к классу пешеходов или фона
- Визуализация: пробежать по всем кусочкам 200*80 новых изображений, если алгоритм относит его к классу пешеходов, то обвести в рамку

Как будем решать задачу классификации?

- Опять объектов мало, а признаков много
- Попробуем применить метод опорных векторов

Проверка качества

- Качество – насколько хорошо работает алгоритм на контрольной выборке



Качество в нашей задаче

Качество в нашей задаче

- Полнота – число правильно найденных классификатором пешеходов / настоящее число пешеходов
- Точность – число верных обнаружений / общее число обнаружений

Неплохо!

- Полнота: 96 %
- Точность: 78 %

- Посмотрим, как работает реальная программа

Что можно предпринять?

- Подавление повторных выделений

Как это сделать?

Жадный алгоритм:

- ищем на изображении картинку с максимальным отступом, относим её к классу пешеходов, отступы для всех «соседних» картинок зануляем
- так продолжаем, пока максимальный отступ картинок не станет меньше порога

Выбор примеров фона

- Проблема: фон очень разнообразен, какие примеры фона выбрать?
- Идея:
  - изначально выбрать лишь несколько примеров
  - обучиться на них
  - применить алгоритм ко всем картинкам из данных фотографий
  - добавить в обучающую выборку картинки, на которых алгоритм ошибся
  - и так далее

Оказывается, две абсолютно разных задачи решаются одинаковыми методами!

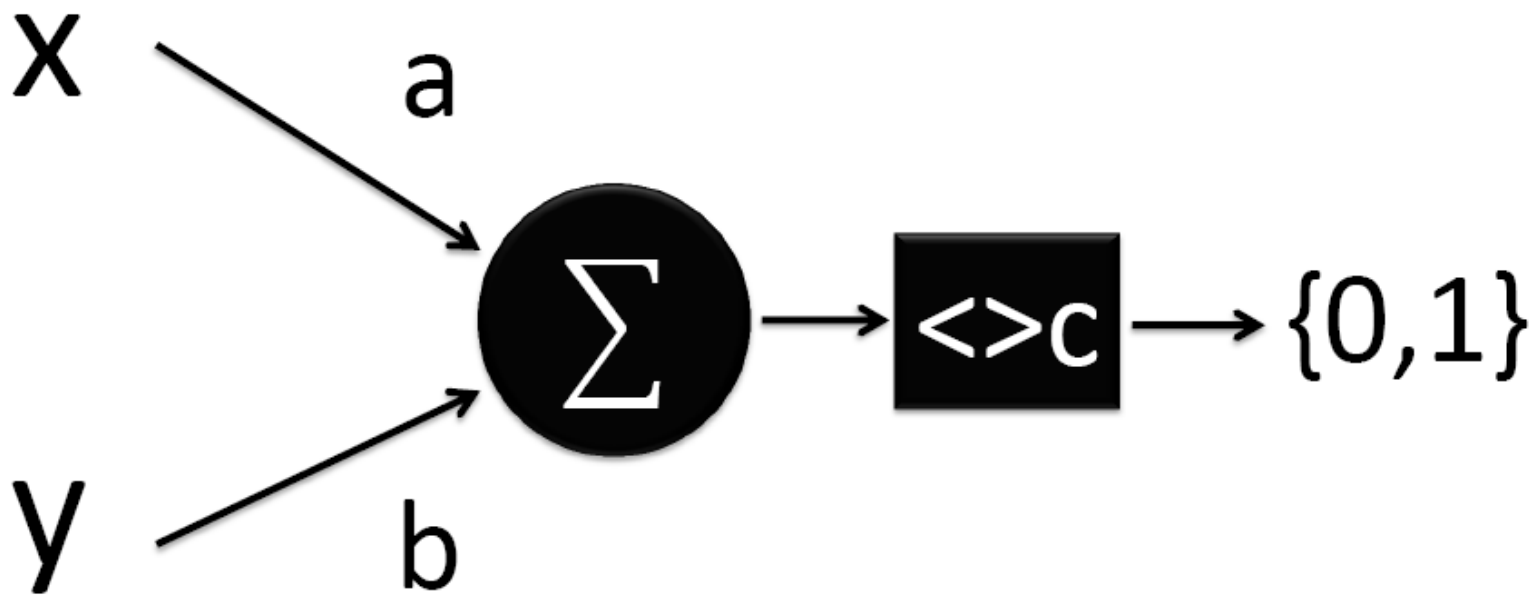
Часть 9

Идеи, взятые из биологии

Перцептрон

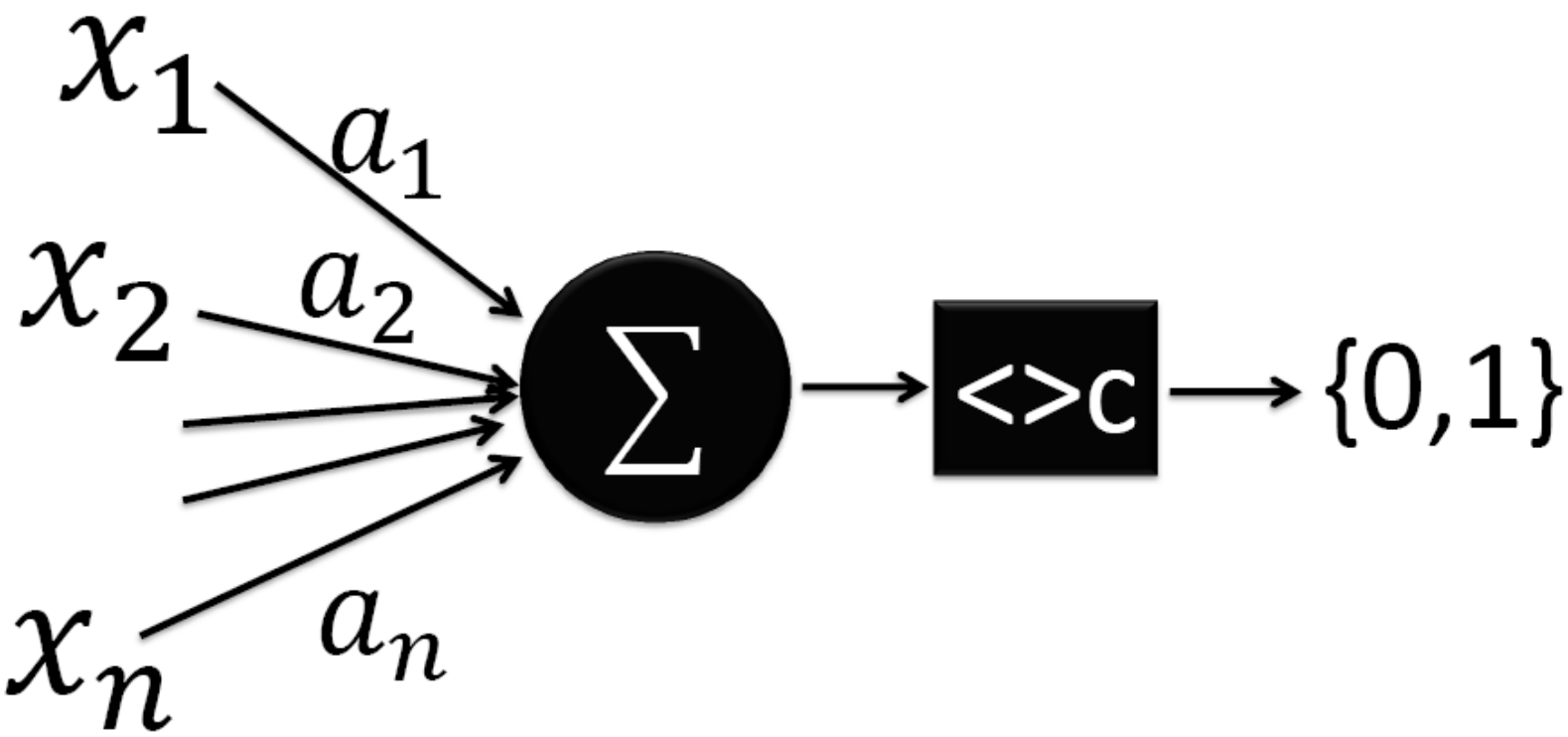
- Перцептрон – модель нервной клетки (нейрона)
- Каждый вход (дендрит) имеет свой вес
- Перцептрон возбуждается при сумме входных сигналов (дендритов) более порога возбуждения
- При двух входах: $a1*x + a2*y > c$ – уравнение полуплоскости
- На выходе (аксоне) – 0 или 1

Представим в виде схемы



a1*x1 + a2*x2 > c

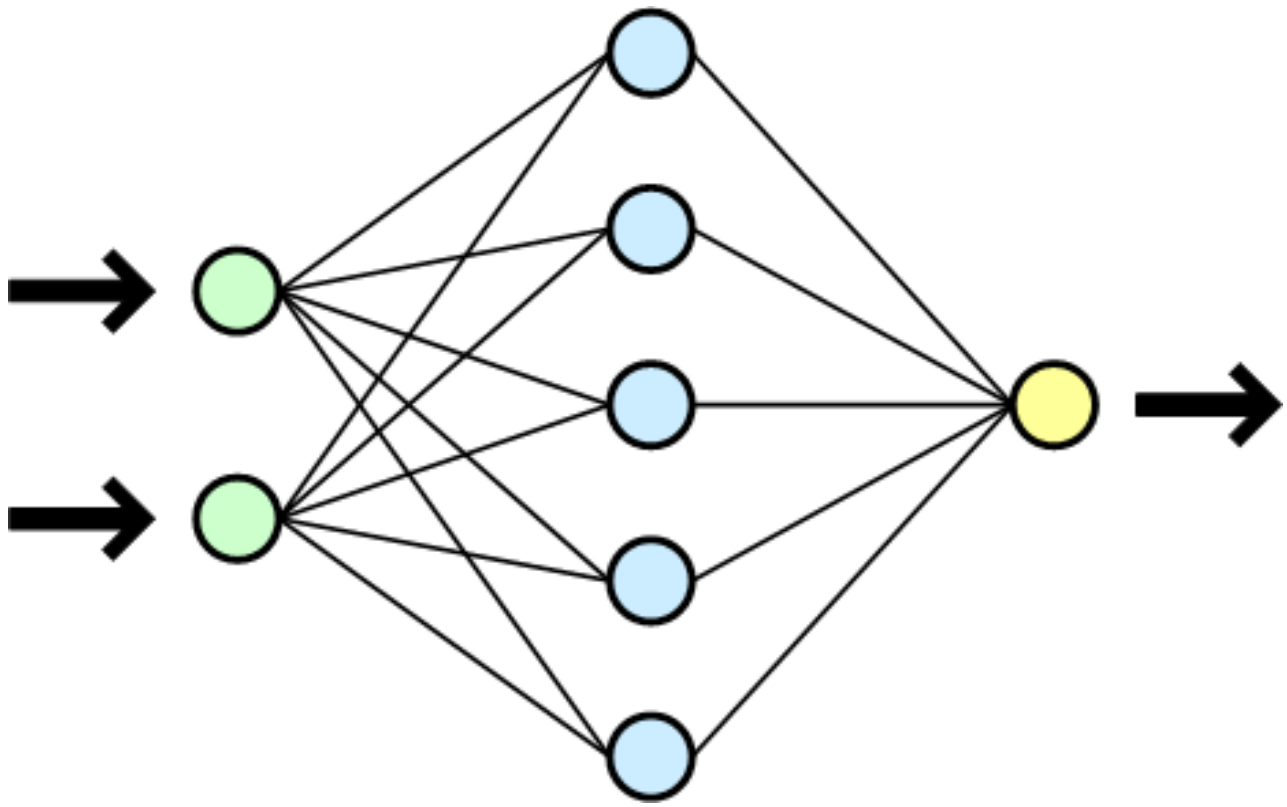
Аналогично для n входов

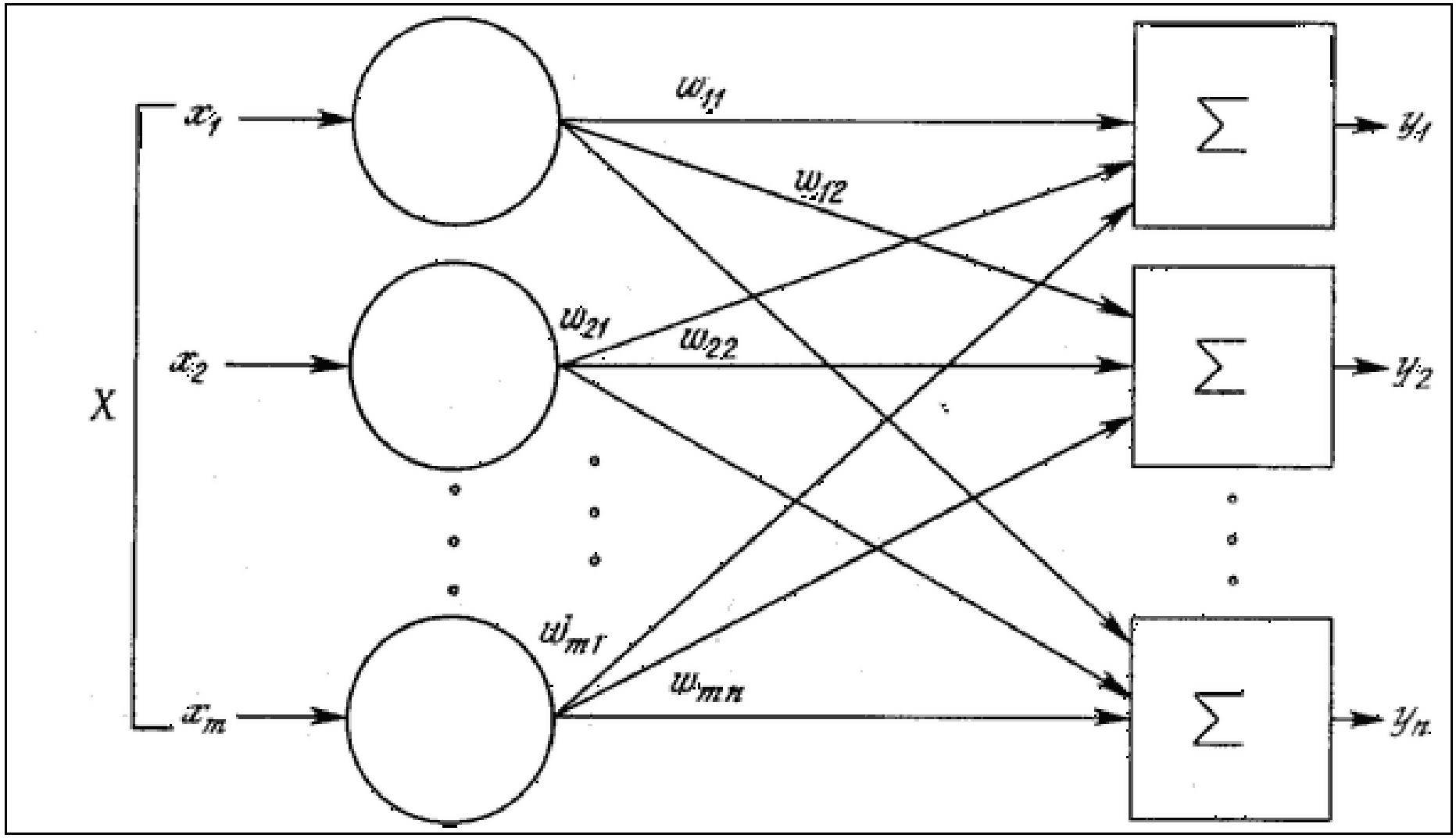


a1*x1 + a2*x2 + ... + an*xn > c

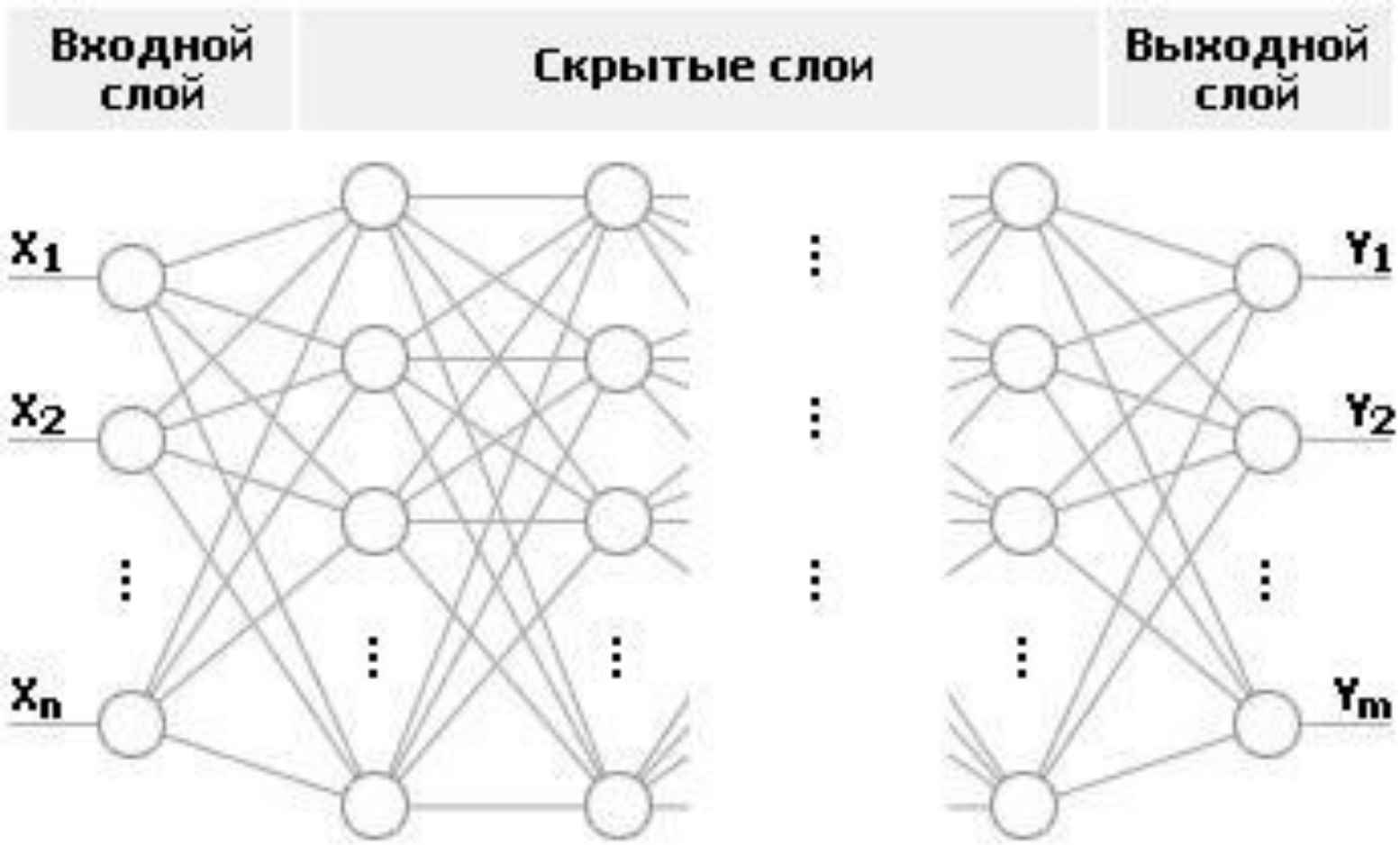
Модель нейросети

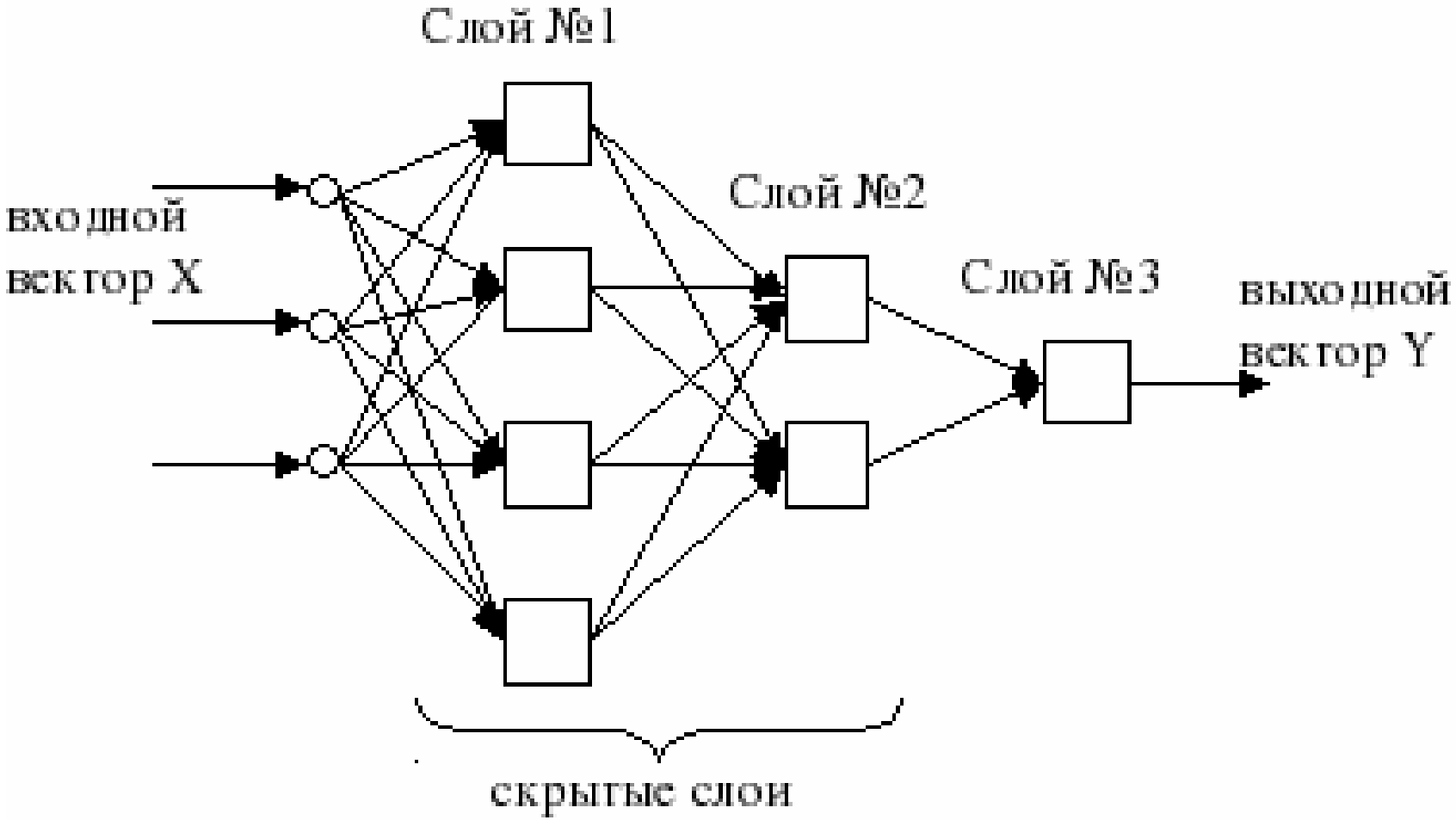
- Перцептроны можно объединять в слои
- Между всеми входными параметрами и всеми перцептронами первого слоя есть связи с некоторым весом (wij)

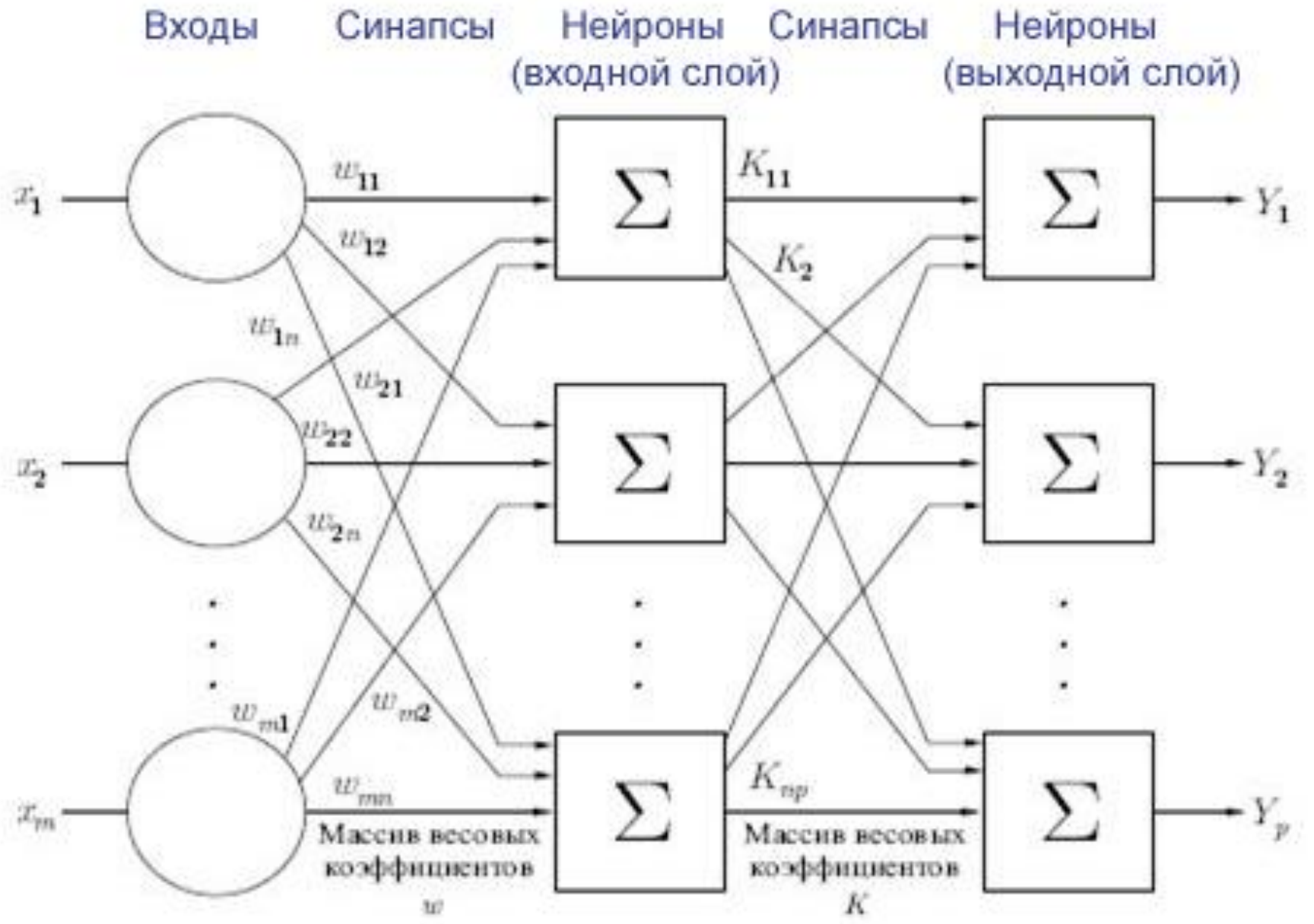



X
x_1
x_2
x_m
w_{11}
w_{12}
w_{21}
w_{22}
w_{m1}
w_{mn}
Σ
Σ
Σ
y_1
y_2
y_n

Слоев может быть много




Слой №1
Слой №2
Слой №3
входной
вектор X
выходной
вектор Y
скрытые слои

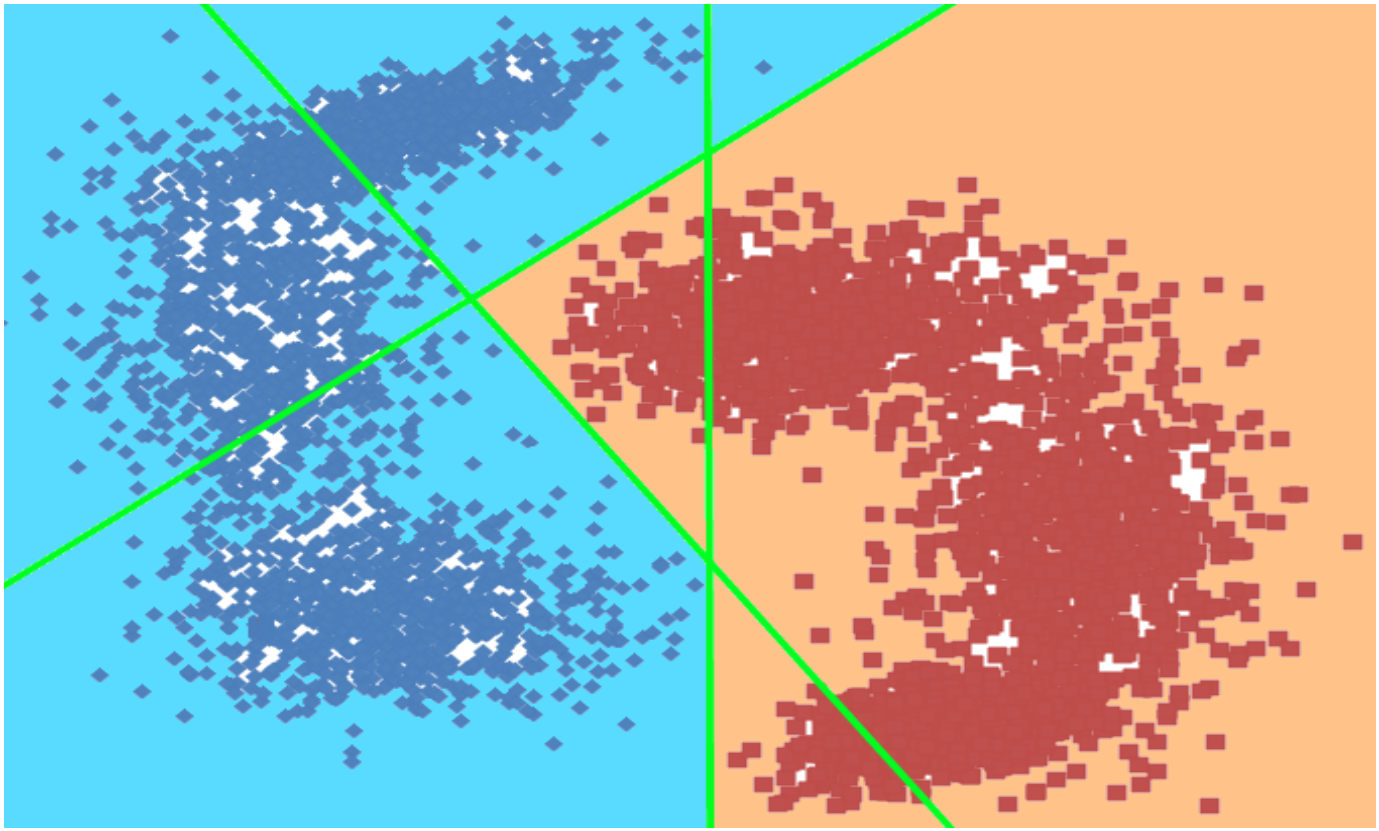

Входы
Синапсы
Нейроны (входной слой)
Синапсы
Нейроны (выходной слой)
x_1
x_2
x_m
w_{11}
w_{12}
w_{1n}
w_{21}
w_{22}
w_{2n}
w_{m1}
w_{m2}
w_{mn}
K_{11}
K_2
K_{np}
Y_1
Y_2
Y_p
Массив весовых коэффициентов w
Массив весовых коэффициентов K

Трехслойная нейросеть может обучиться до 100% точности на обучающей выборке

Доказательство:

- Первый слой разбивает пространство признаков гиперплоскостями на отдельные непересекающиеся области.
- Подбираем перцептроны первого слоя так, чтобы в каждую область попадали объекты только одного класса.

