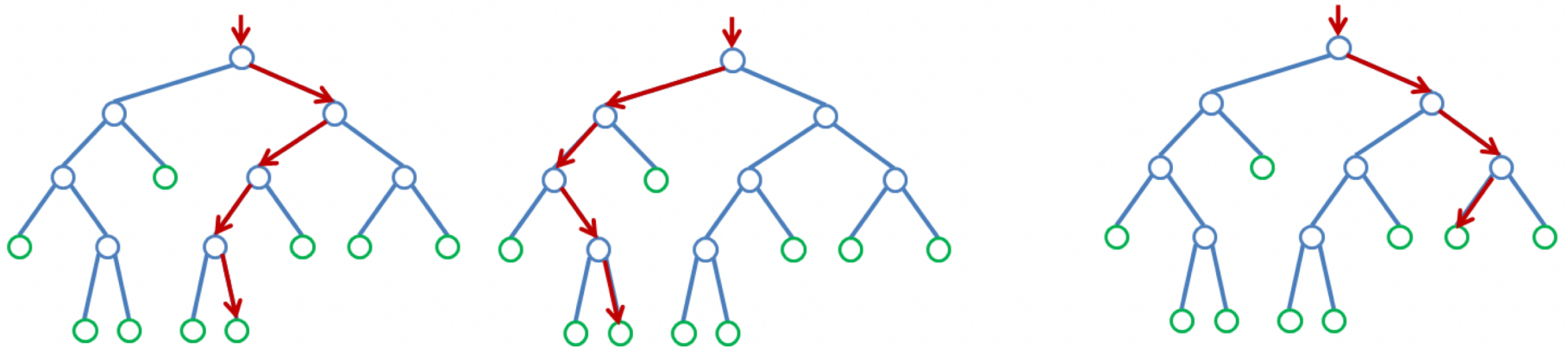
Open
Data
Science

Тема 5.
Виды решающих деревьев и метрические алгоритмы.

Еще немного о решающих деревьях

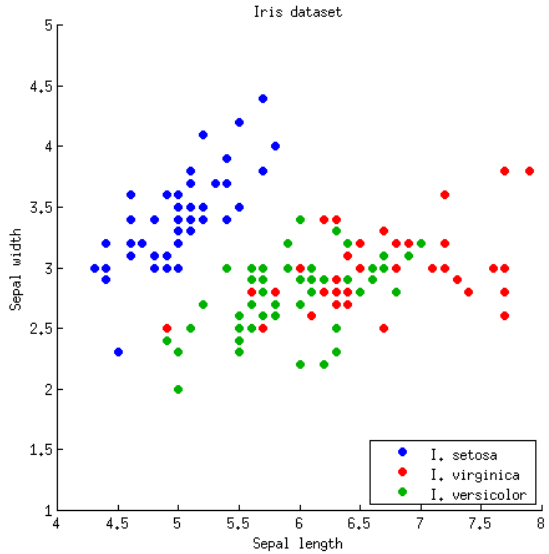
Признак (Возраст)	Признак (Город)	Целевой признак
17	Москва	0
20	Москва	0
23	Москва	1
26	Санкт-Петербург	1
27	Санкт-Петербург	0
35	Нижний Новгород	1

Extremely Randomized Trees



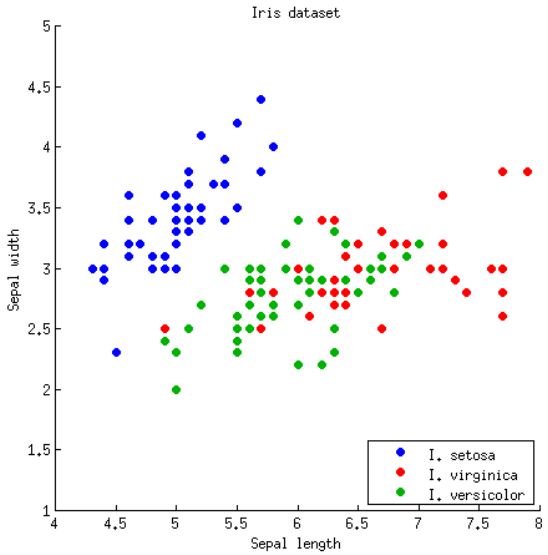
- Проверяем только один признак в узле
- Используем все признаки для построения дерева
- Работают в разы быстрее, чем стандартные деревья решений

Perfect random trees ensembles



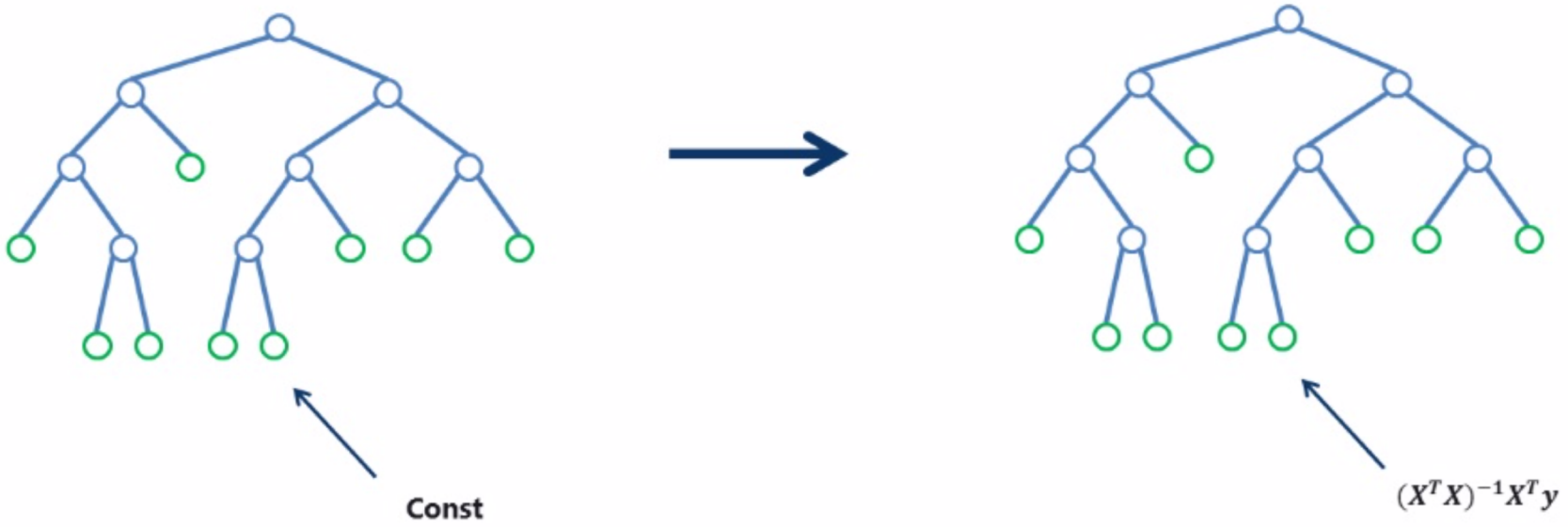
- Быстрое построение деревьев
- Качество сопоставимо с Random Forest
- Работает только с задачами классификации

Дерево решений с линейными признаками (вариант с сеткой)