Пример разбиения пространства 3 перцептронами



- На выходе первого слоя мы получаем вектор из нулей и единиц. Это некоторая вершина единичного гиперкуба.
- Каждая вершина гиперкуба у нас соответствует объектам только одного класса.
- Перцептроны второго слоя разрезают гиперкуб гиперплоскостями.
- Подбираем перцептроны второго слоя так, чтобы отрезать от куба вершины только класса 1.
- Если нам на третий слой пришел нулевой вектор – значит мы имеем дело с вершиной класса 0. В любом другом случае – с вершиной класса 1.

- Поэтому перцептрон третьего слоя просто суммирует все выходы второго слоя с положительными коэффициентами.
- Математическое доказательство закончено. Ура.

Выбор конфигурации нейросети

- Каждый уровень в оптимальном случае разделяет объекты предыдущего на некоторые сущности
- Количество нейронов в каждом уровне должно быть не меньше различных сущностей на данном уровне
- Сколько выбрать уровней?

В человеческом мозгу

- В человеческом мозге скорость распространения сигнала около 2 м/с (а по телу – до 120 м/с)
- Скорость реакции человека – 0.2 секунды
- Средняя длина аксона – 1 см
- Количество уровней, которое проходит сигнал, примерно равно (T*2)/0.01 = T*50
- Минимальное количество уровней равно 10
- Однако в компьютерных нейросетях часто более эфективно меньшее количество уровней

Механизмы обучения

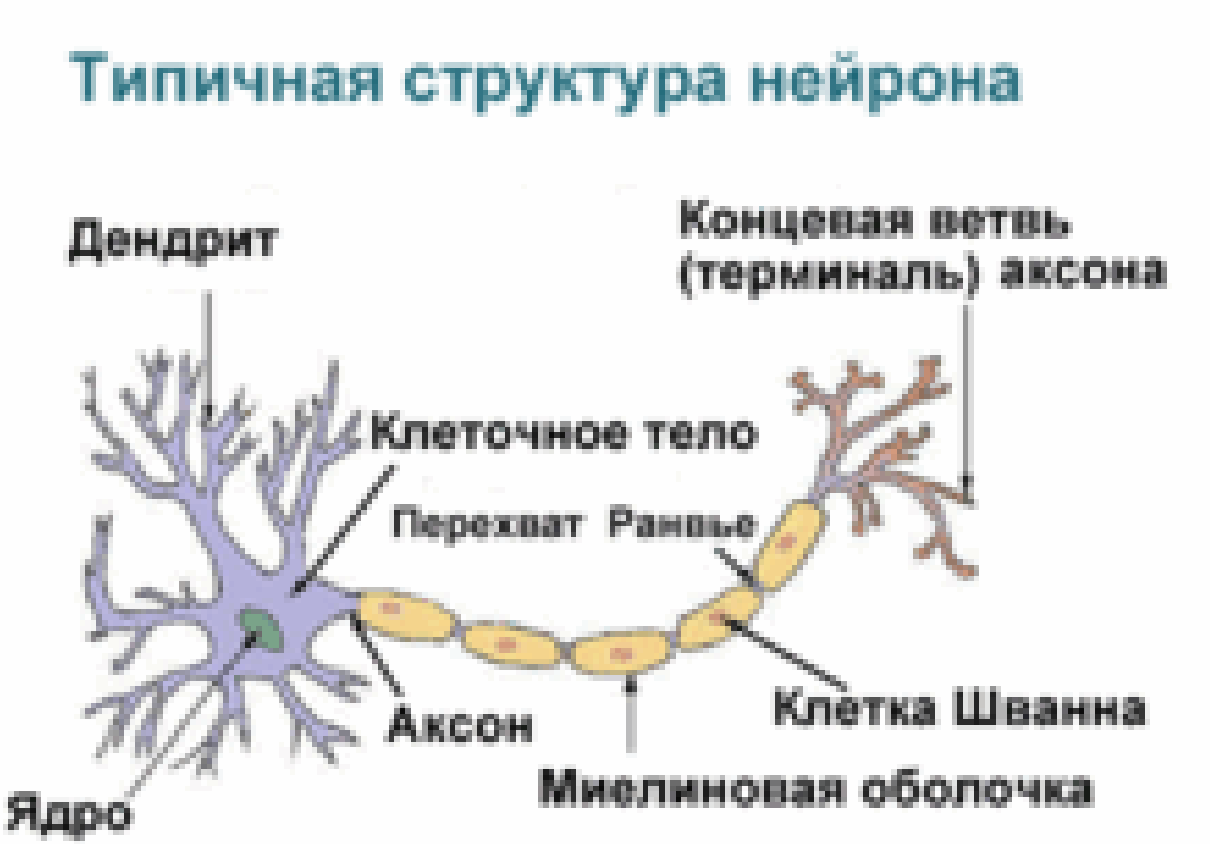
- Нейросеть однозначно задается матрицей (или матрицами) весов
- От нейросети есть функция оценки – среднеквадратичное отклонение результата от ожидаемого
- На этих матрицах запускаются другие алгоритмы машинного обучения: генетический алгоритм, градиентный спуск, метод отжига...
- Метод обратного распространения ошибки – вариация на тему градиентного спуска

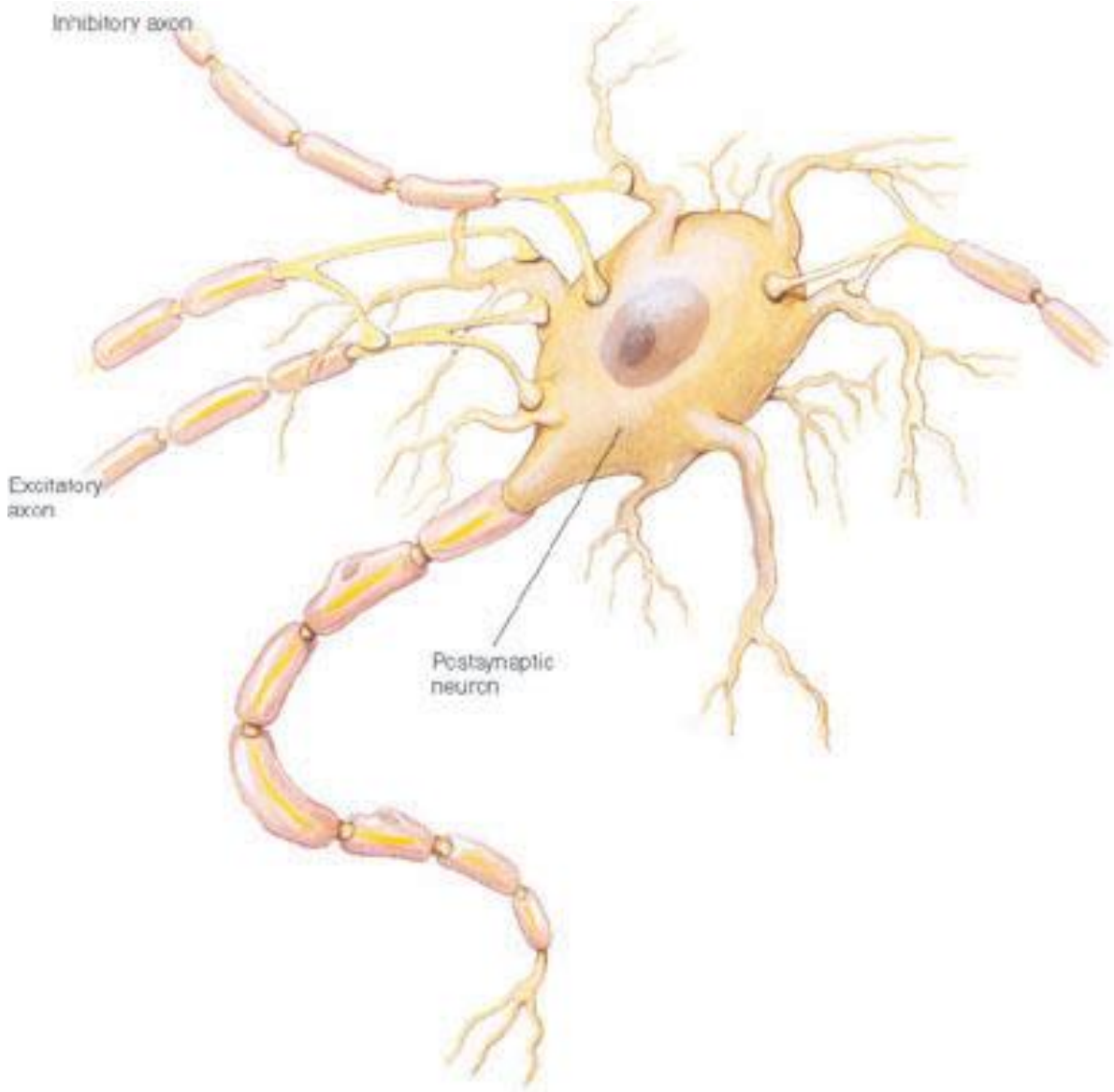
Недостатки нейронных сетей

- Алгоритм обучения может застопориться (так называемый «паралич» сети)
- Нейронная сеть часто очень сильно переобучается
- Имеет мало общего с реальными нейросетями в человеческом мозгу
- Нелинейность пространственно-временной суммации (особенно существенна для сигналов, приходящих по возбуждающим и тормозящим синапсам); временные задержки; эффекты синхронизации и частотной модуляции; рефлекторность

Нейробиология

- Нейрон – не перцептрон



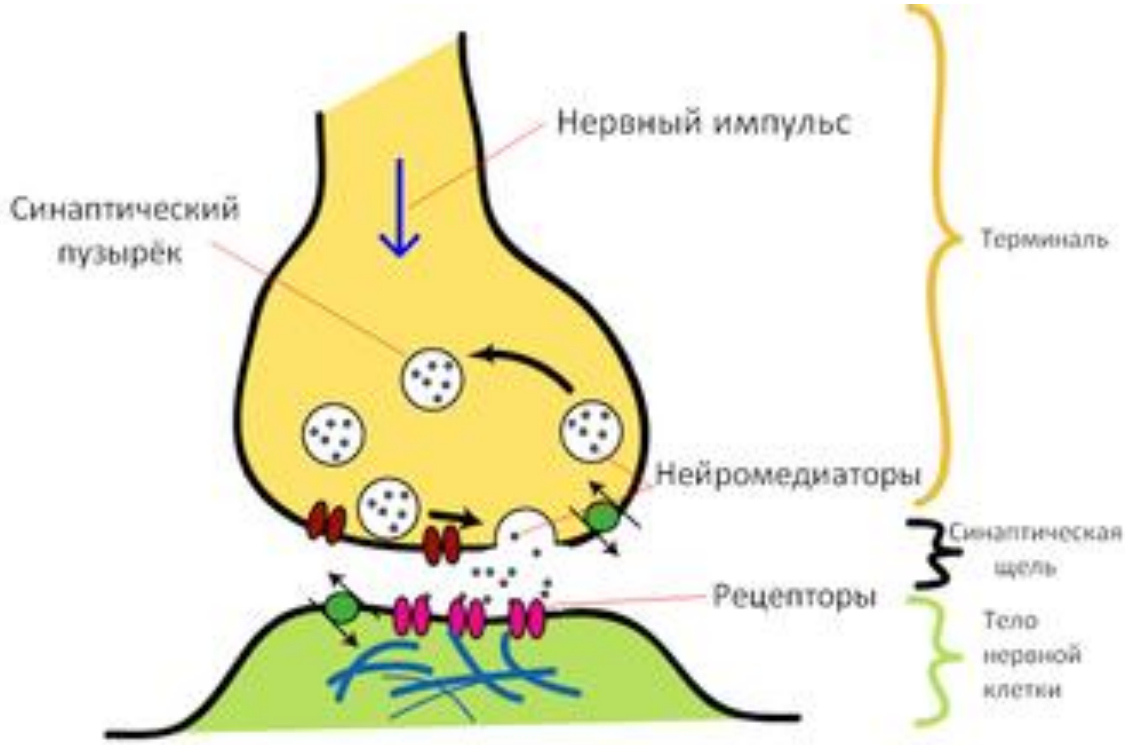

Inhibitory axon
Excitatory axon
Postsynaptic neuron

В чем отличие биологических нейросетей от компьютерных?

- Нейроны появляются в процессе жизни (нейрогенез), на этом, например, основана долговременная память
- У каждого нейрона ограниченное число выходов, которые геометрически расположены недалеко
- Нейроны могут мигрировать
- Расположение нейронов в пространстве влияет на работу и обучение нейросети
- В реальных нейросетях встречаются циклы
- …

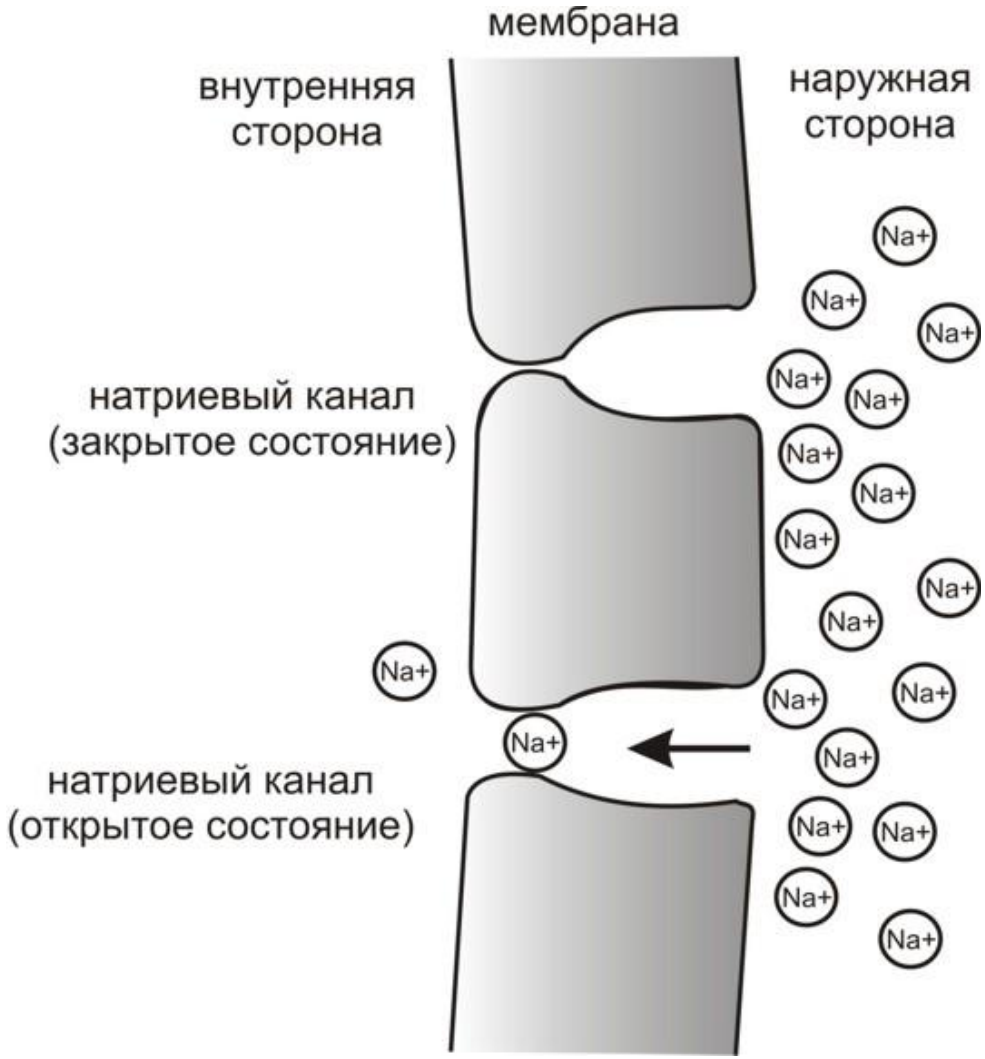
Синапс

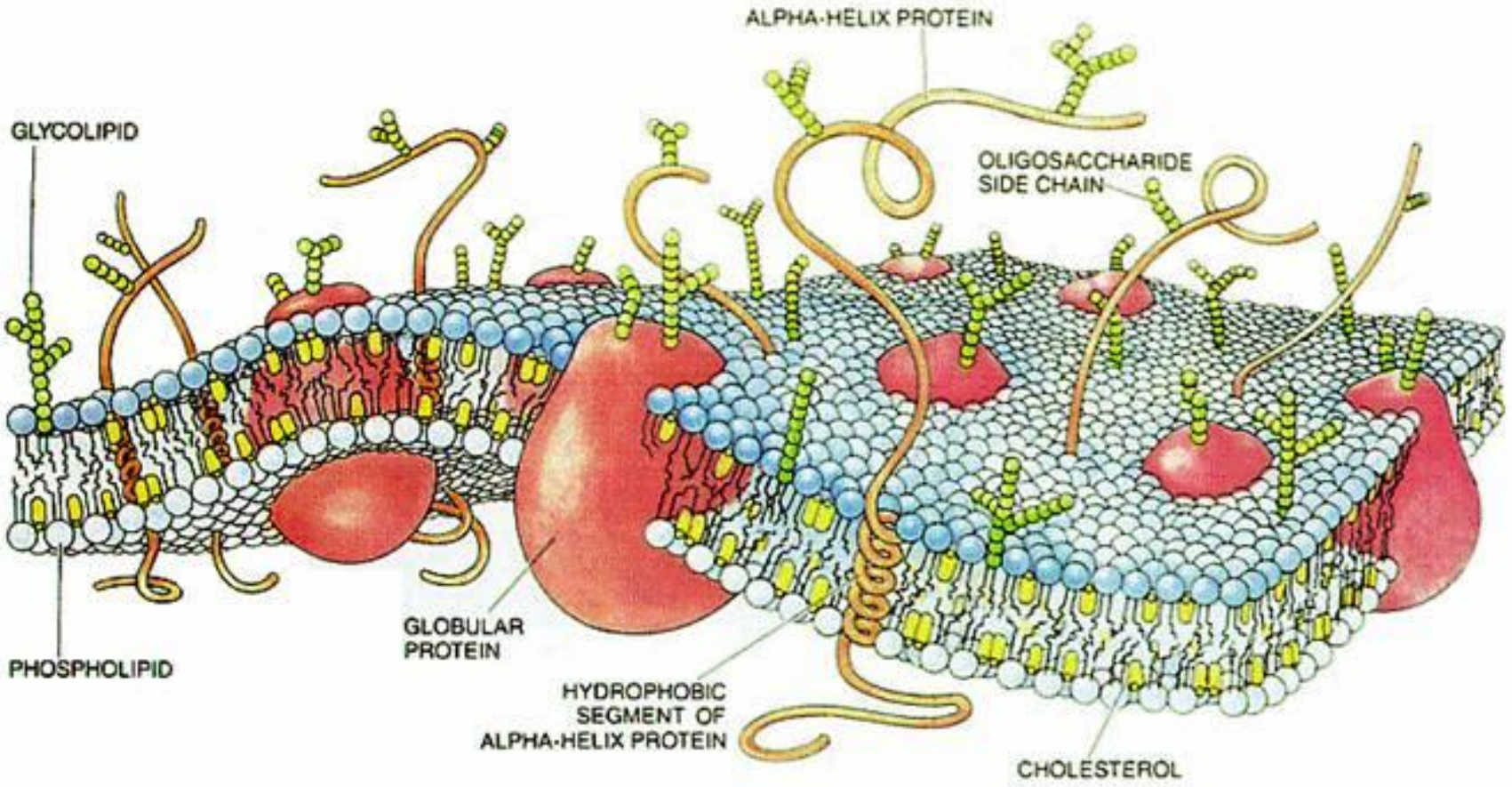
Между нейронами сигнал передается с помощью химических веществ, выбрасываемых с аксона в мозговую жидкость и воспринимаемых рецепторами на дендрите



Механизм возбуждения нейрона

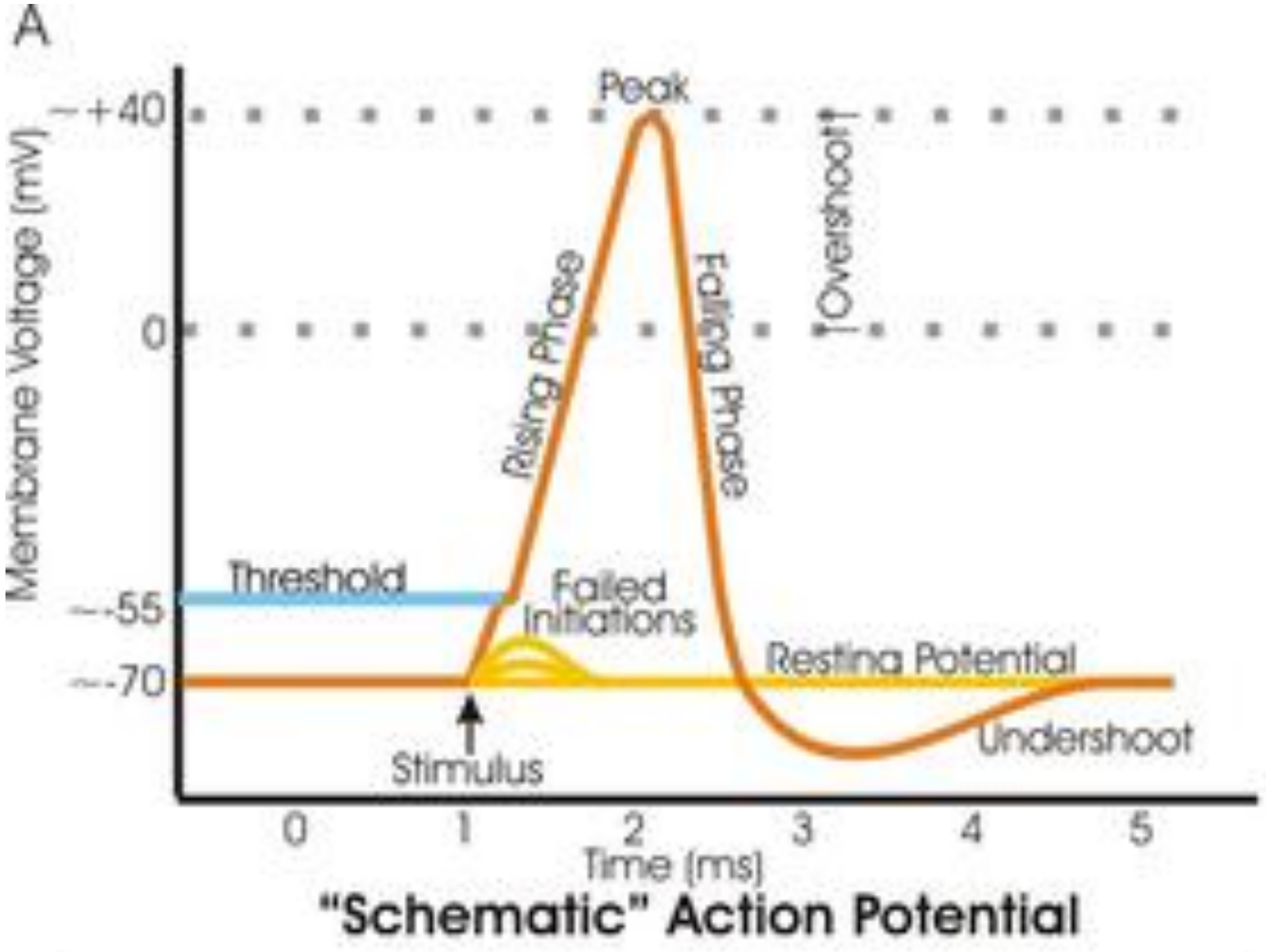
- Нейрон – клетка, у него есть мембрана
- На мембране живой клетки есть разность потенциалов внутри и снаружи – поляризация (из-за избирательной проницаемости)


мембрана
внутренняя сторона
наружная сторона
натриевый канал (закрытое состояние)
натриевый канал (открытое состояние)
Na+

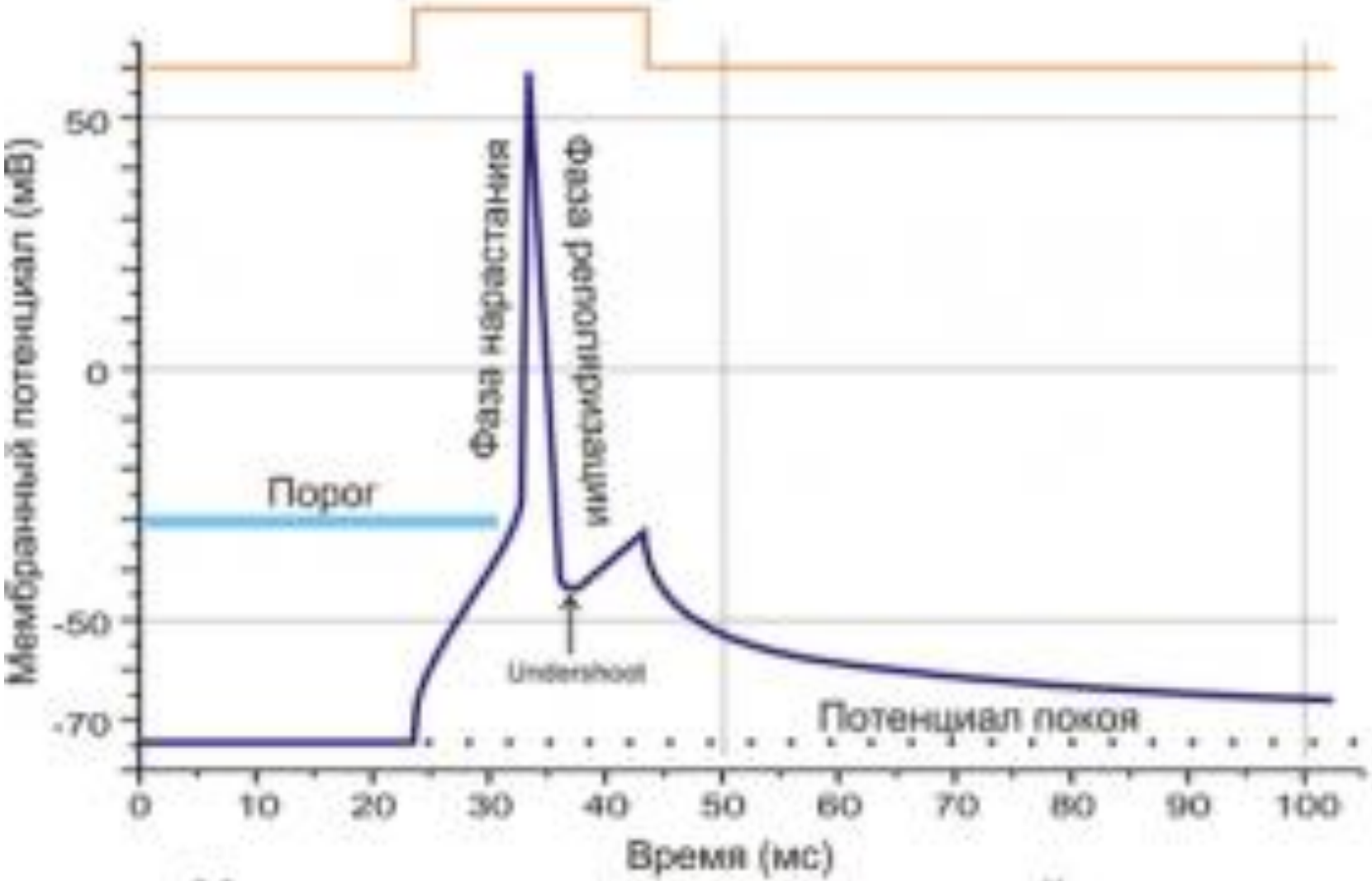

ALPHA-HELIX PROTEIN
GLYCOLIPID
OLIGOSACCHARIDE
SIDE CHAIN
GLOBULAR
PROTEIN
PHOSPHOLIPID
HYDROPHOBIC
SEGMENT OF
ALPHA-HELIX PROTEIN
CHOLESTEROL

Механизм возбуждения нейрона

- Реполяризация – смена поляризации мембраны
- После возбуждения нейрона некоторое время он невосприимчив к внешним сигналам (гиперполяризация)
- После многократного возбуждения нейрон "устает" и возбуждается слабее



"Schematic" Action Potential



Ход реального потенциала действия

Высшая нервная деятельность, условные рефлексы

- Условный рефлекс (собака Павлова): возникает (укрепляется) ассоциация между действиями, которые часто повторяются одновременно
- Существуют ансамбли нейронов, связанные с какой-то одной сущностью
- Если ассоциации между ансамблями идут циклически, получается навязчивая мысль, которая вызывает сама себя

Высшая нервная деятельность

- В результате большой нагрузки, например, во время стресса, нейрон может умереть. Тогда вещества, стимулирующие возбуждение нейронов, выбрасываются в мозговую жидкость, что может вызвать цепную реакцию.
- ...

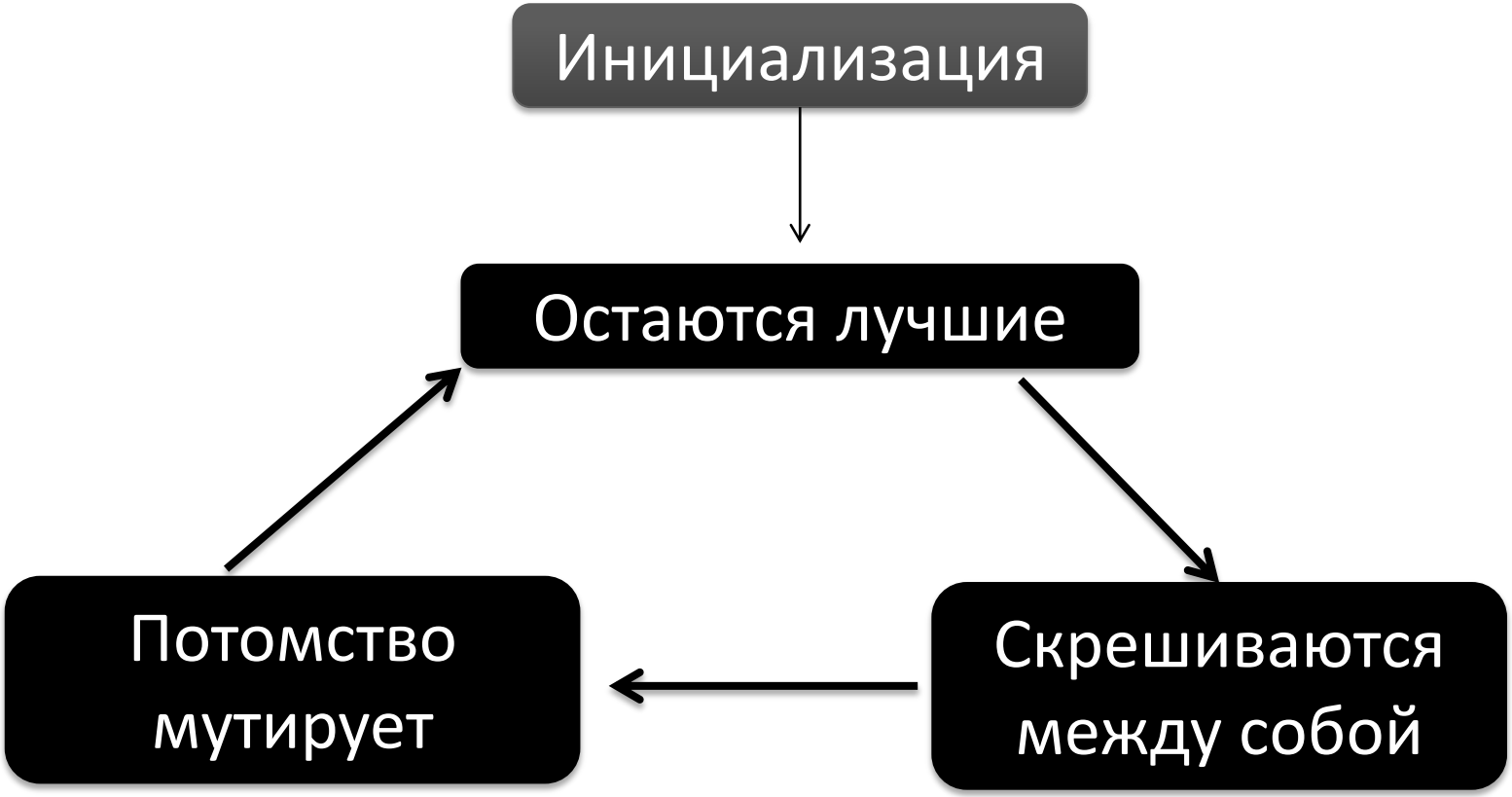
Почему сложно создать модель биологической нейросети

- Большое количество элементов (10-100 млрд. нейронов) в мозгу
- Необходимо моделировать движение мозговой жидкости, питательных веществ и медиаторов (химических веществ, передающих возбуждение) в ней
- Большое количество разнообразных химических соединений в мозгу
- Необходимо моделировать внутреннюю структуру нейронов
- Механизмы работы нейросетей ещё не изучены

Другой принцип из биологии: естественный отбор

- Слабые (плохие) индивиды умирают, не оставляя потомства
- Сильные (хорошие) индивиды выживают, скрещиваются, оставляют похожее на себя потомство

Генетический алгоритм



Генетическое построение машинок

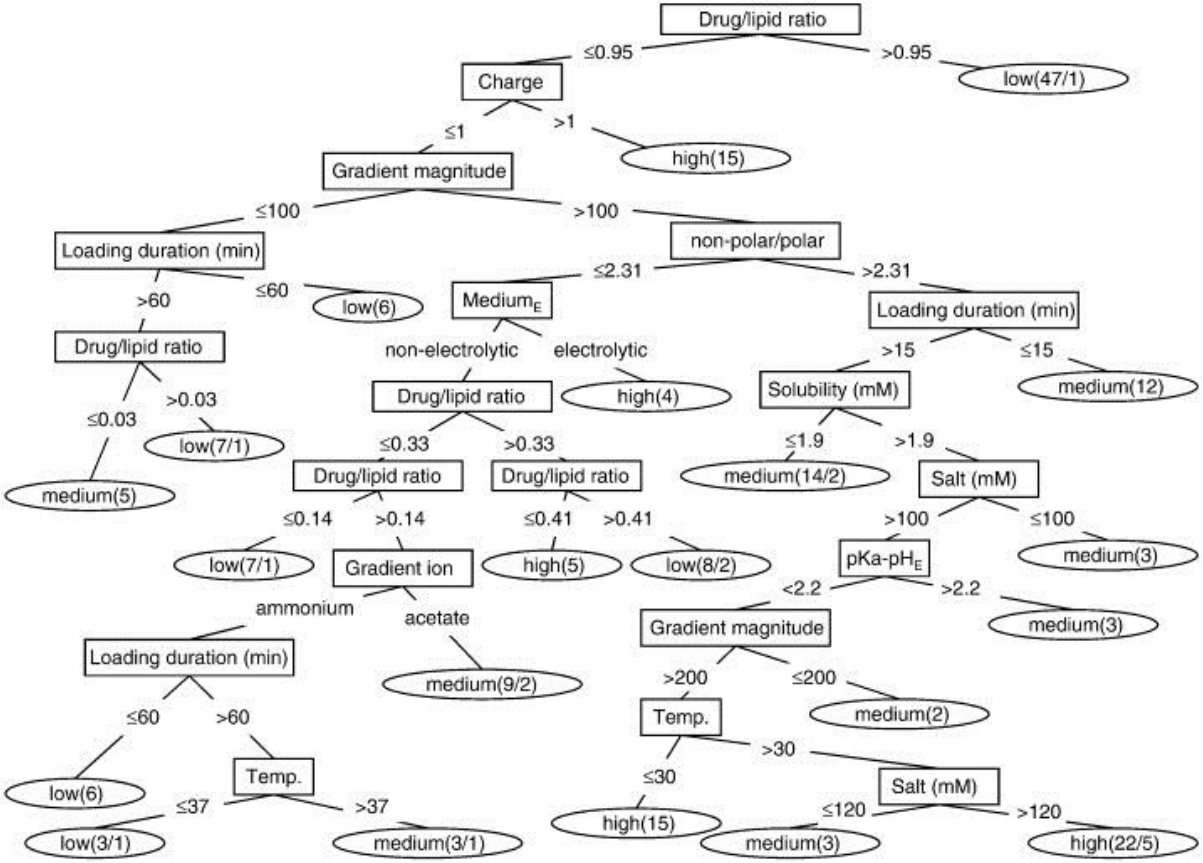
http://megaswf.com/file/1006490

Лекция 3

Часть 10

Решающие деревья и композиции алгоритмов

Экспертные системы, составленные вручную

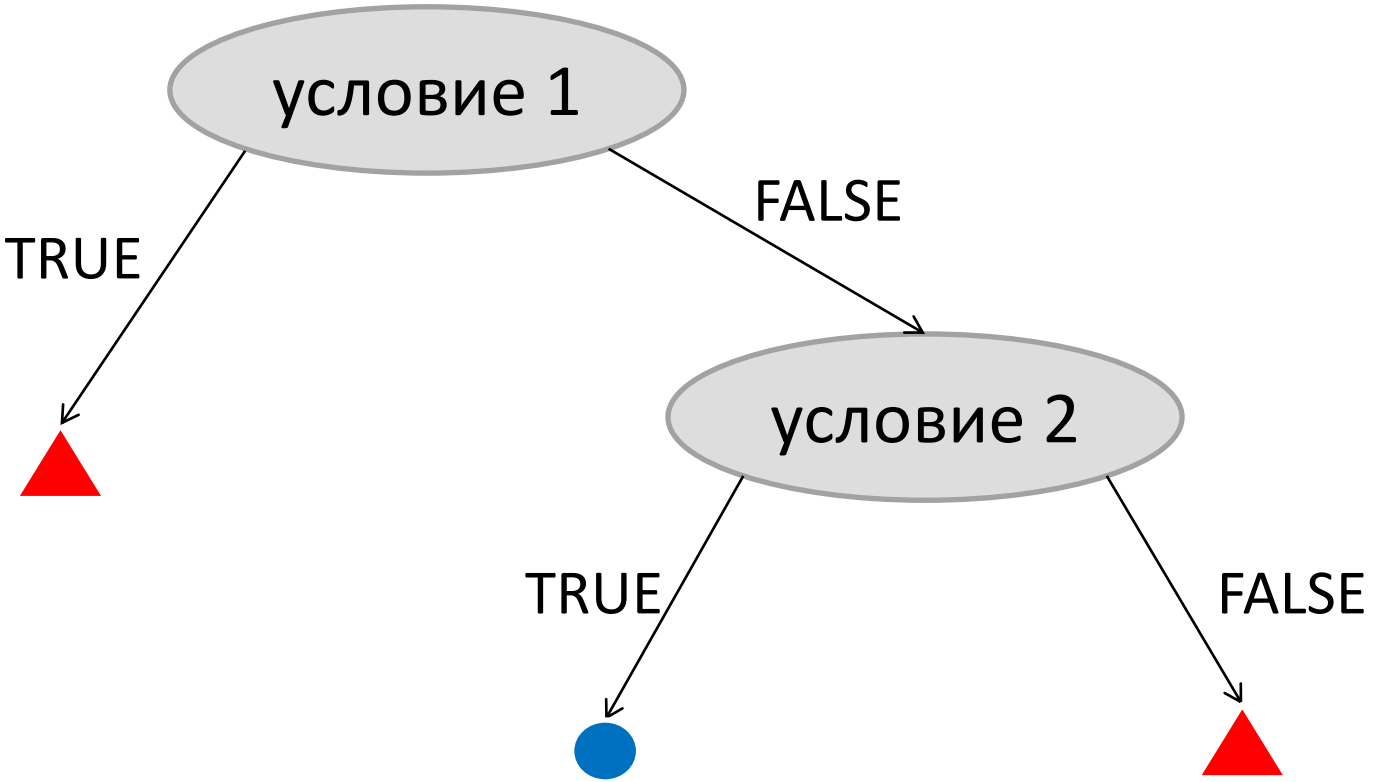


Решающее дерево для предсказания эффективности введения лекарства

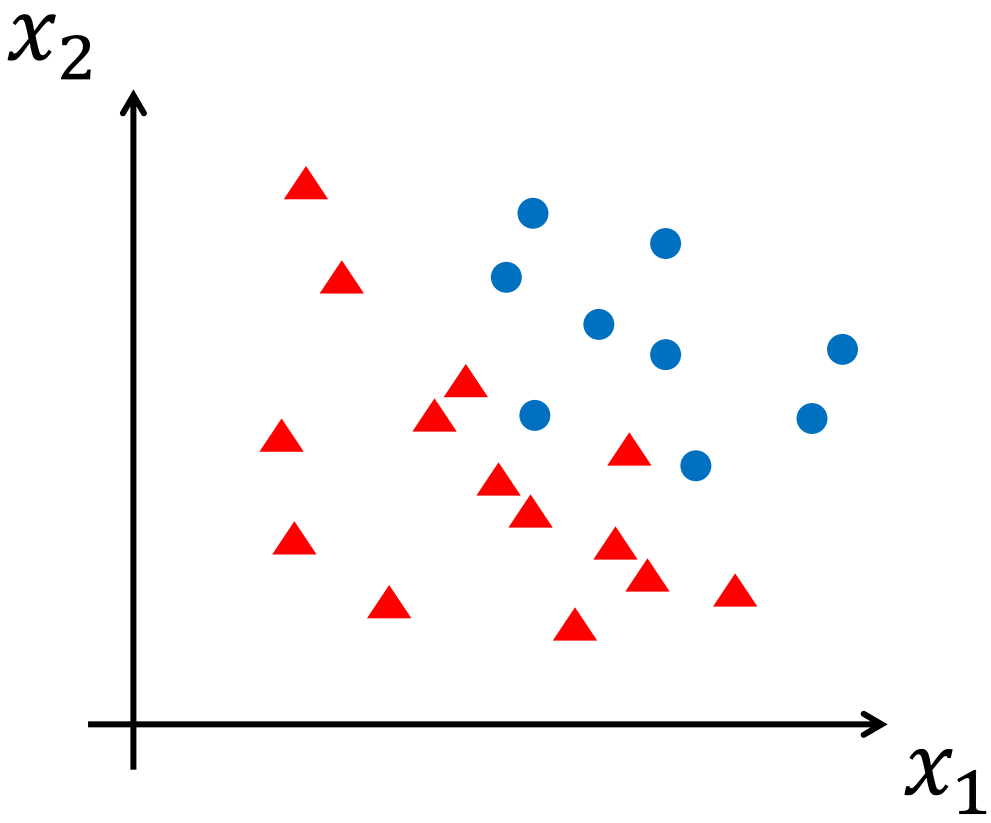
В чем проблемы?

- Правила составлялись вручную экспертами
- Мнения экспертов расходятся
- Эскперты могут ошибаться
- Эксперт не в состоянии проанализировать все данные

Построим дерево автоматически



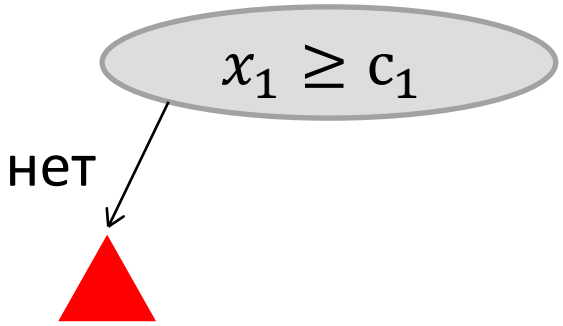
Какие условия будут в дереве?

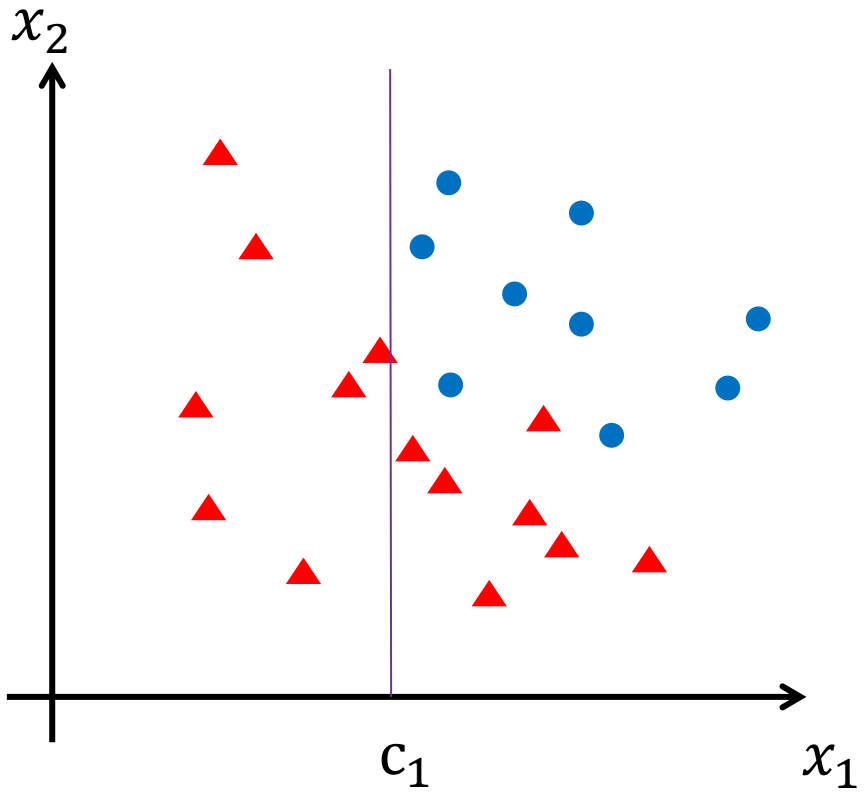


Попробуем использовать пороговые условия перехода в виде пороговых правил: x>c

Начнем строить дерево

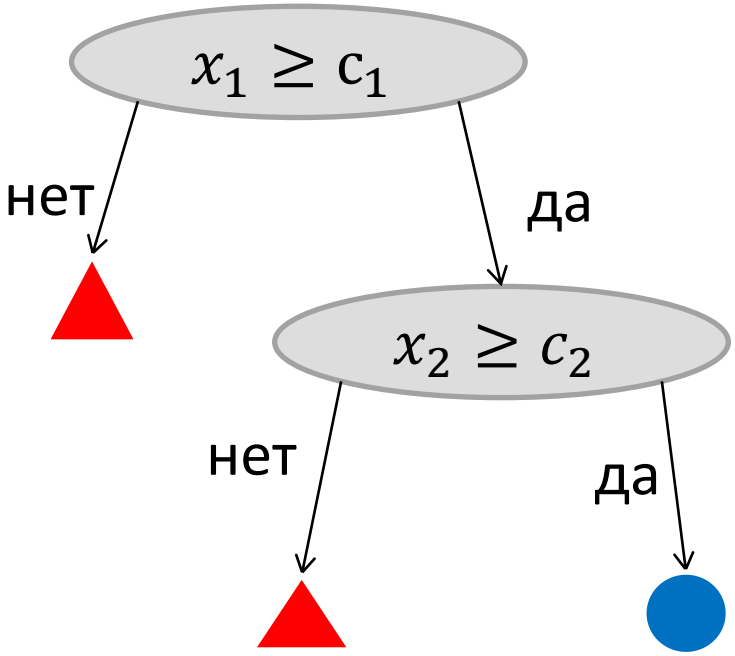
- Будем действовать жадно
- Каждый раз берем наиболее «информативное» разделение всей области

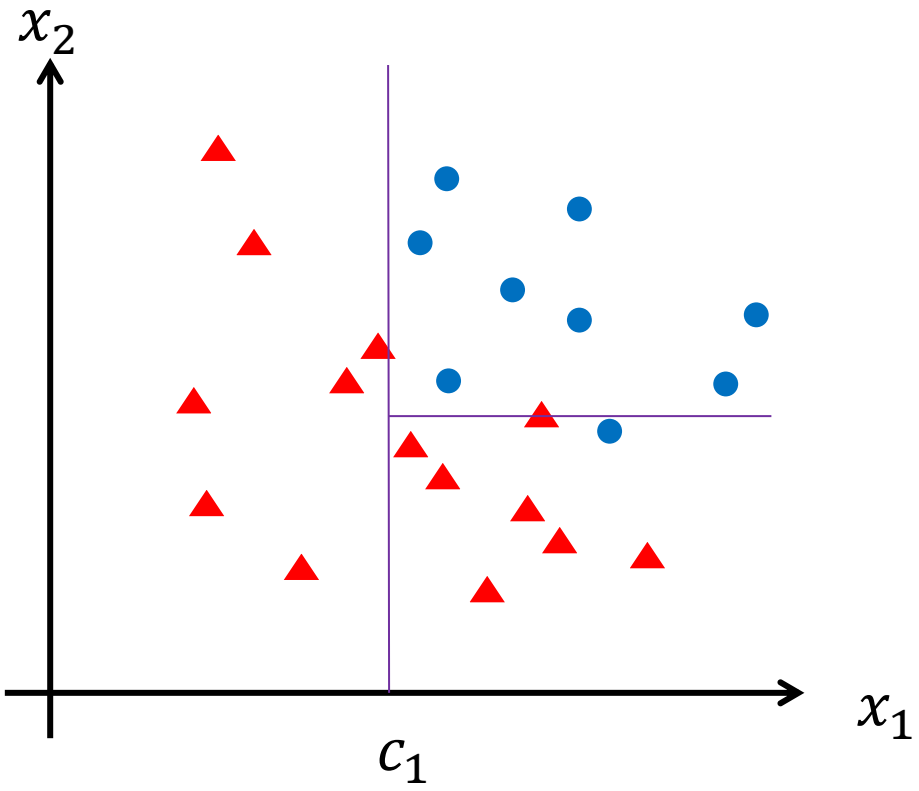




Строим дерево

Каждый раз берем наиболее «информативное» разделение текущей области





Как выбирать условия?

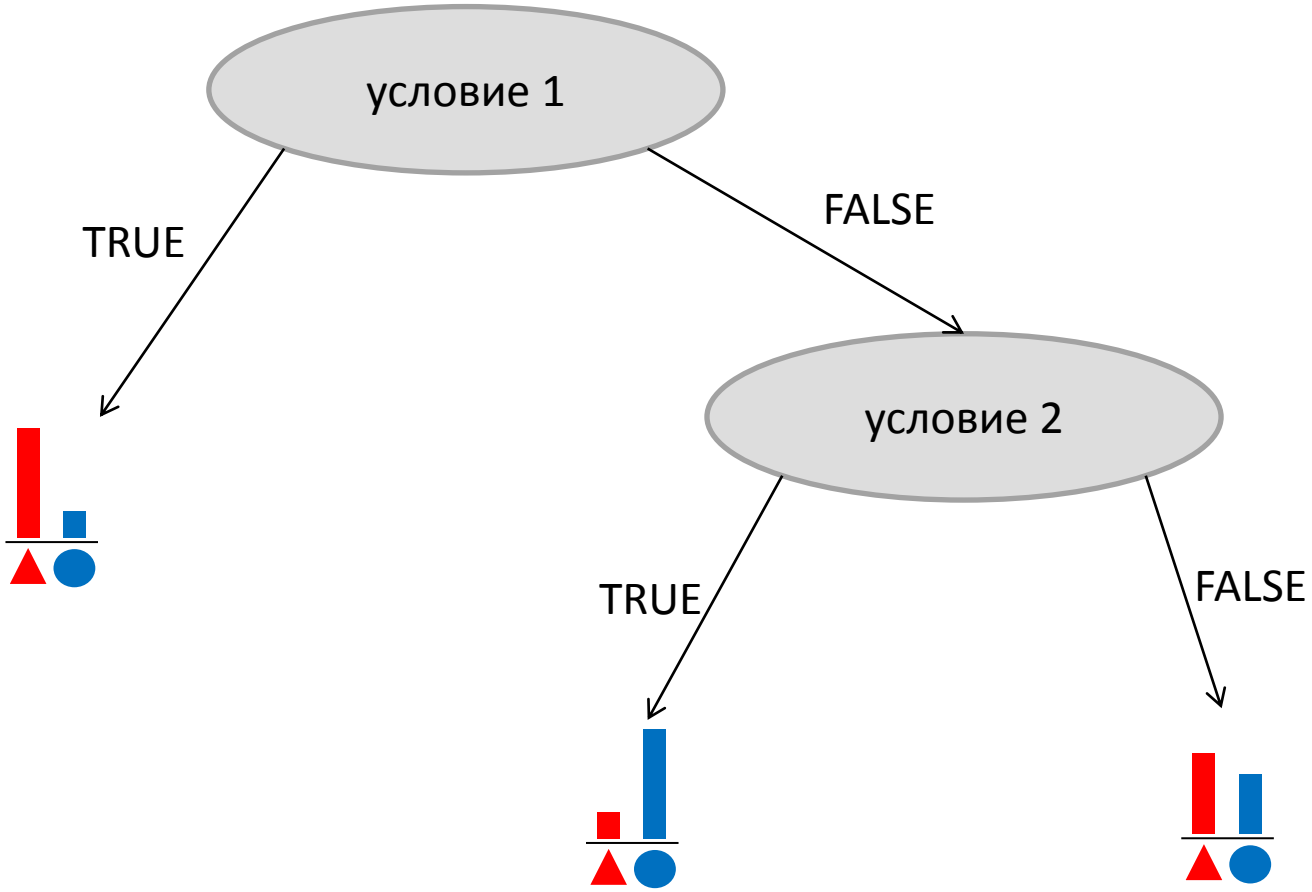
Перебираем признаки по очереди. Лучшее разбиение признака:

- Отделить один класс как можно сильнее
- Максимизируем число пар объектов, у которых одинаковый класс и одинаковый ответ на условие – критерий Джини
- Максимизируем число пар объектов, у которых разный класс и разные ответы на условие – критерий Донского
- Объединяем предыдущие
- Более сложные вероятностные соображения

Улучшения

- Если информативность условия меньше порога, то прекращаем строить дерево
- Разбиваем обучение на две части. Пробегаем по всем поддеревьям и заменяем их левым или правым потомком, если они допускают заметно меньше ошибок

Будем возвращать вещественную степень принадлежности классу от -∞ до +∞



Композиция алгоритмов

- Пусть есть какой-то набор из T алгоритмов:

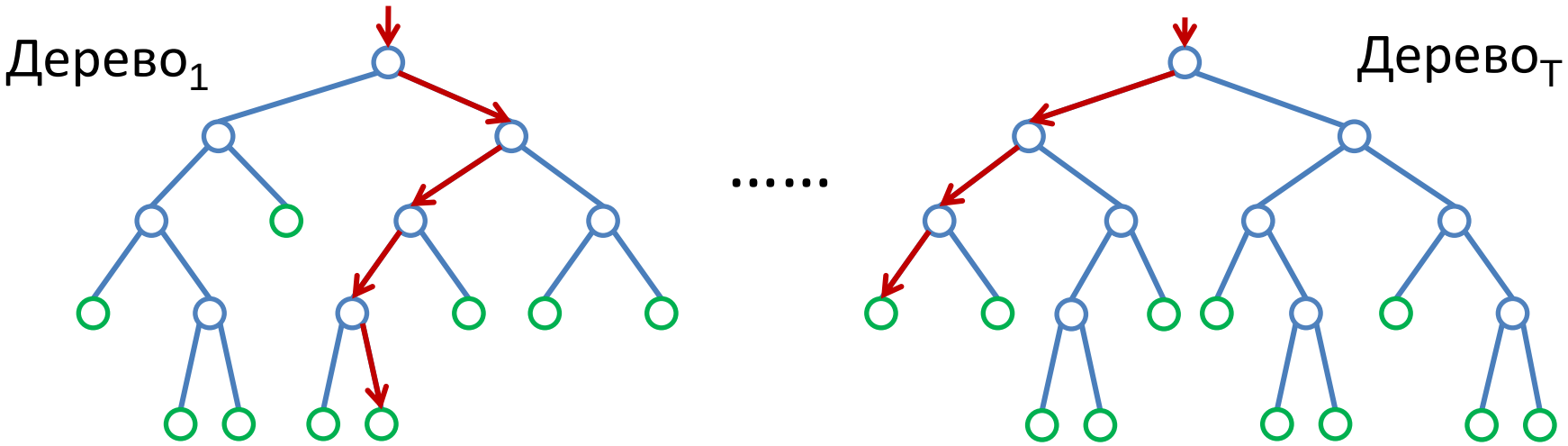
$$a_1, a_2, a_3, \dots, a_T$$

- Финальный алгоритм:

$$result := \frac{1}{T}\sum_{t=1}^{T} a_t$$

Случайный лес

Построим композицию из решающих деревьев

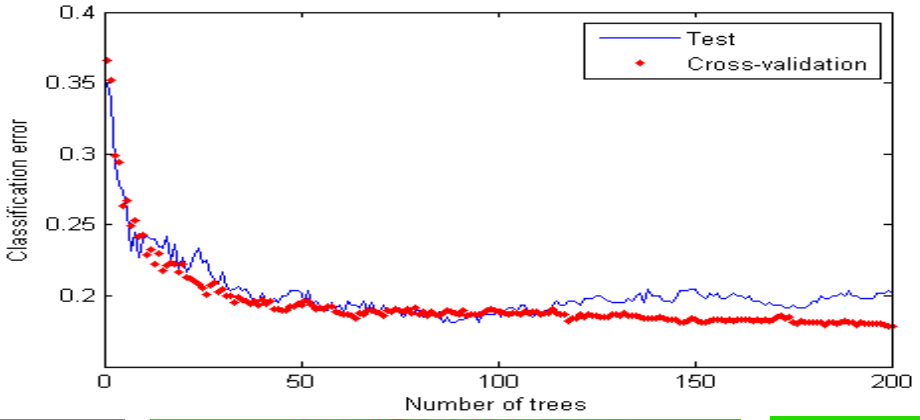


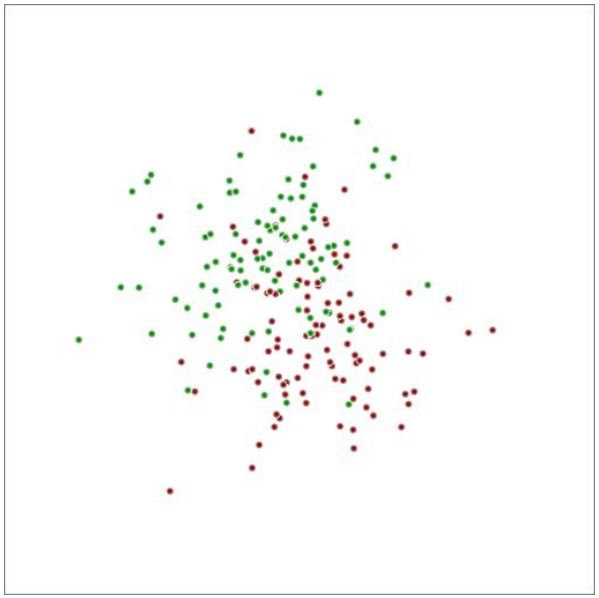
Как сделать деревья существенно разными?

Используем случайные подвыборки данных

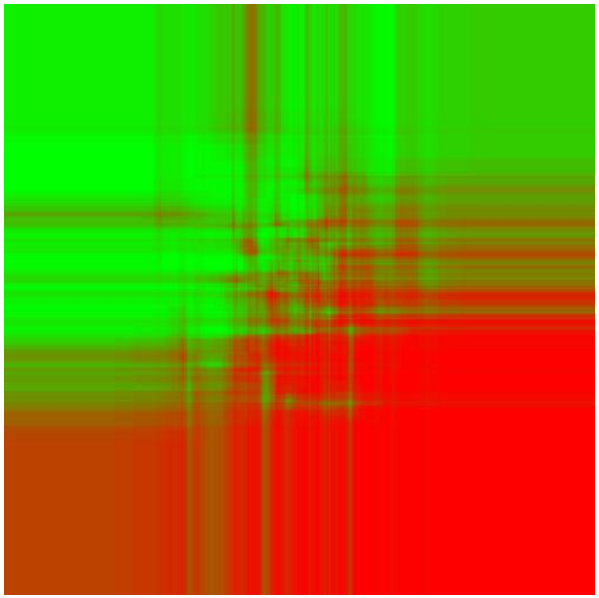
- Выберем произвольные объекты из обучающей выборки
- Обучимся по ним, получим первый алгоритм
- Опять выберем произвольные объекты (возможно, те же)
- ….
- Получаем набор деревьев, из них построим композицию

Как работает случайный лес?

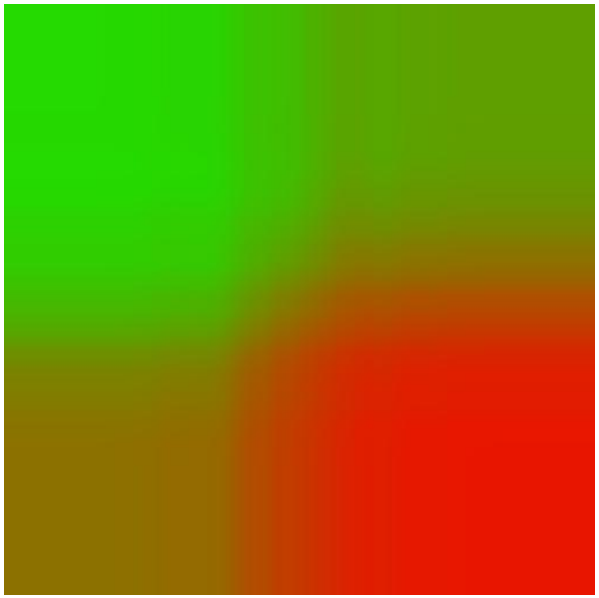




данные



Вероятностные выходы в признаковом пространстве



Очень много деревьев (2000)

Особенности случайного леса

- Работает с любыми видами признаков
- Общепризнанно лучший алгоритм классификации
- Тяжело интерпретируется человеком
- Долго строится

Подбор коэффициентов

- Финальный алгоритм:

$$result := \sum_{t=0}^{T} w_t a_t$$

- Это же уравнение гиперплоскости! Применим метод опорных векторов для новых признаков и найдем оптимальную плоскость

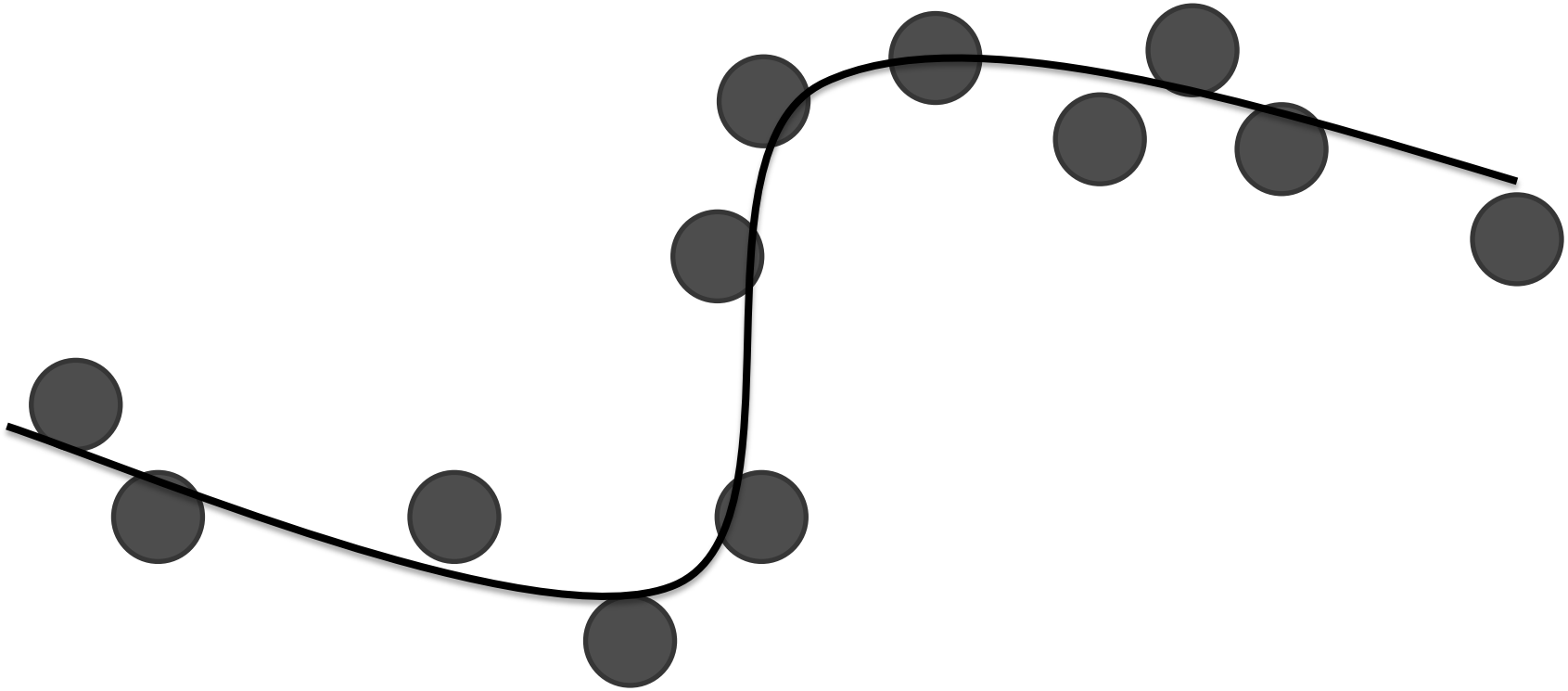
Умный подсчет весов и коэффициентов

- Произвольное выбрасывание объектов равносильно присваиванию объектам весов 0 и 1
- Можно считать веса умнее
- Строим алгоритмы по очереди так, чтобы следующий старался максимально исправить ошибки всех предыдущих

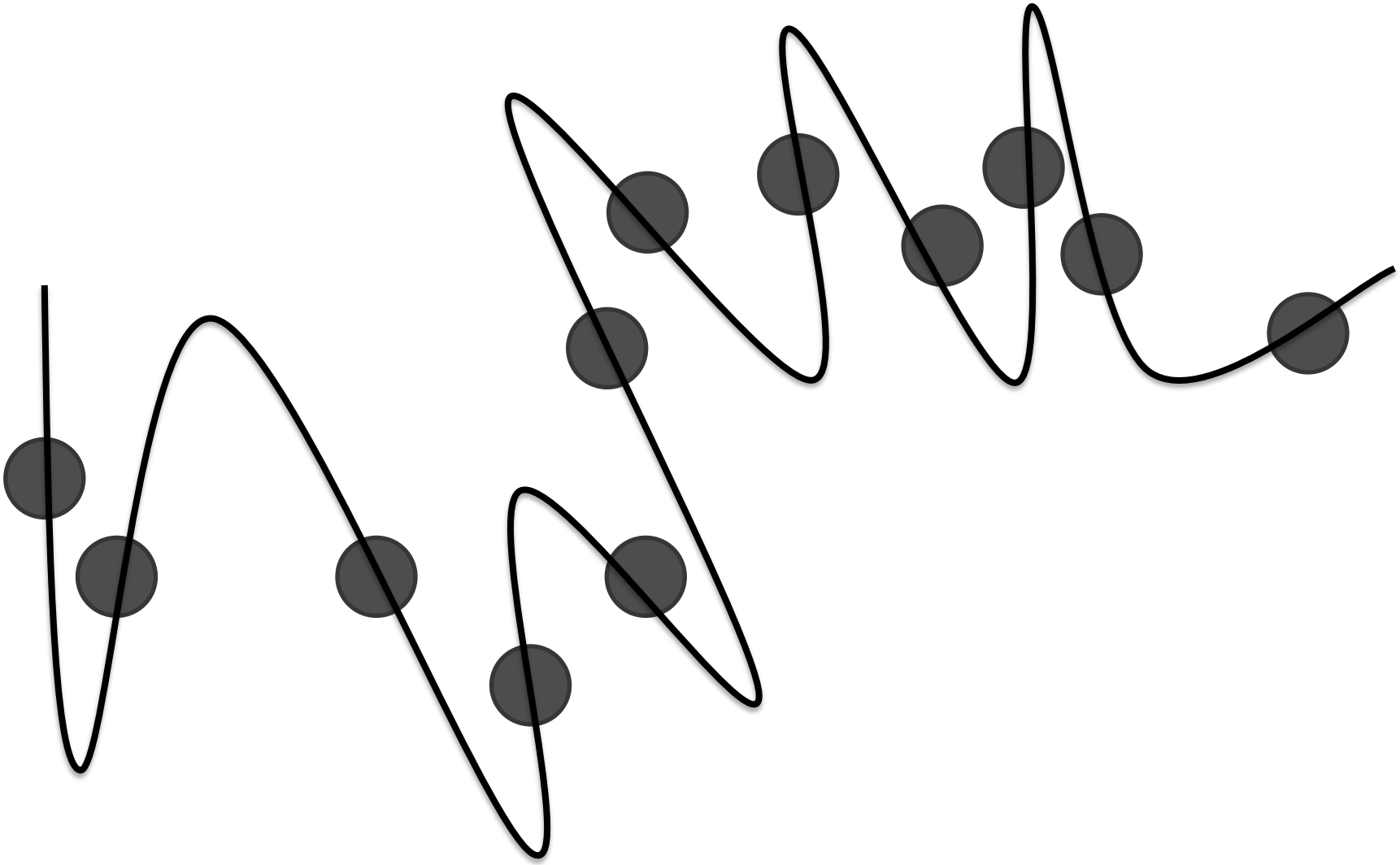
Часть 11

Регрессия

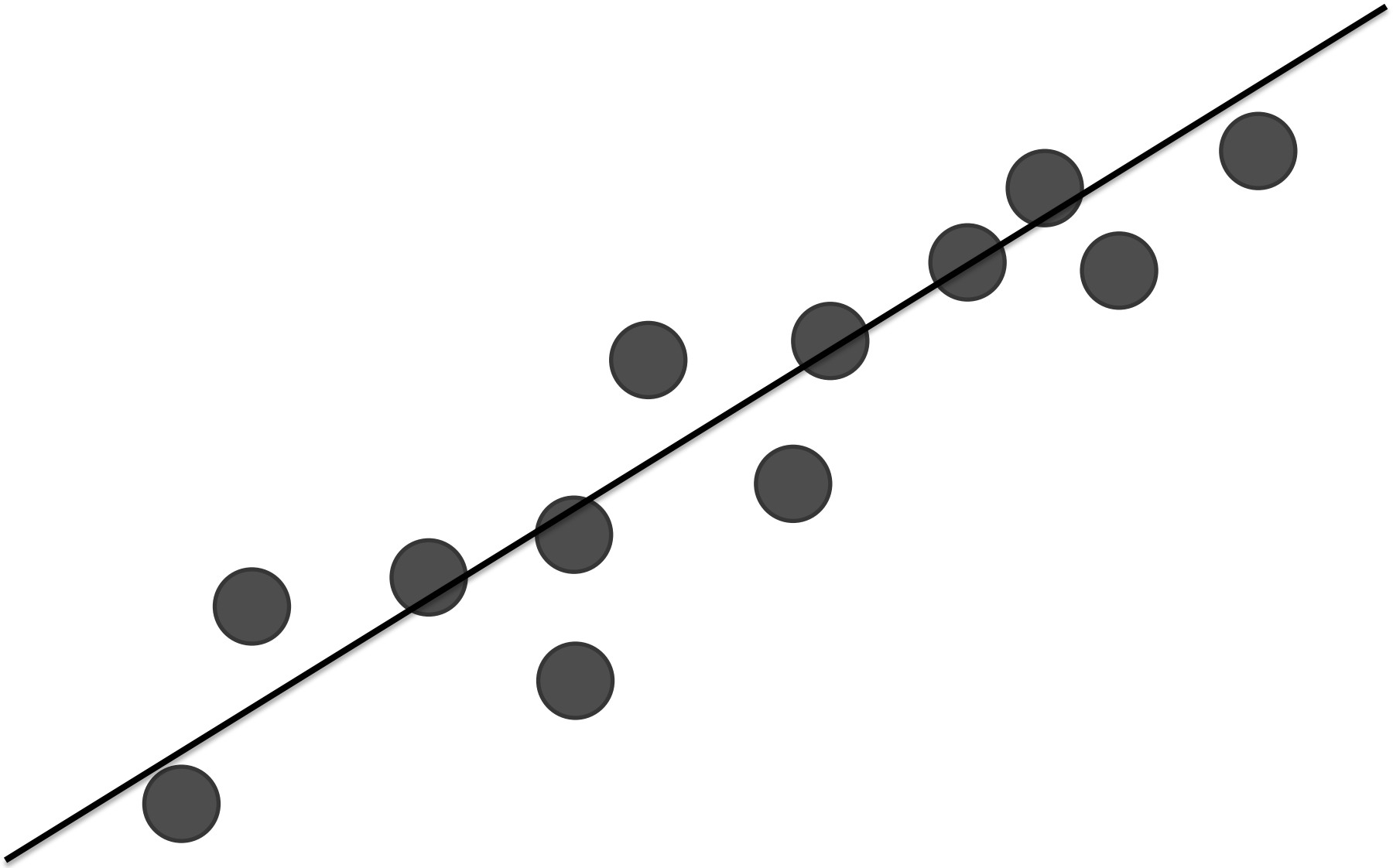
Хотим восстановить кривую



Переобучение в регрессии

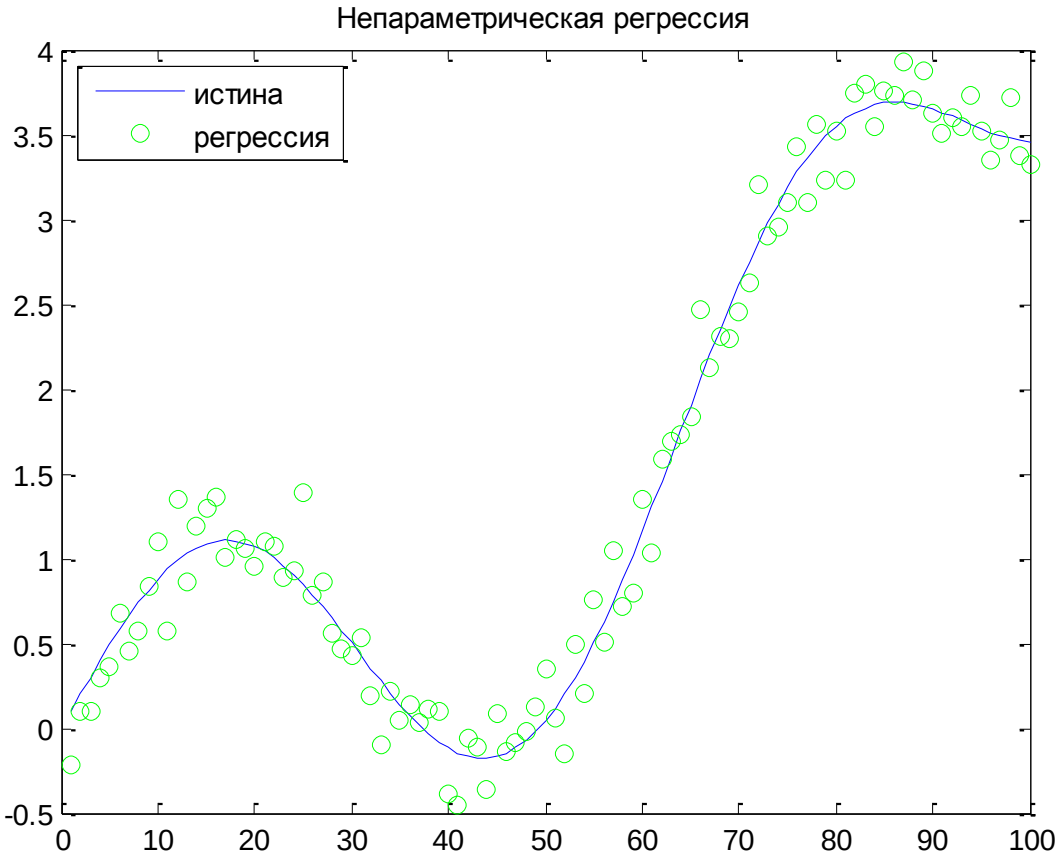


Линейная регрессия



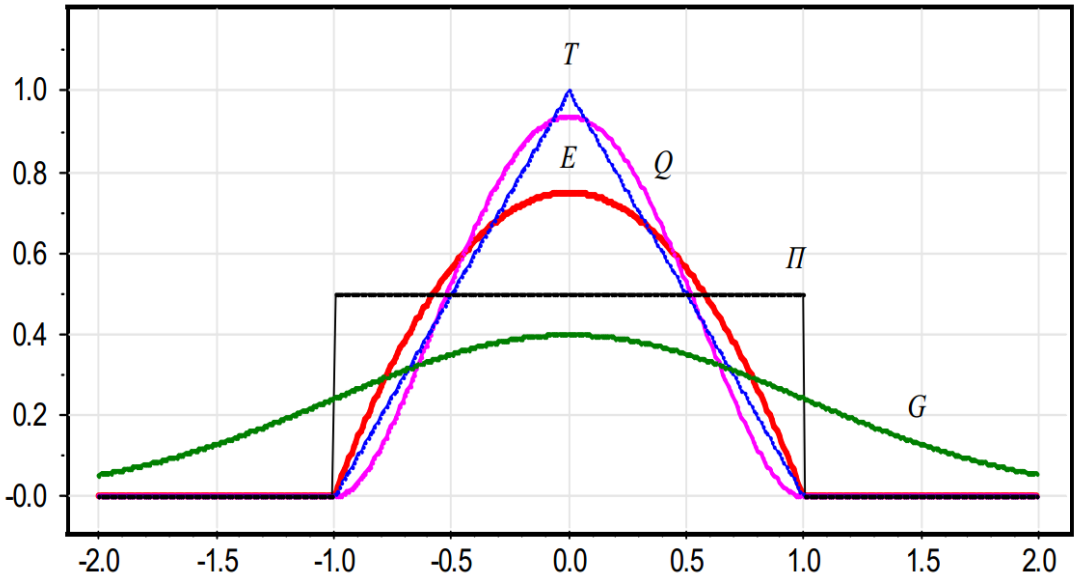
Как строить?

- Рассмотрим одномерный случай:

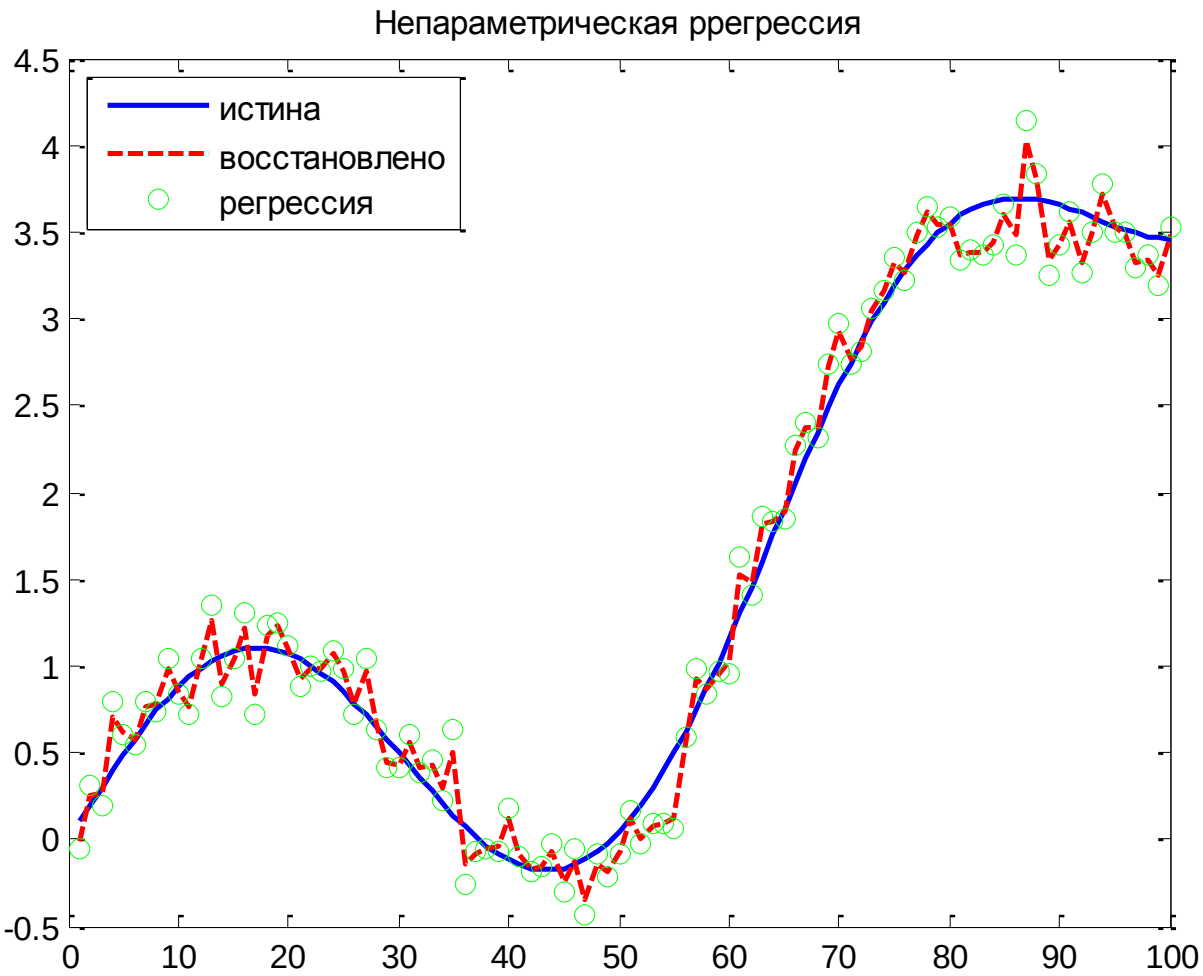


Сглаживание функций

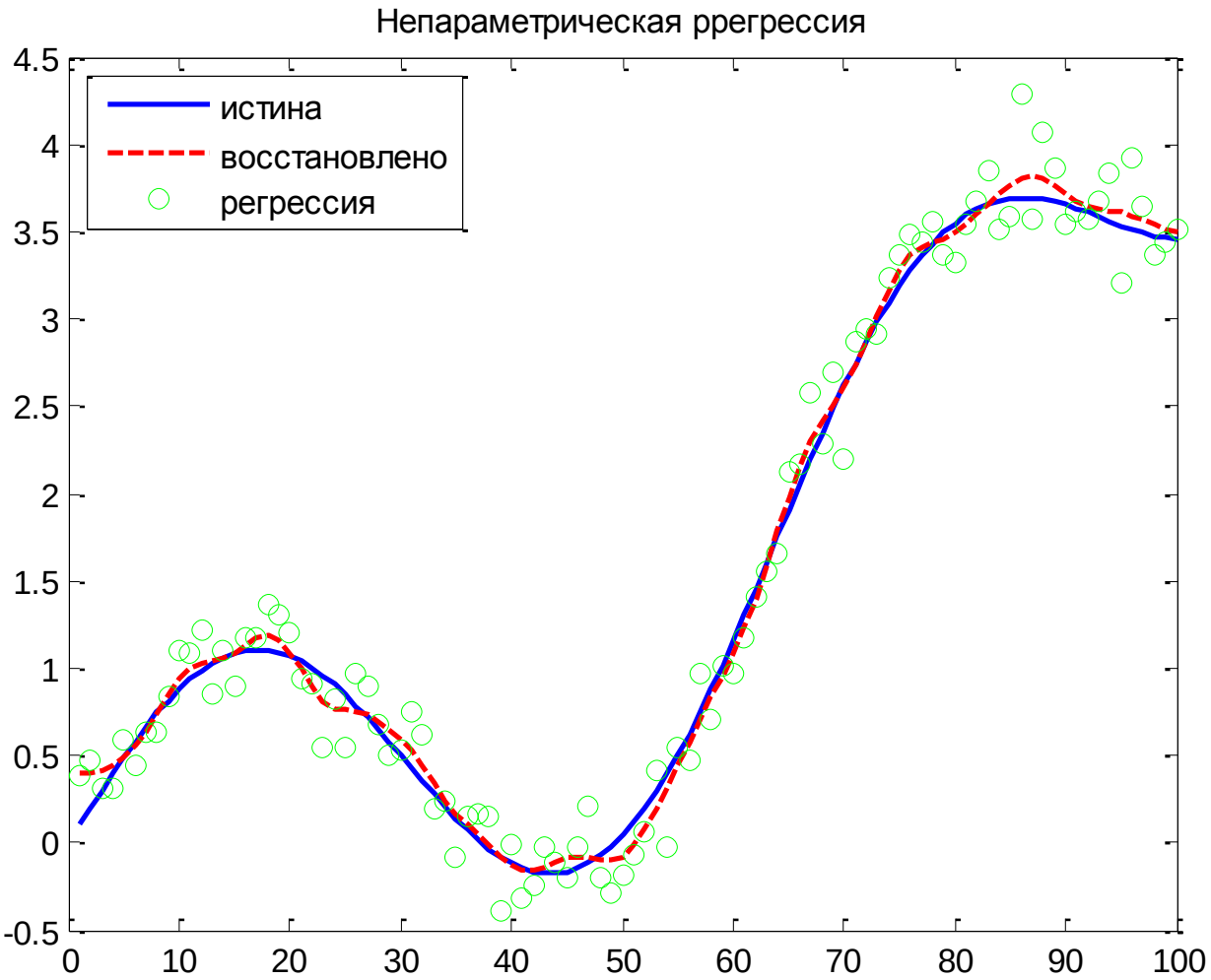
- Среднее по всем точкам
- Среднее по фиксированному отрезку вокруг точки
- Среднее по всем точкам с весами



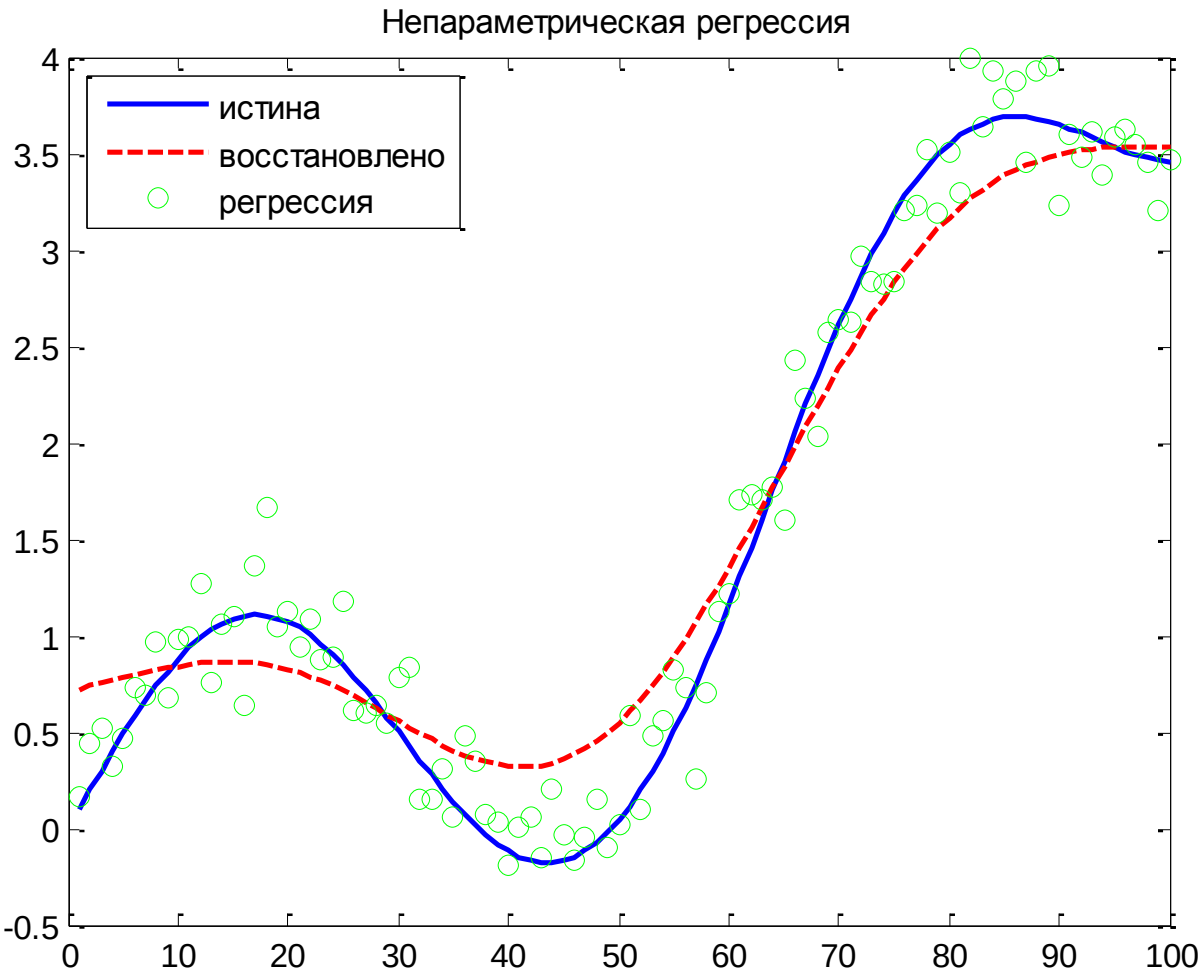
Маленькое окно



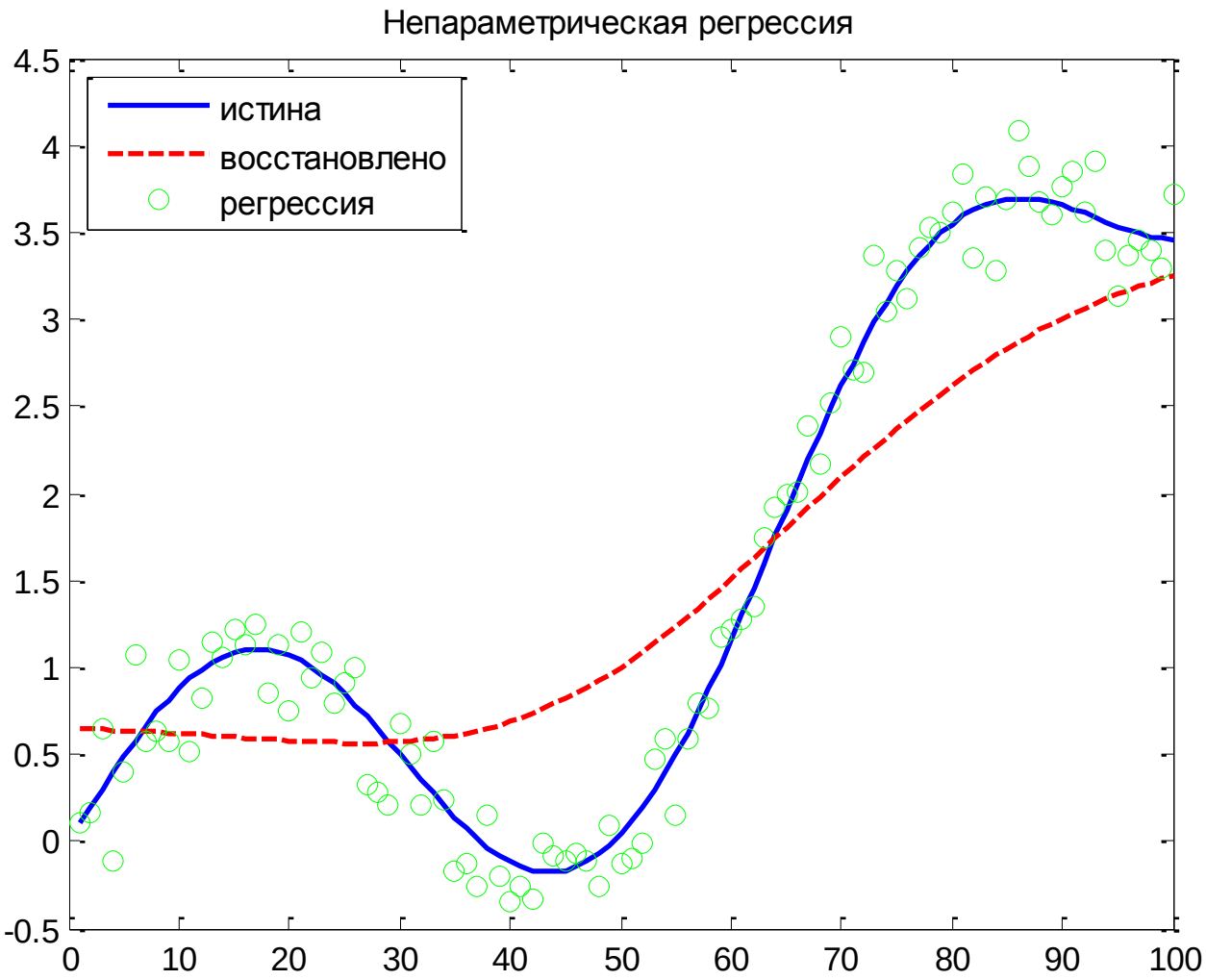
Окно побольше



Окно еще больше



Окно еще больше



Часть 12

Прогнозирование московских пробок

Постановка задачи

- Даны файлы с данными о средних скоростях на каждой улице Москвы
- Дана информация о 30 днях подряд (с 16 до 22 часов) и неполная информация о 31ом дне (с 16 до 18 часов)
- Даны длины улиц, рекомедованная средняя скорость и т.п.
- Надо предсказать скорость на каждой улице с 18 до 22 часов 31го дня

Первые идеи

- Учитывать выбросы
- Учитывать пропуски в данных
- Учитывать дни недели

Дни недели

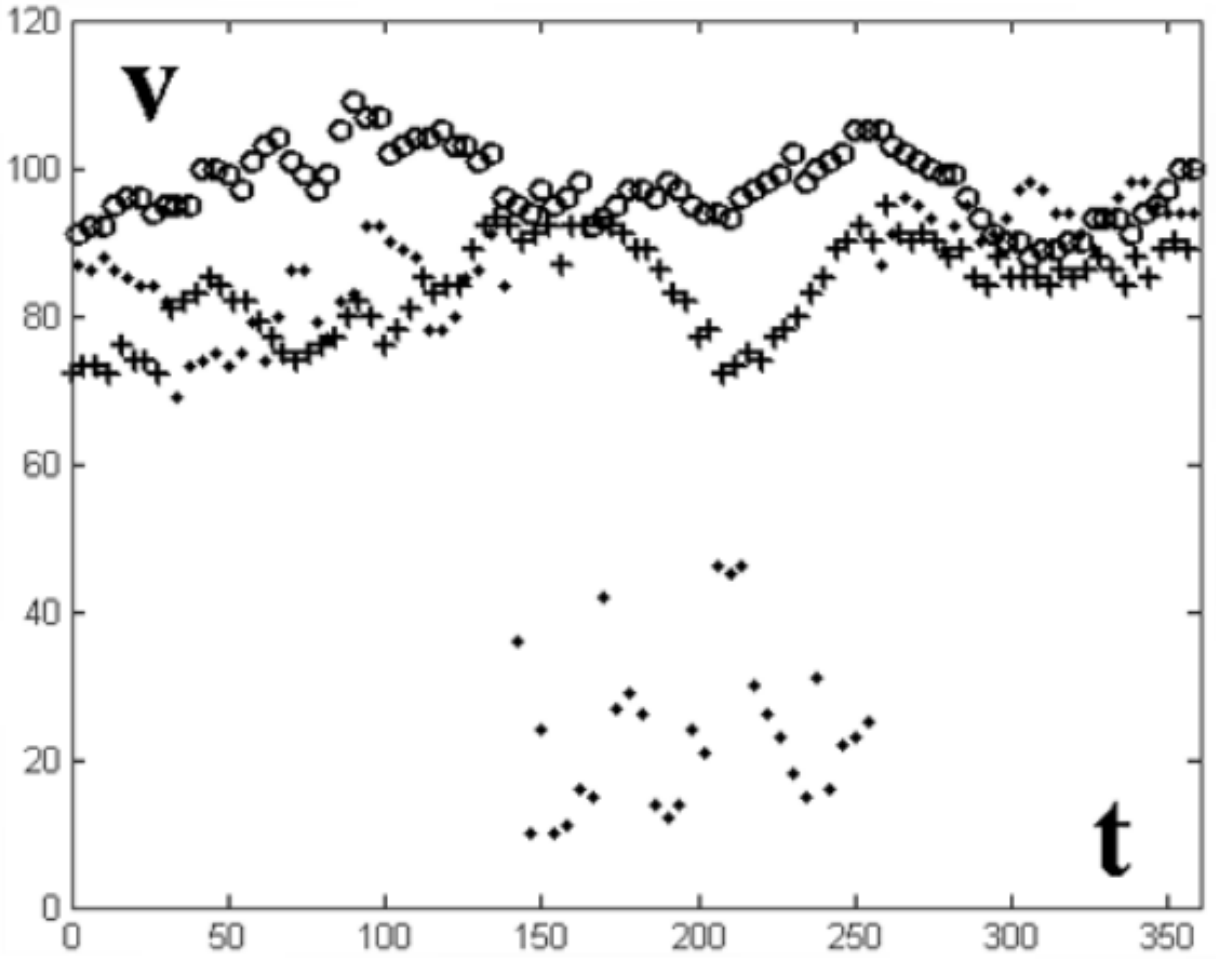


Рис. 4. Понедельник (сплошная), пятница (точки) и воскресенье (кружки) последней известной недели на улице 925236

Неплохое решение

- Сглаживаем каждый понедельник каждой улицы
- В качестве ответа – средняя скорость по сглаженным прошлым понедельникам

В чем проблемы?

- Улучшения – введение новых параметров
- Сложно проверять качество

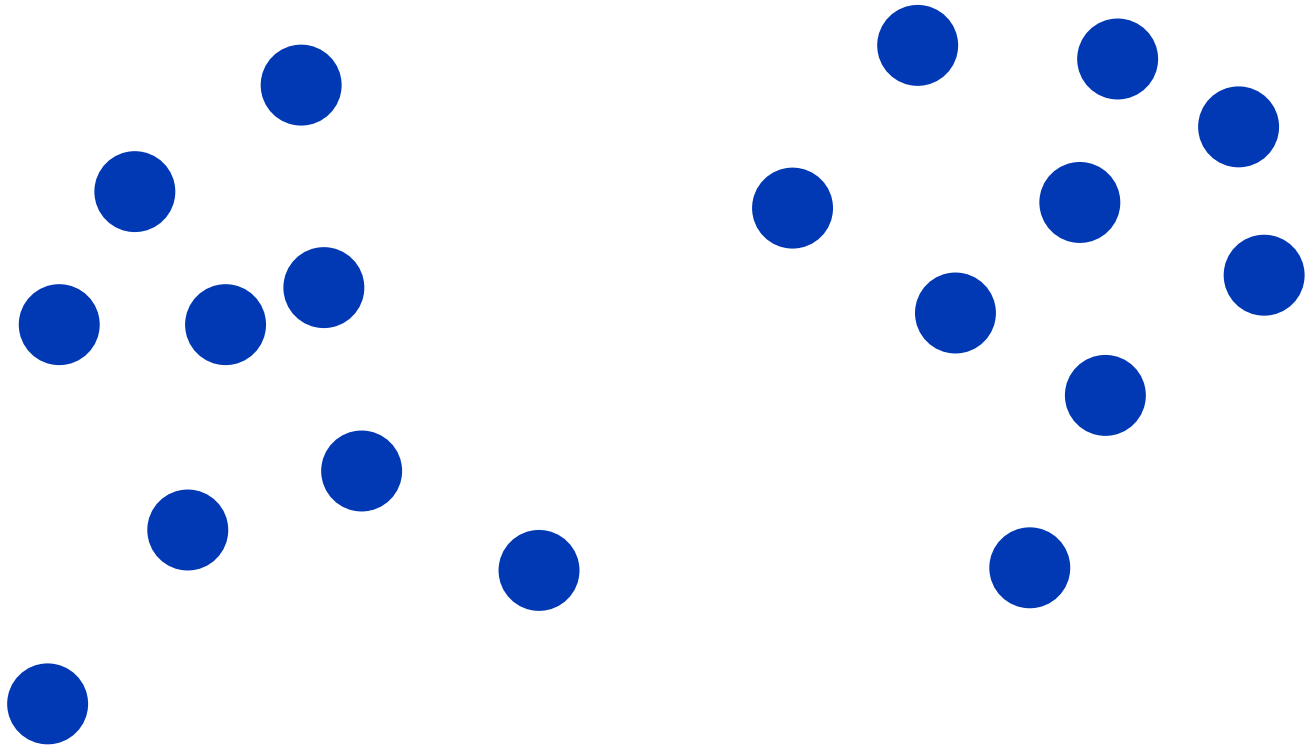
Другие идеи

- Учитывать другие дни недели с весами
- Учитывать недели с весами
- Учитывать скорости с 16 до 18 часов в искомый день
- Находить похожие улицы

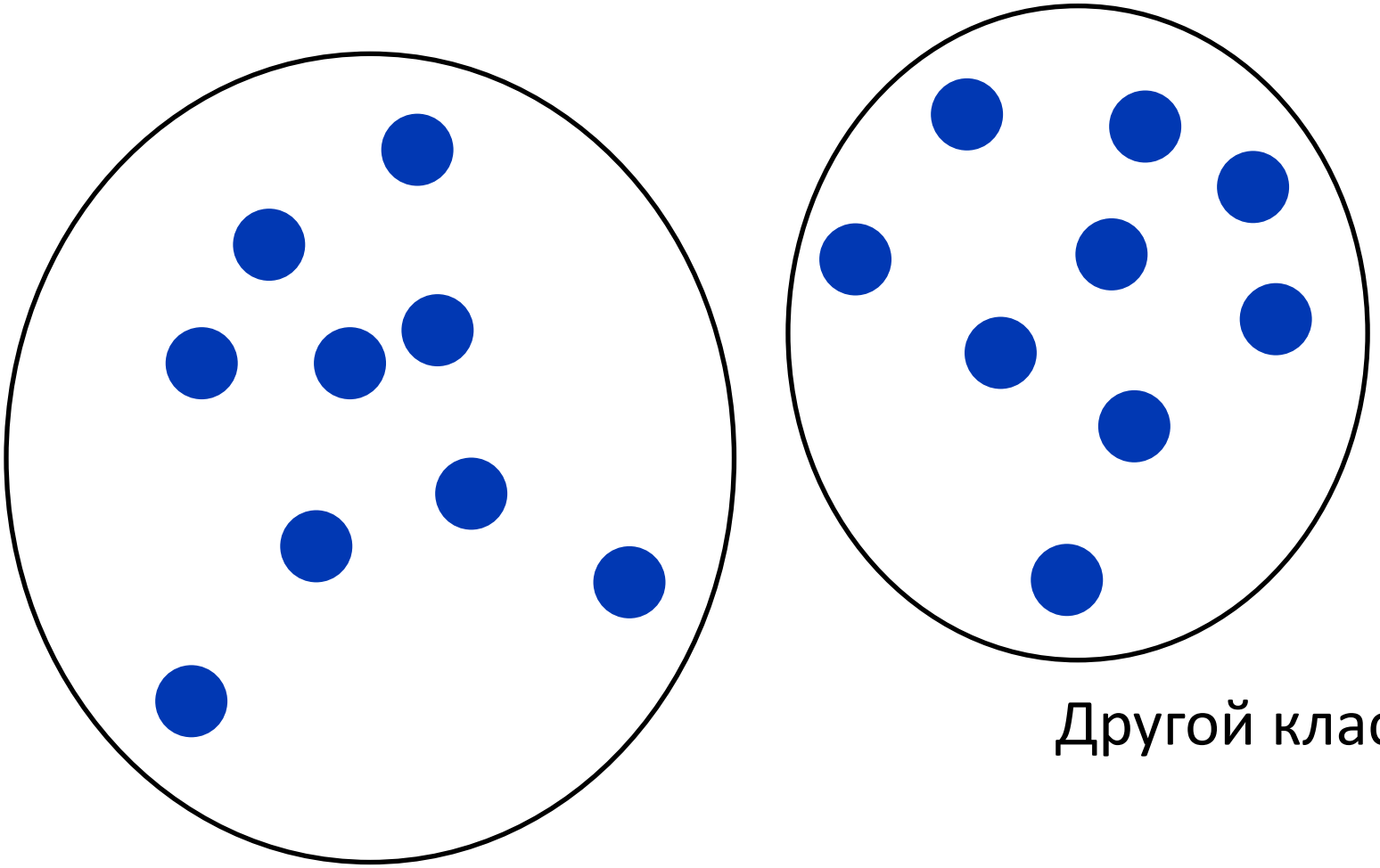
Часть 13

Кластеризация

Кластеризация – обучение без учителя



Задача кластеризации



Один кластер

Другой кластер

Постановка задачи

- Дано: множество объектов
- Надо: разделить объекты на группы (кластеры) таким образом, что:
  - Каждый кластер состоит из близких объектов
  - Объекты разных кластеров существенно различны

Задача не до конца формализована

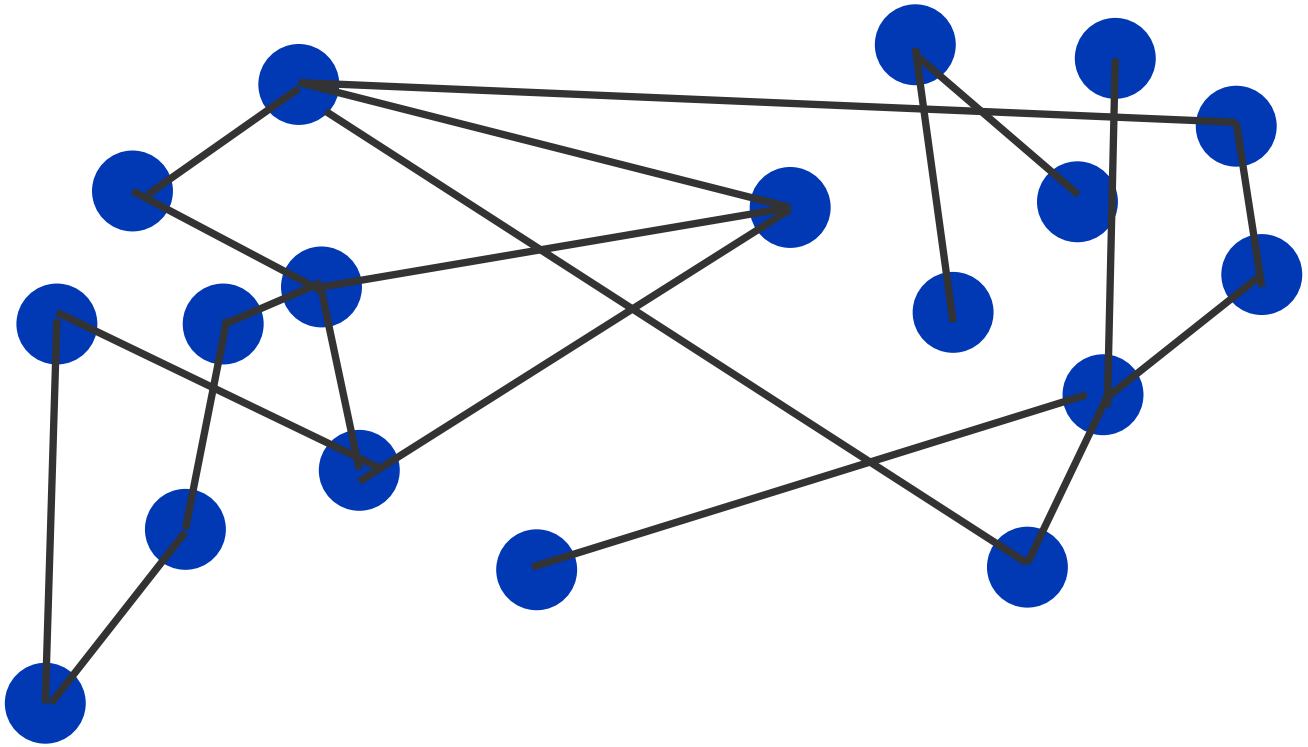
- Что значит близкие объекты?
- Что значит существенно различные объекты?
- Какое разбиение на кластеры лучше? Как мерять качество?
- Даже нужное число кластеров неизвестно!

Задача до конца не формализована

- Главное – научиться считать расстояние между объектами.
- Стремимся сделать расстояние между объектами внутри кластера меньше, между объектами разных кластеров – больше.

Представим данные в виде полного графа

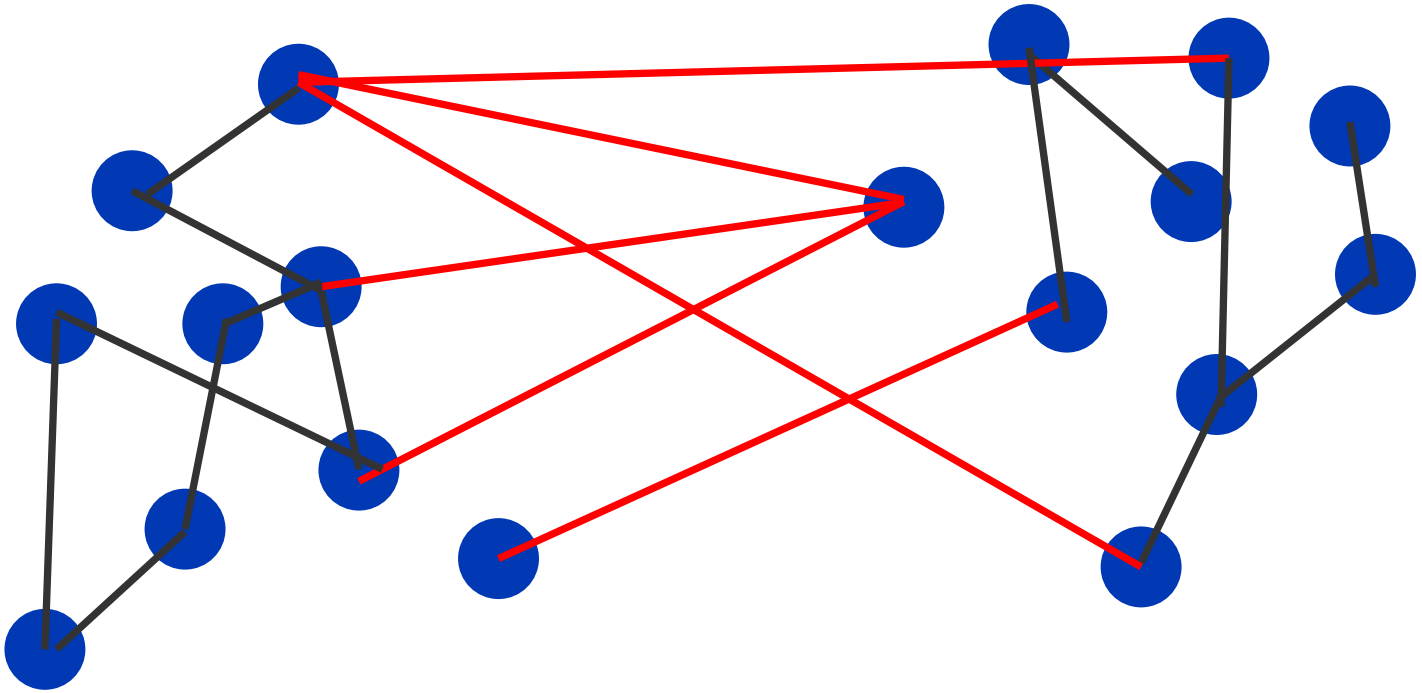
- Вершины графа – объекты
- Ребра – расстояния между ними



Алгоритм связных компонент

- Пусть мы хотим получить от K_1 до K_2 кластеров
- Алгоритм (возможно бинпоиском):
  - фиксируем некоторое расстояние R;
  - удаляем ребра длиной больше R;
  - K = число связных компонент;
  - если $K < K_1$, увеличиваем R, если $K > K_2$, уменьшаем R
- Повторяем заново, пока нас не устроит число кластеров

Алгоритм связных компонент



Недостатки:

- Неудобный параметр R
- Шумовые объекты все портят

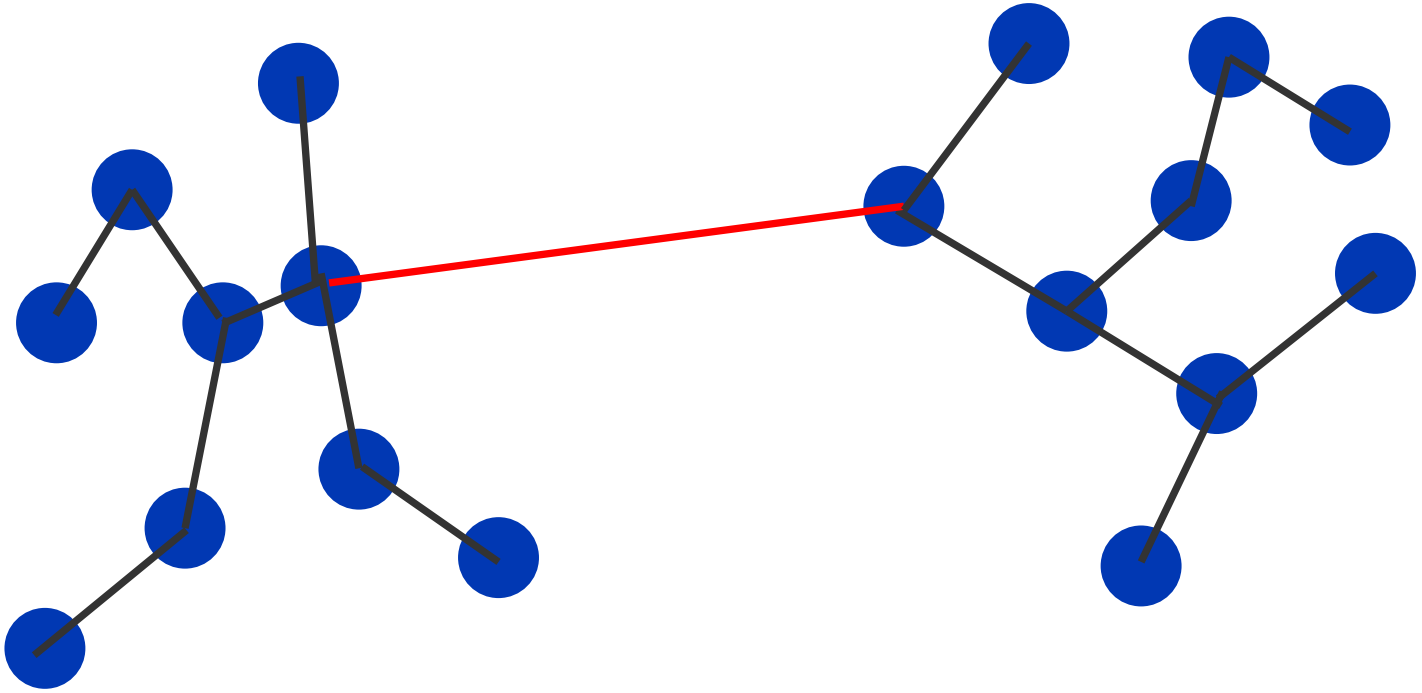
Вспомним алгоритм Краскала:

Найти пару вершин (i, j) с наименьшим расстоянием и соединить их ребром чтобы не образовывалось циклов.

Получится минимальное остовное дерево.

Как разбить на компоненты

- Получили остовное дерево
- Удалим К – 1 самых длинных рёбер
- Каждая компонента связности – отдельный кластер (К штук).



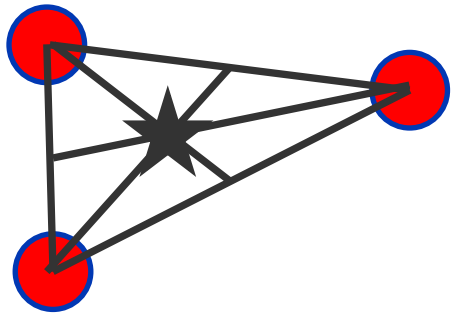
Достоинства и недостатки:

- Достоинство:
  - задается число кластеров K
- Недостаток:
  - шум снова все портит

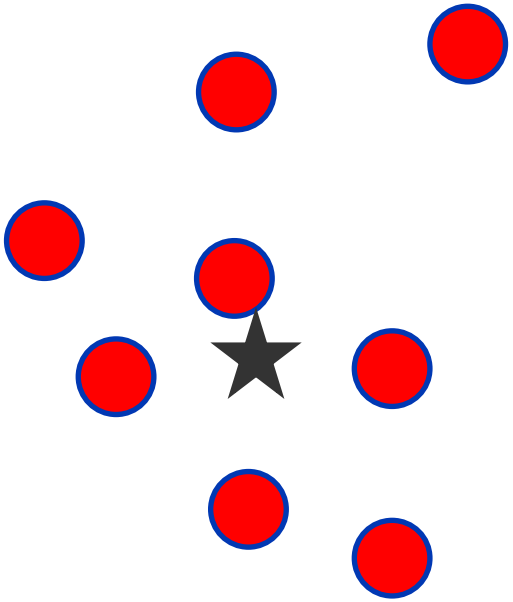
Лирическое отступление: центр масс

Две точки:

Три точки:



Произвольное облако точек:



Алгоритм ФОРЭЛ

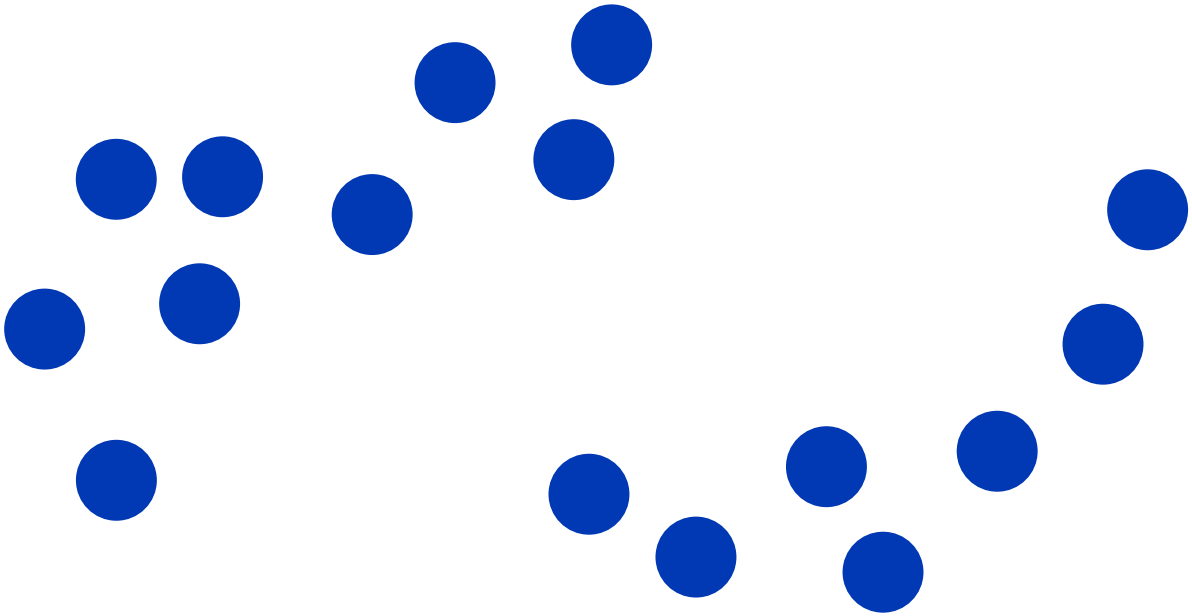
Шаг 1: объединяем точки в маленькие кластеры

пусть U — множество некластеризованных точек;

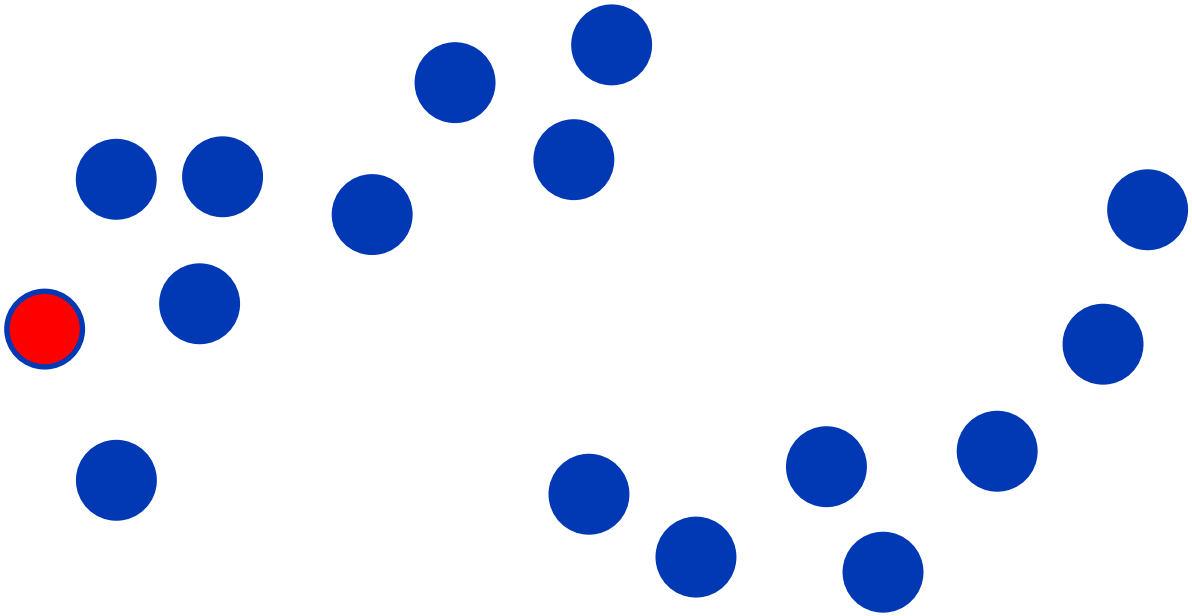
пока в выборке есть некластеризованные точки:

- взять случайную точку x из U;
- повторять:
  - образовать кластер с центром в x и расстоянием до объектов не больше R;
  - переместить центр в центр масс кластера;
- пока состав кластера не стабилизируется;
- удалить точки нового кластера из множества U .

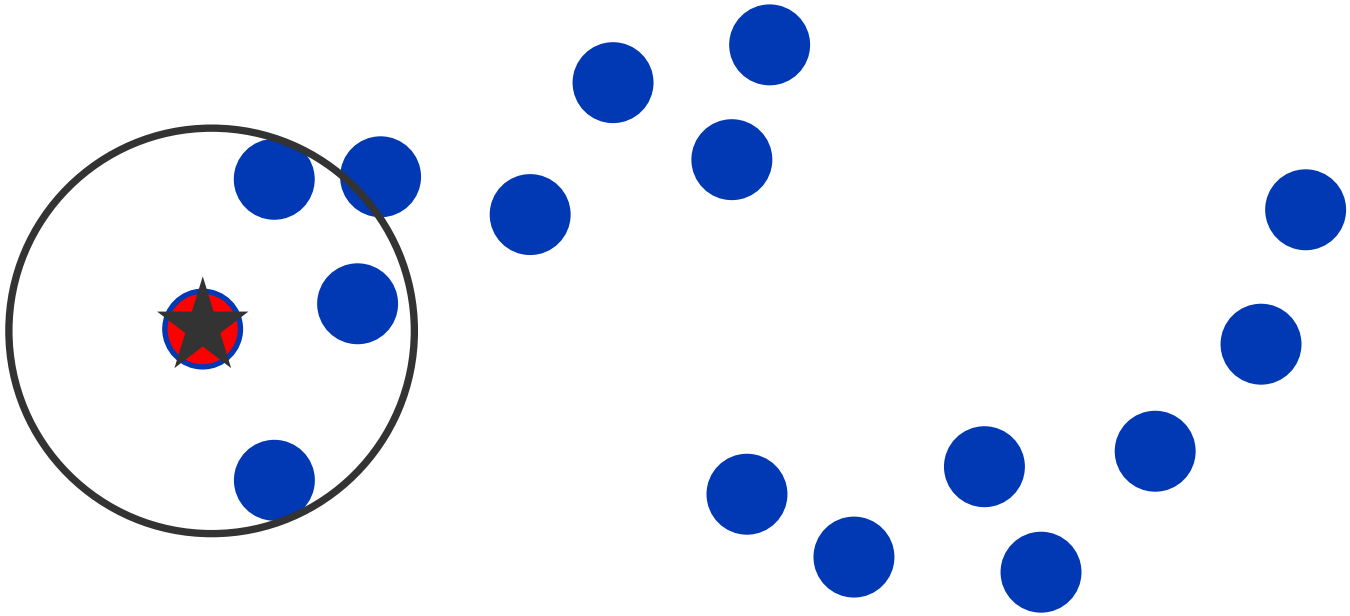
Алгоритм ФОРЭЛ, шаг 1



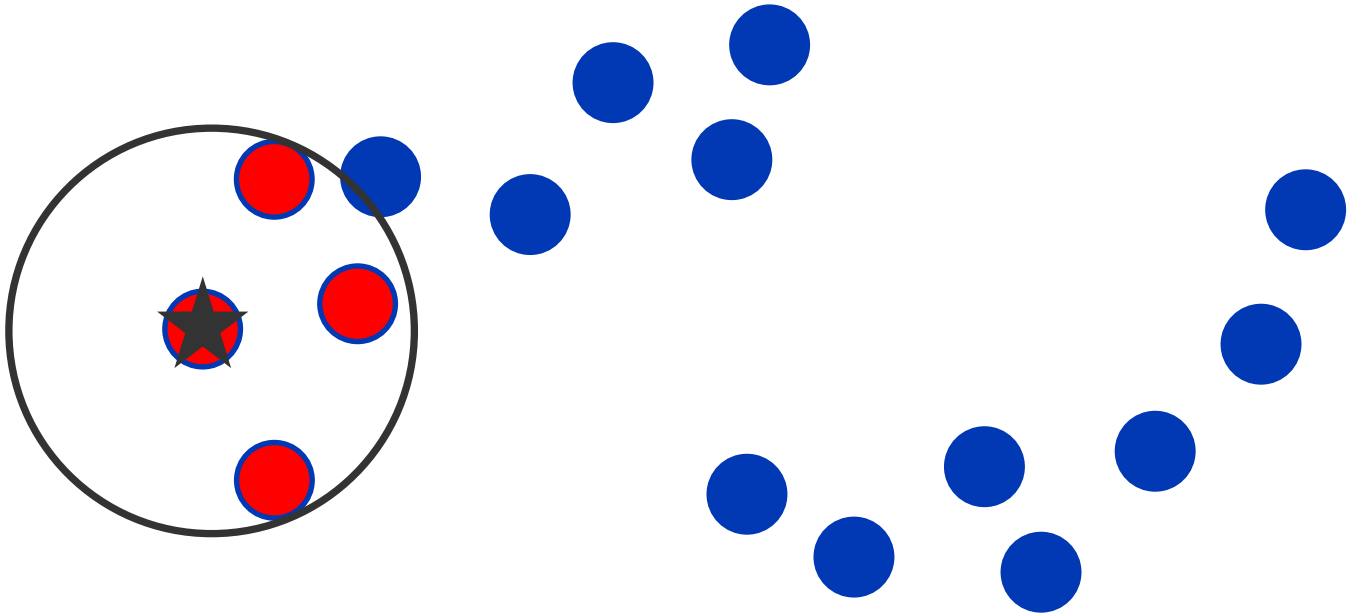
Алгоритм ФОРЭЛ, шаг 1



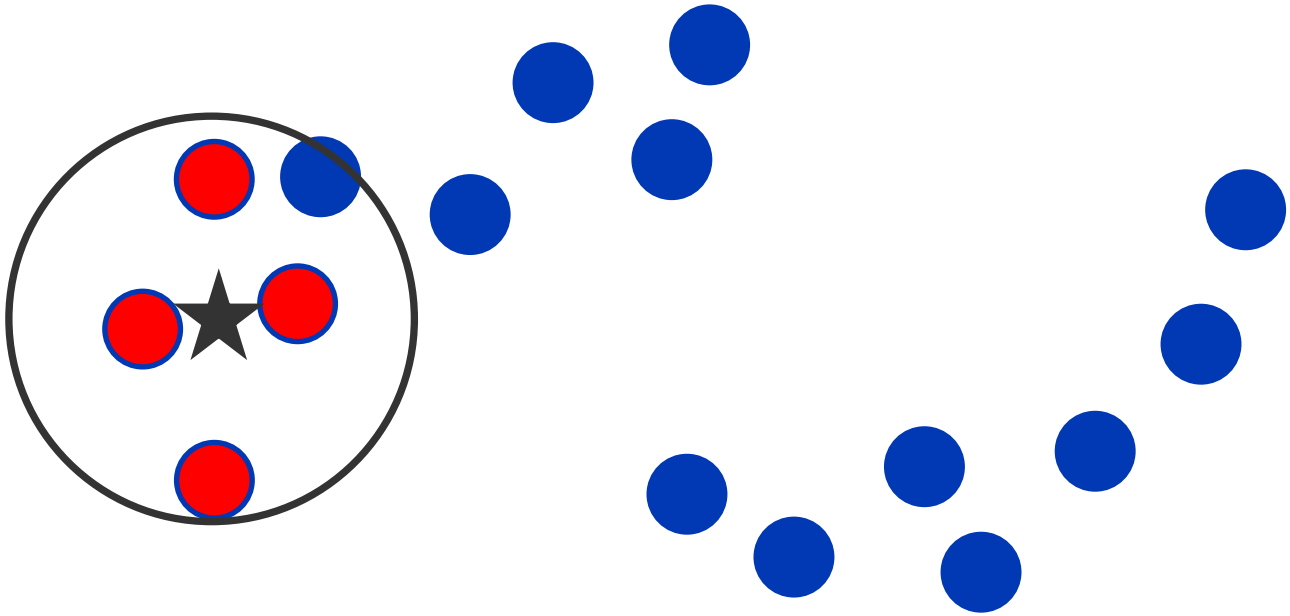
Алгоритм ФОРЭЛ, шаг 1



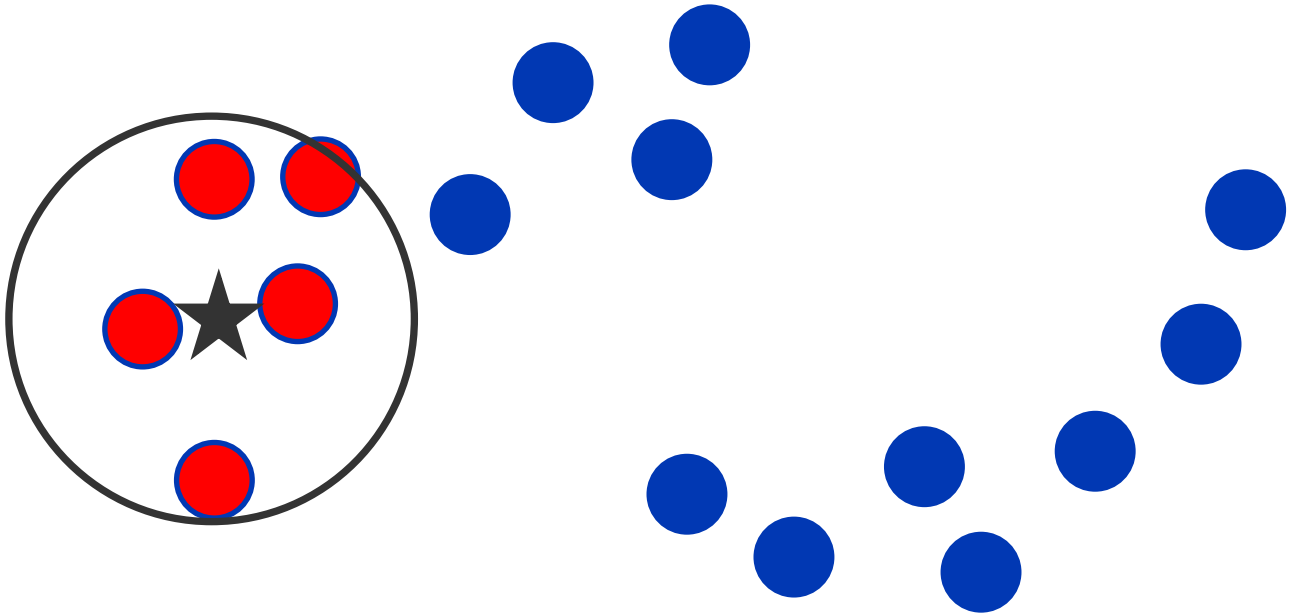
Алгоритм ФОРЭЛ, шаг 1



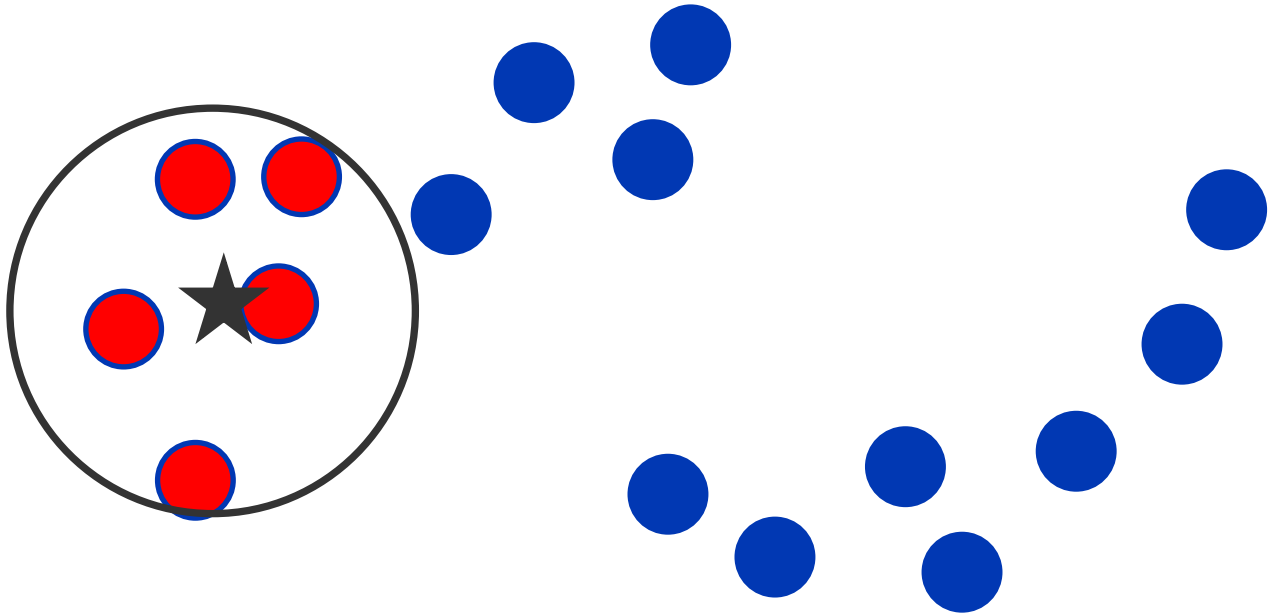
Алгоритм ФОРЭЛ, шаг 1



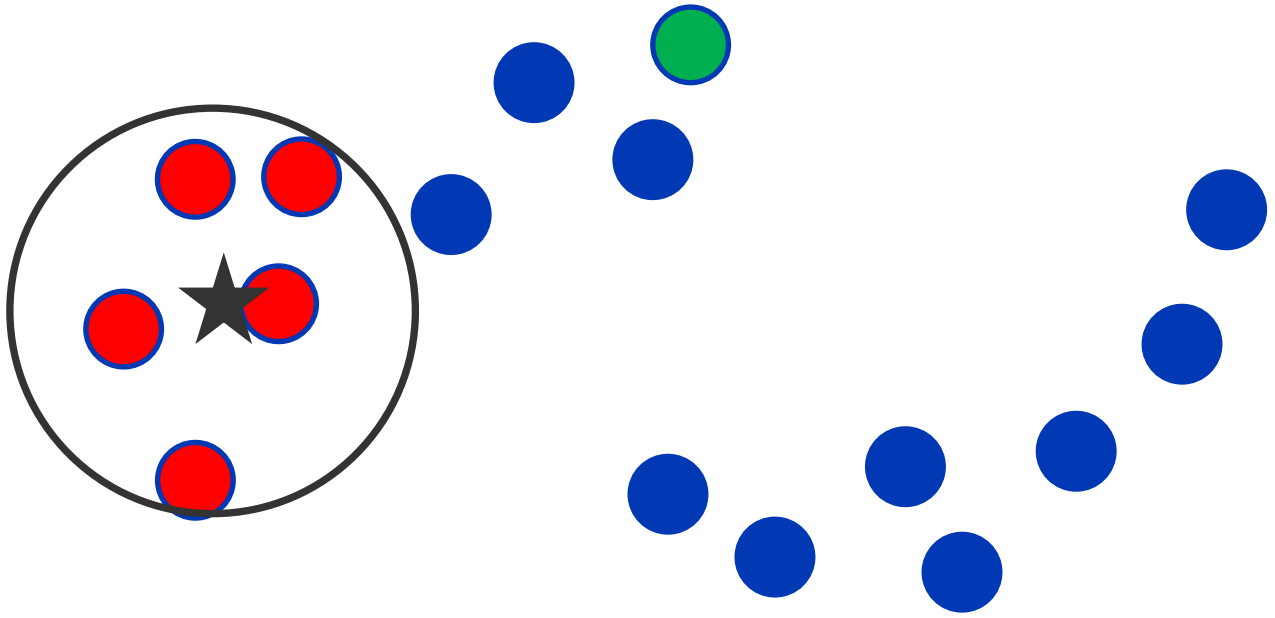
Алгоритм ФОРЭЛ, шаг 1



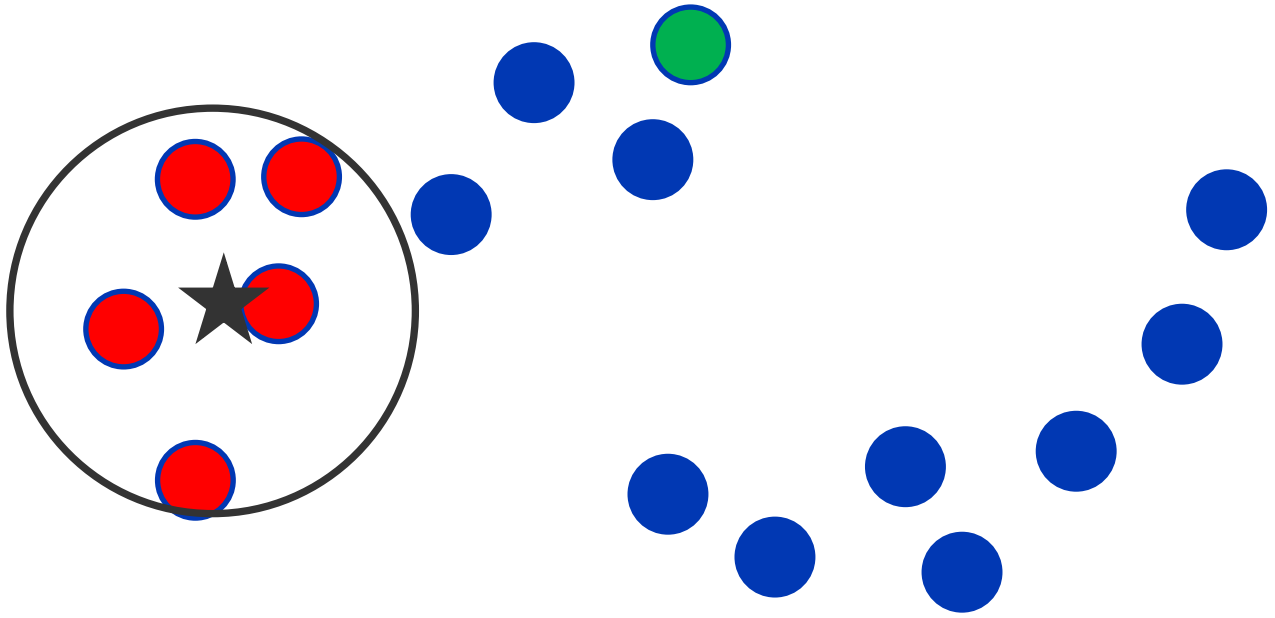
Алгоритм ФОРЭЛ, шаг 1



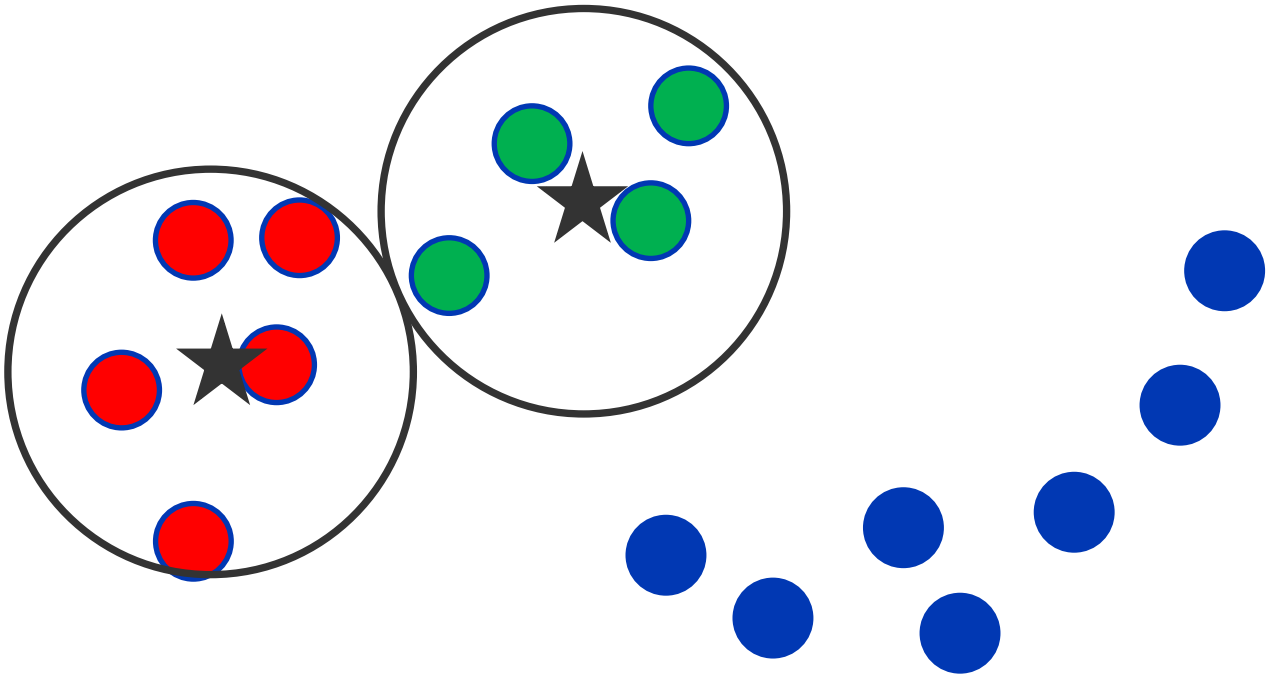
Алгоритм ФОРЭЛ, шаг 1



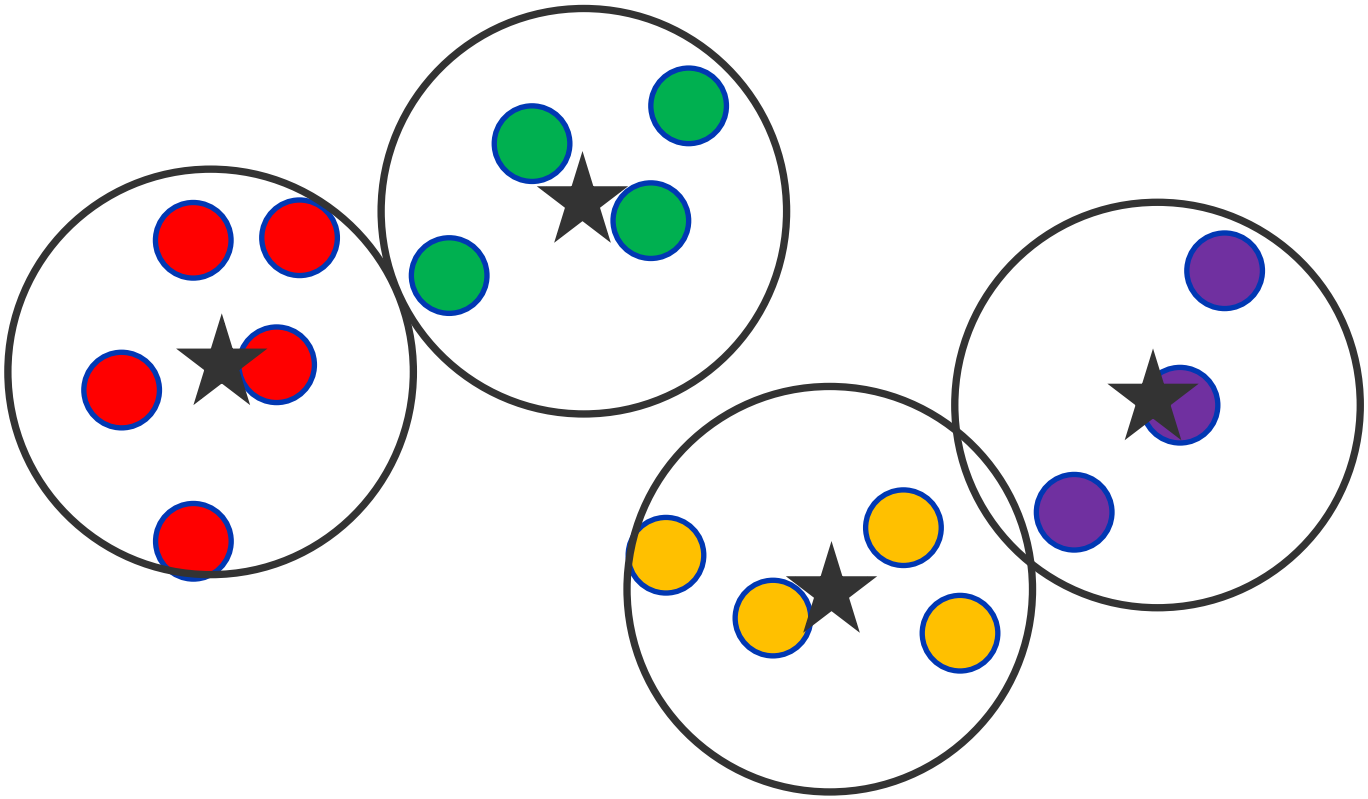
Алгоритм ФОРЭЛ, шаг 1



Алгоритм ФОРЭЛ, шаг 1



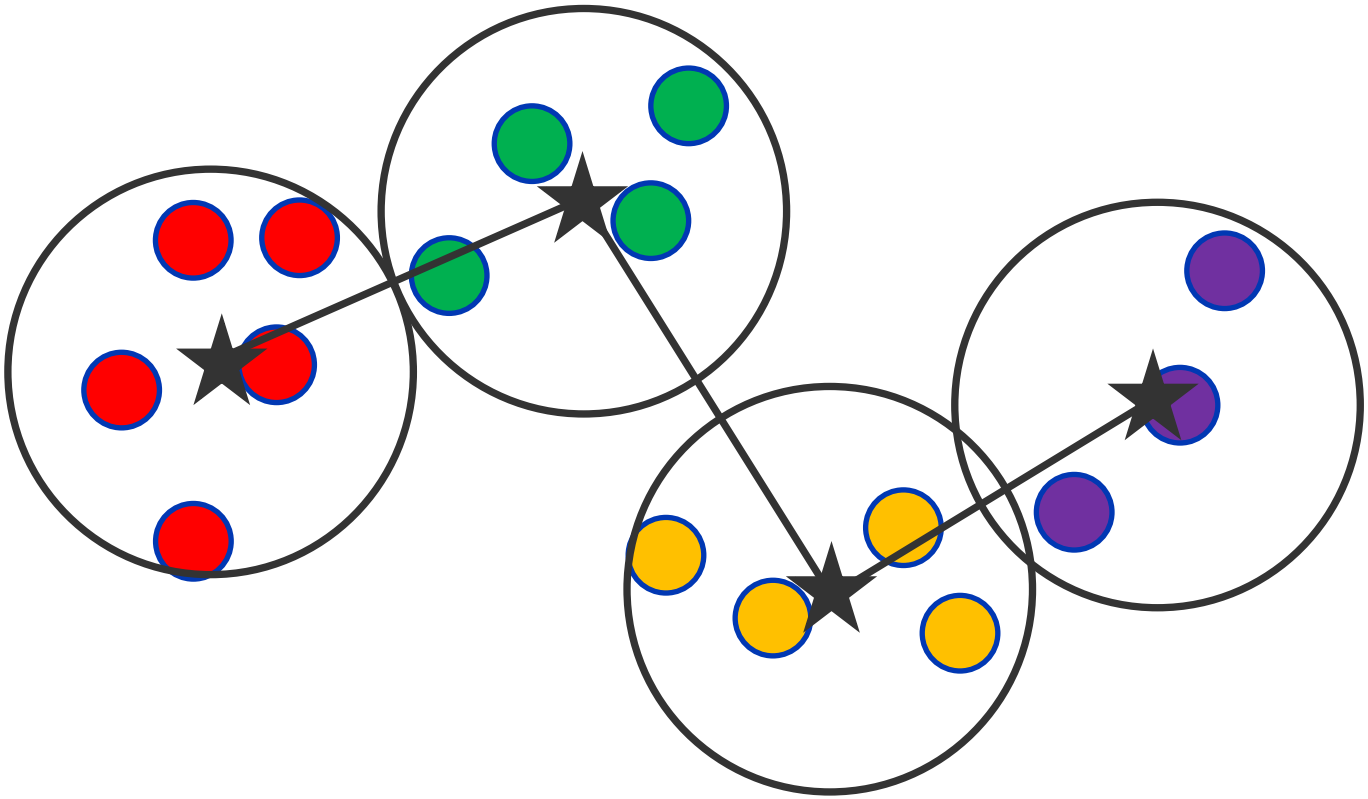
Алгоритм ФОРЭЛ, шаг 1



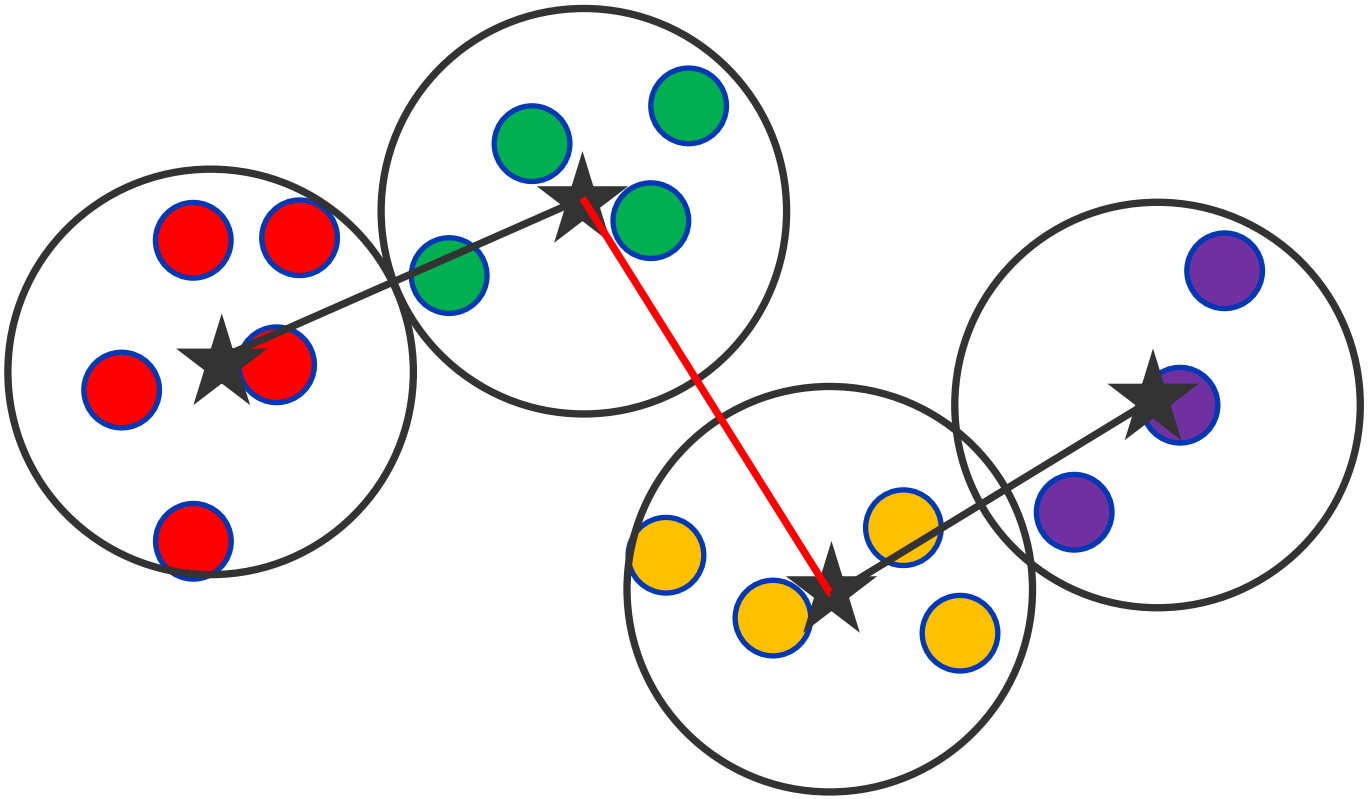
Алгоритм ФОРЭЛ

- Шаг 2: применим графовый алгоритм к множеству центров маленьких кластеров

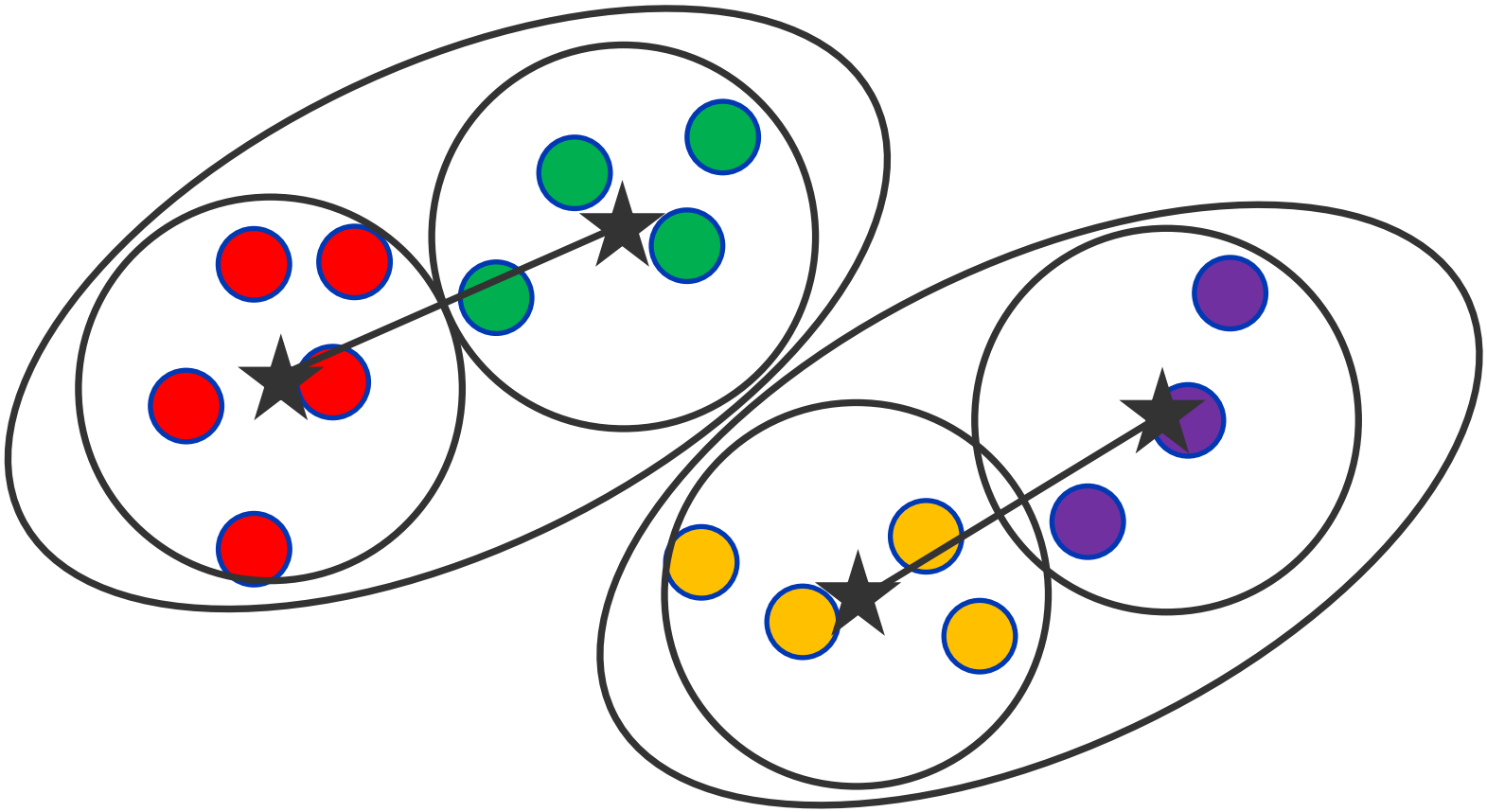
Алгоритм ФОРЭЛ, шаг 2



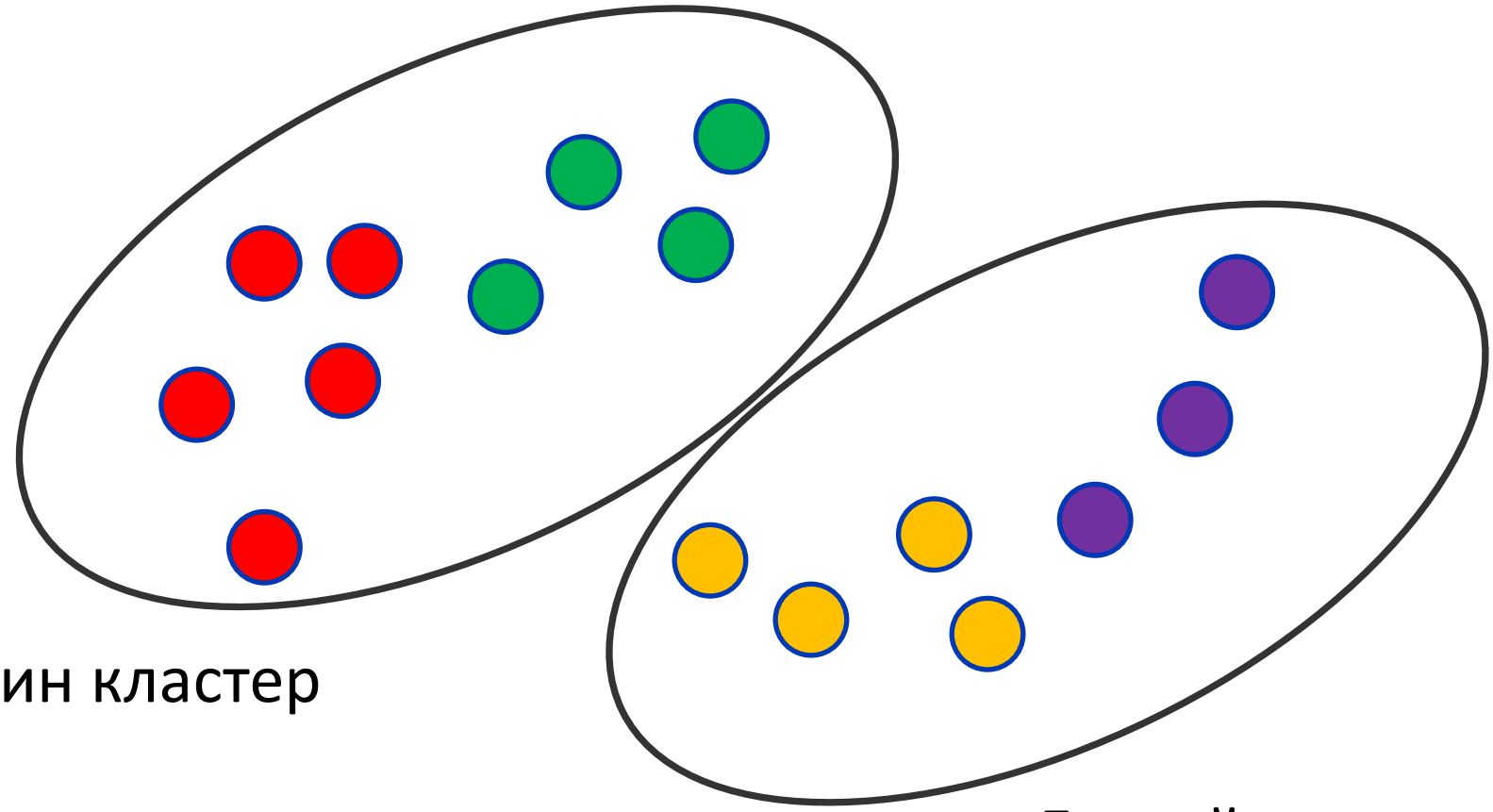
Алгоритм ФОРЭЛ, шаг 2



Алгоритм ФОРЭЛ, шаг 2



Алгоритм ФОРЭЛ, шаг 2



Один кластер

Другой кластер

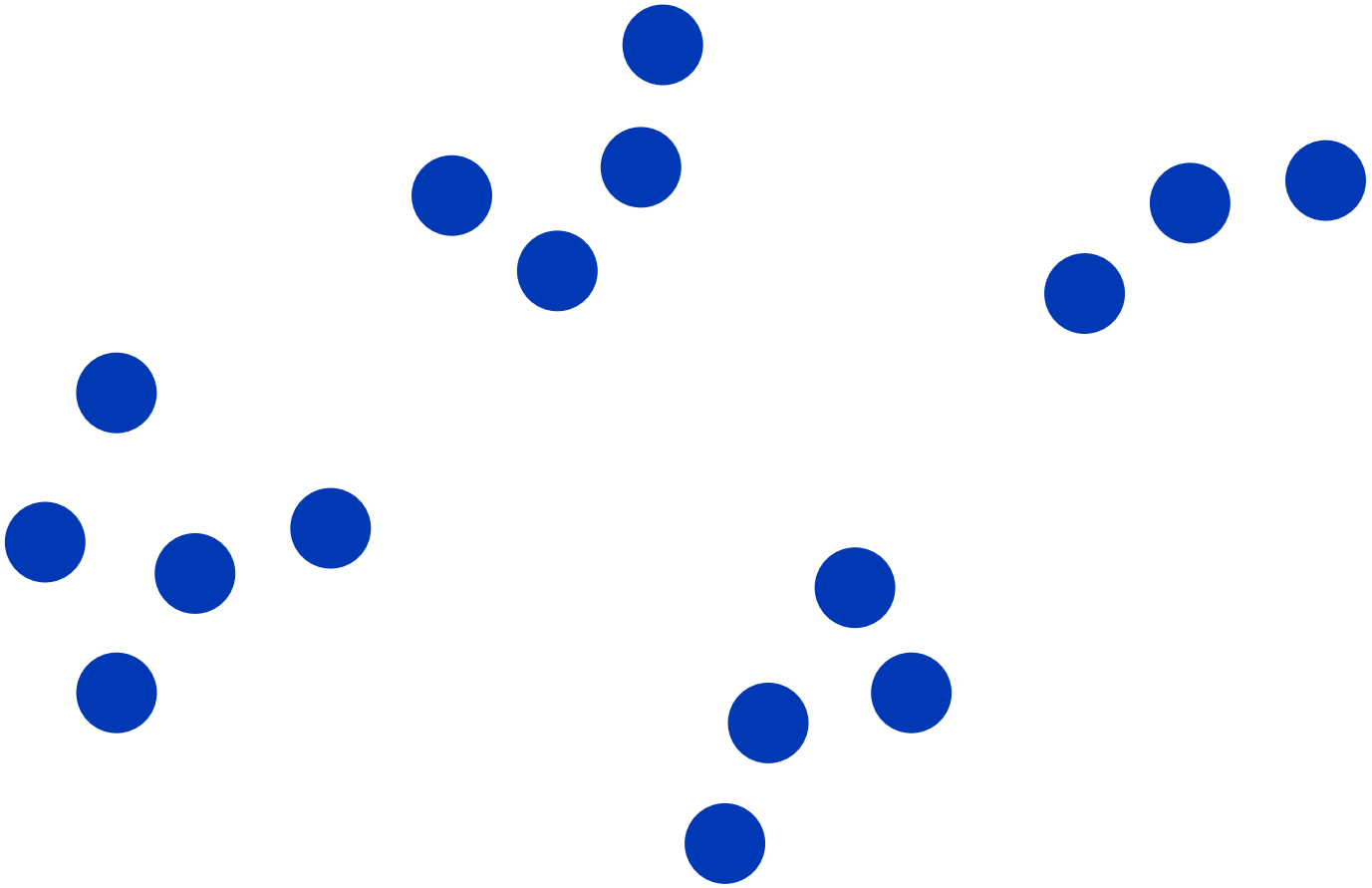
ФОРЭЛ

- Двухуровневая структура
- Параметр R управляется

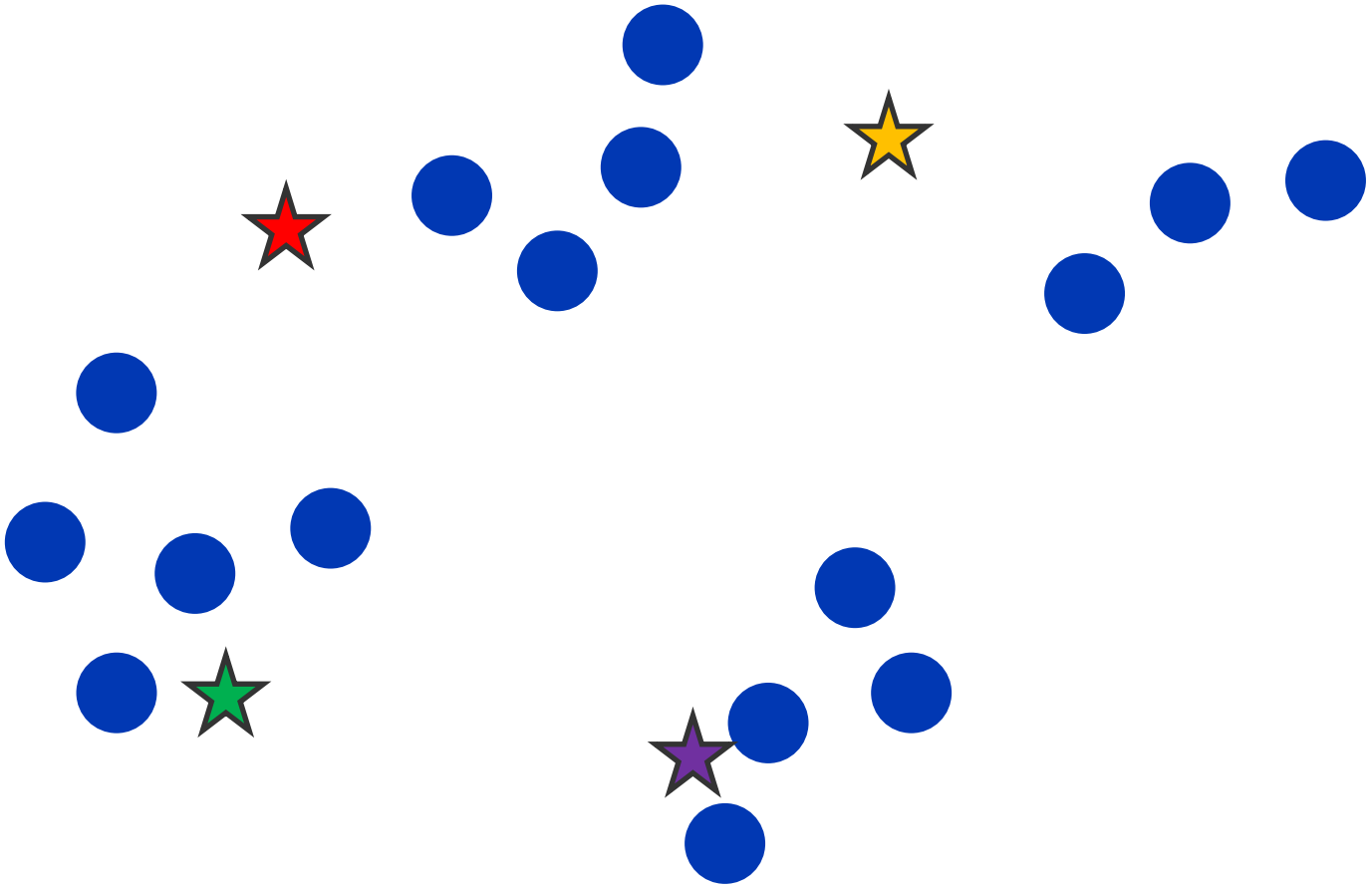
Метод k-средних

- выбрать начальные положения центров кластеров;
- повторять:
  - отнести каждый объект к ближайшему центру;
  - перенести новые положения центров в центры масс кластеров;
- пока положения центров не перестанут изменяться

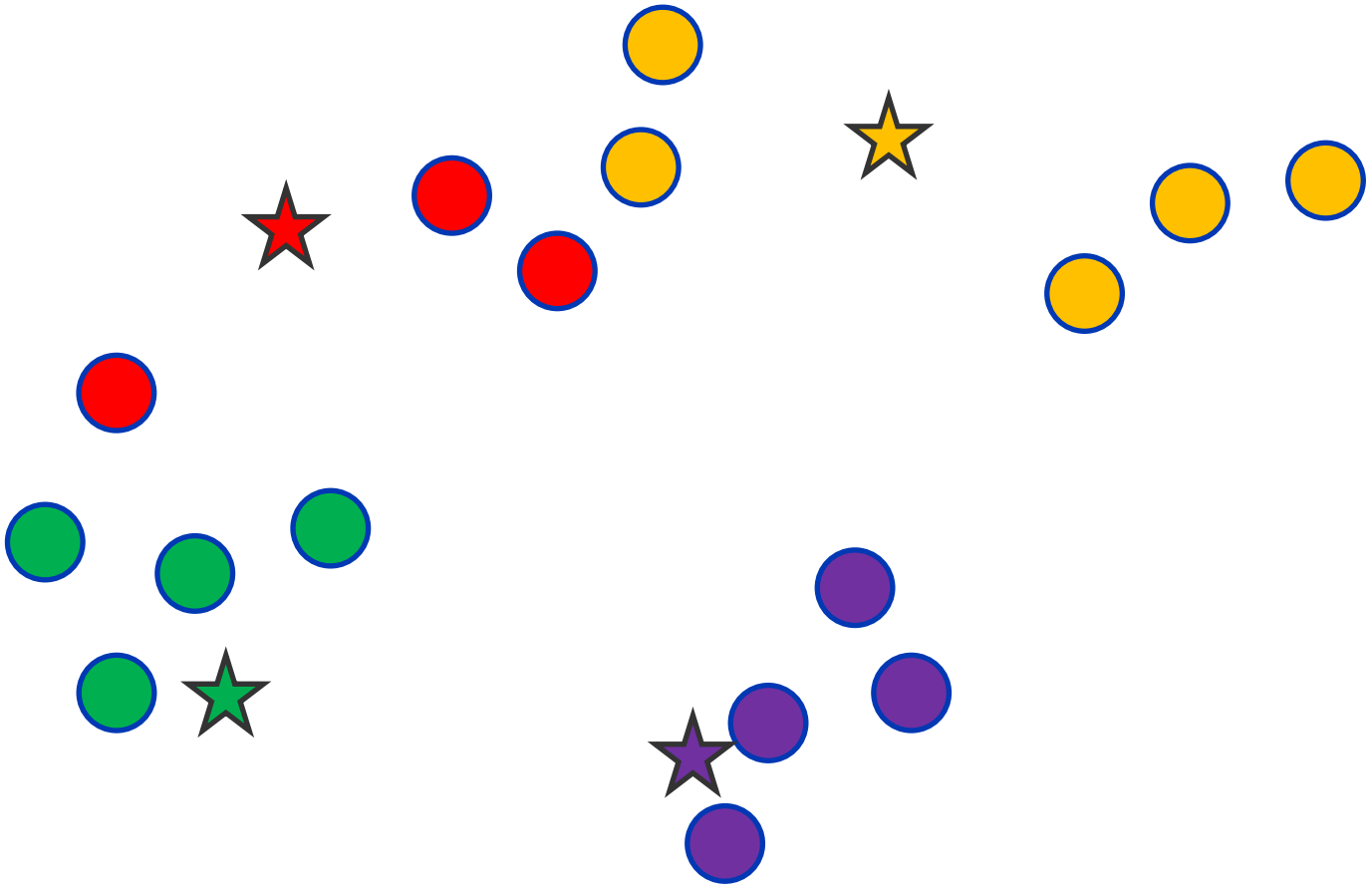
Метод k-средних



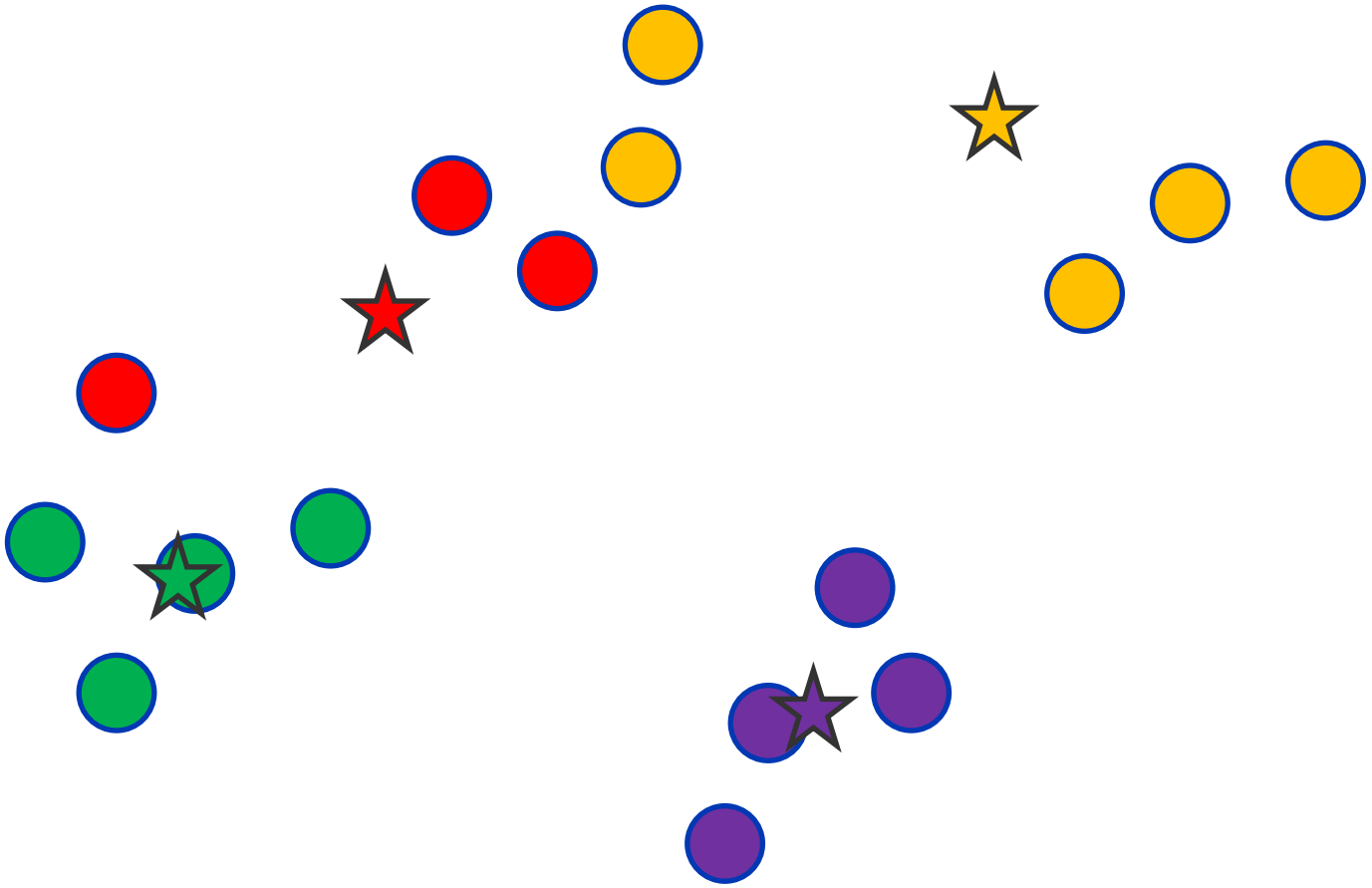
Начальное приближение центров кластеров



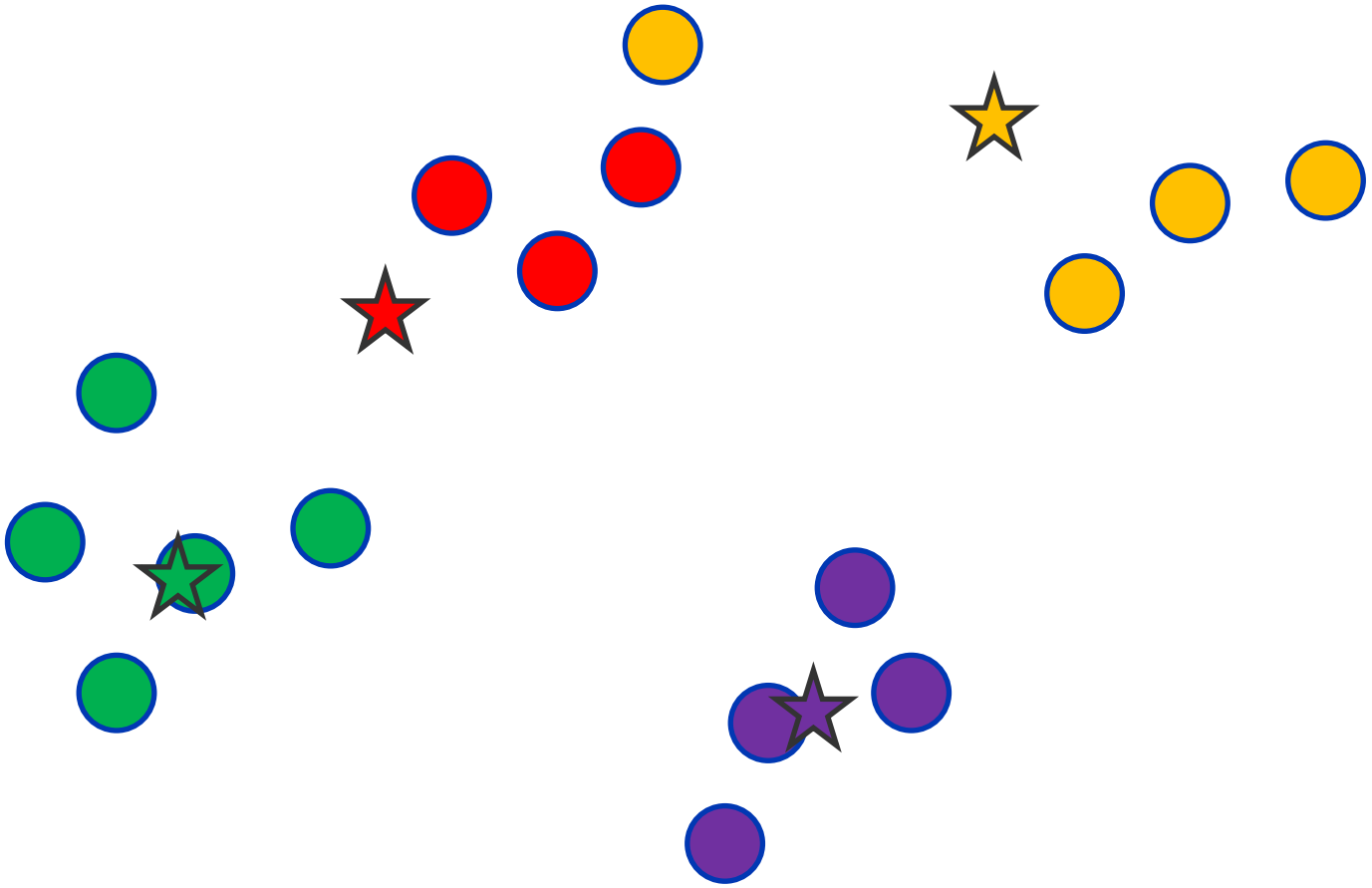
Распределяем объекты ближайшим центрам



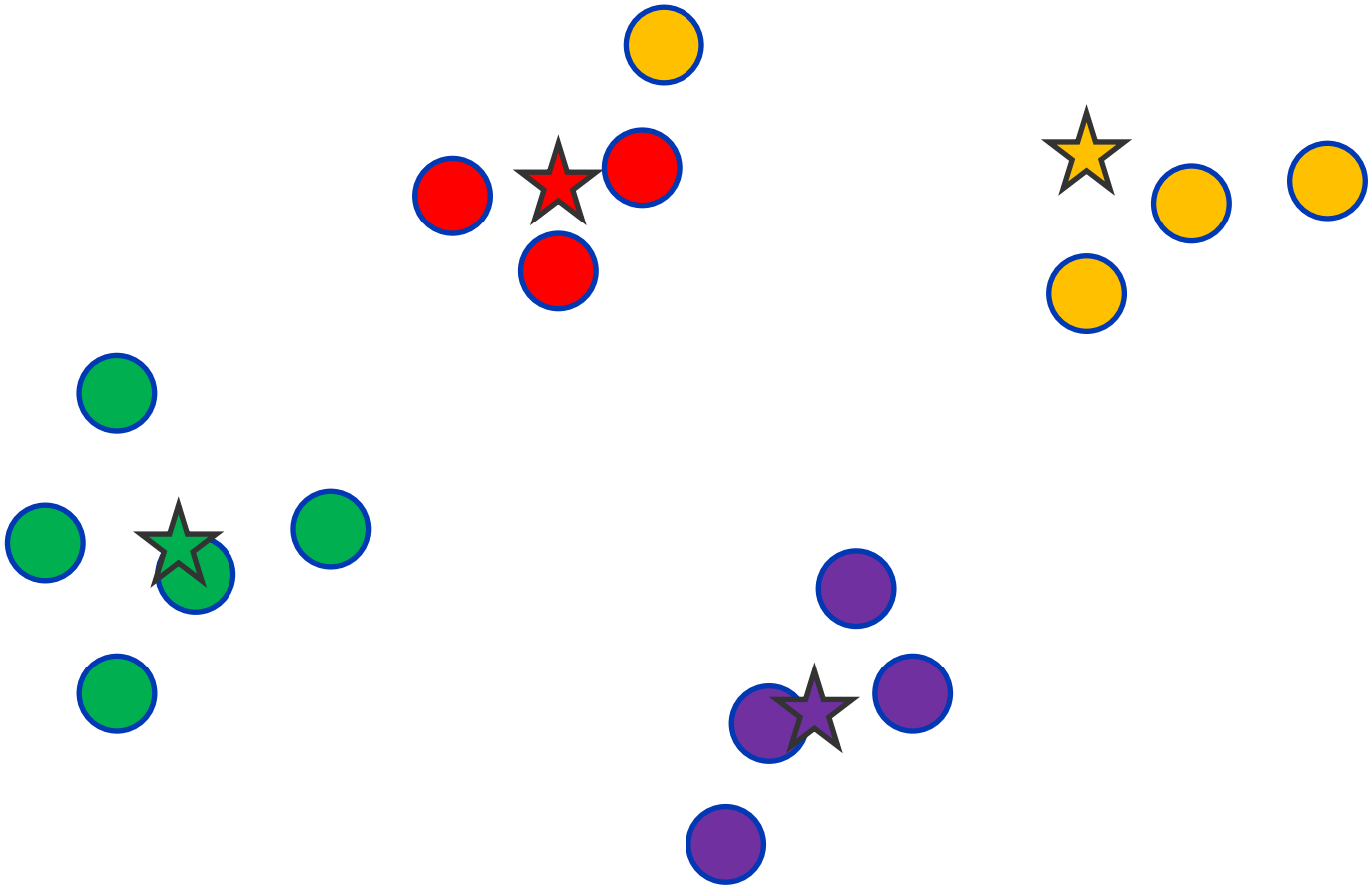
Уточняем положения центров



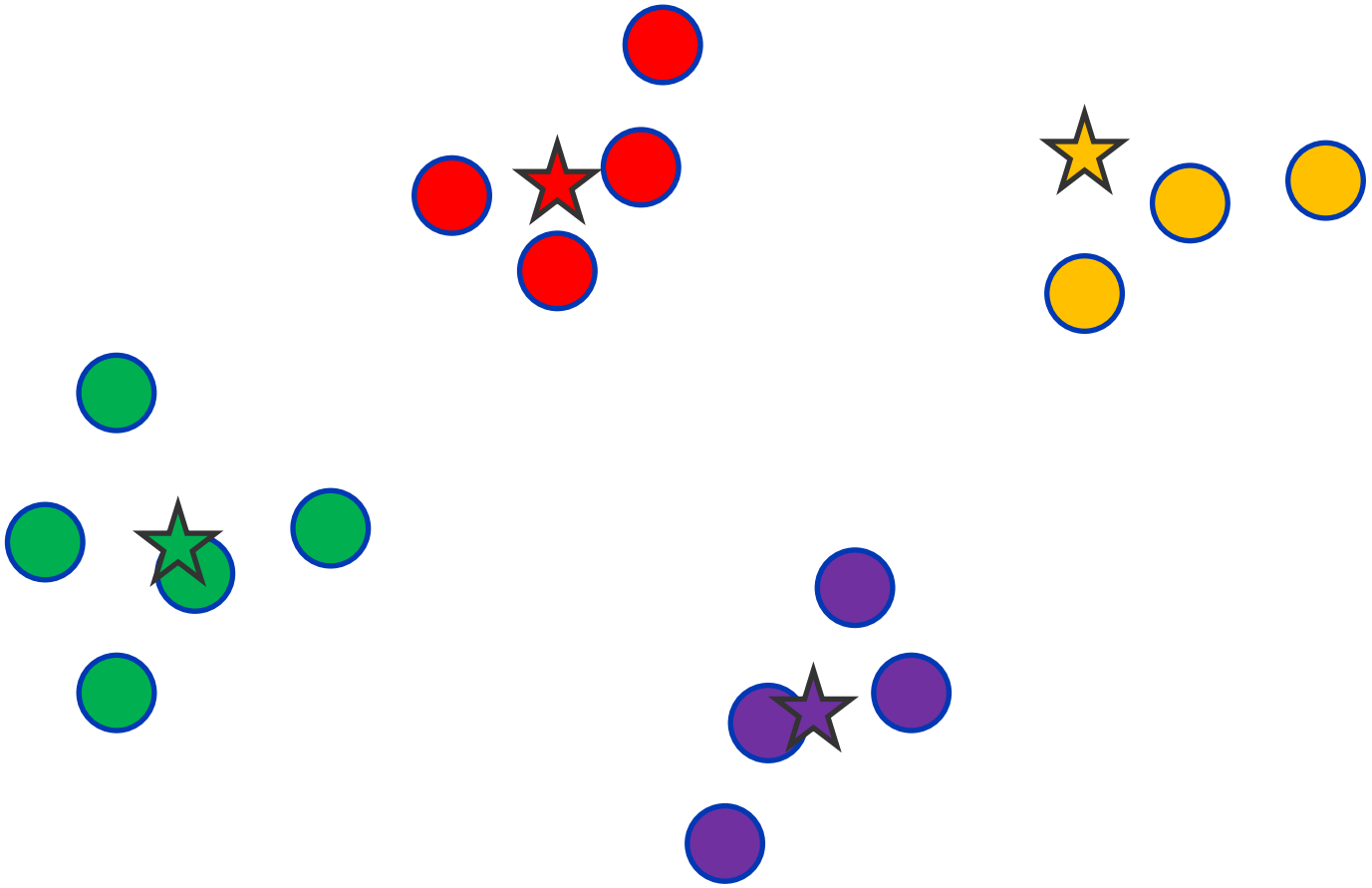
Перераспределяем объекты



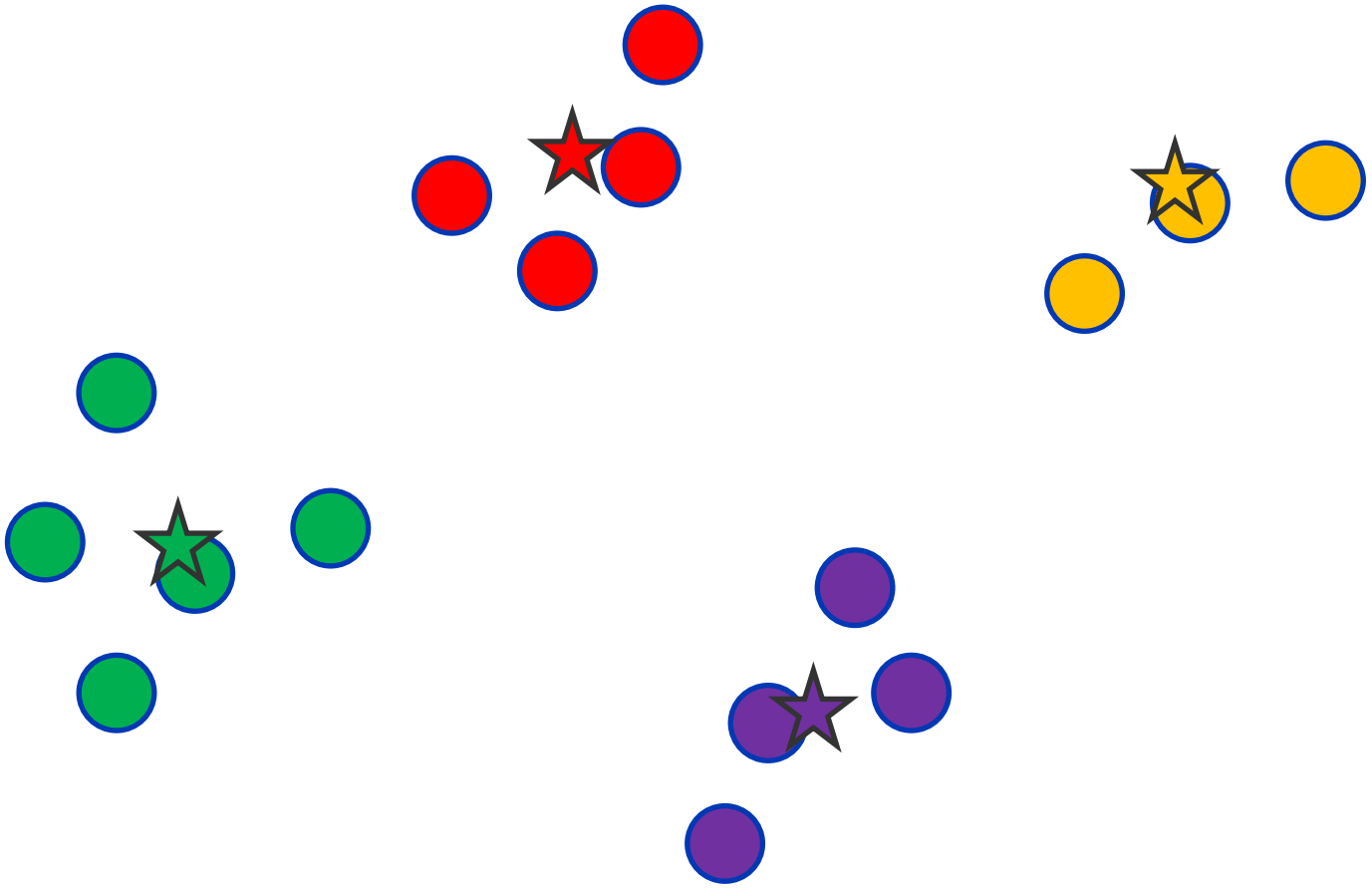
Снова уточняем положения центров



И ещё раз – перераспределяем объекты…



… и уточняем положения центров. Кластеризация готова!



А ведь картинки могут быть и совсем другие…

кластеры могут соединяться перемычками

кластеры могут накладываться на разрежен-
ный фон из редко расположенных объектов

кластеры могут перекрываться

А ведь картинки могут быть и совсем другие…

кластеры могут образовываться не по сходству, а по иным типам регулярностей

кластеры могут вообще отсутствовать

Зачем нужна задача кластеризации?

- Упростить дальнейшую обработку данных: разбить множество объектов на группы схожих объектов, чтобы работать с каждой группой в отдельности
- Сократить объём хранимых данных, оставив по одному представителю от каждого кластера (задачи сжатия данных)
- Выделить нетипичные объекты, которые не подходят ни к одному из кластеров
- Построить иерархию множества объектов

Заключение

Что еще есть в машинном обучении?

- Ассоциативные правила
- Частичное обучение
- Пропуски в данных
- Структурный анализ данных

Решения задач

- Отбор признаков
- Перебор параметров
- Задачи имеют нестандартную постановку
- Методы часто приходится придумывать самому

Полезные материалы и ссылки

Литература

- Курс лекций К.В. Воронцова по машинному обучению
- Вики-ресурс, посвященный машинному обучению
- Лекции Юрия Лифшица по структурам и алгоритмам для поиска ближайших соседей (на английском, хотя лектор из Питера)
- Классические лекции по машинному обучению (на английском)
- Отчет победителей соревнования по прогнозированию пробок
- Отчет по решению, занявшему третье место на соревновании по категоризации текстов
- Методическое пособие по Matlab и алгоритмам машинного обучения

Соревнования и задачи

- Классификация ирисов
- Категоризация текстов
- Распознавание пешеходов
- Прогнозирование пробок

Очень интересные статьи о решении реальных задач:

- Дьяконов А.Г. «Введение в анализ данных»
- Дьяконов А.Г. «Шаманство в анализе данных»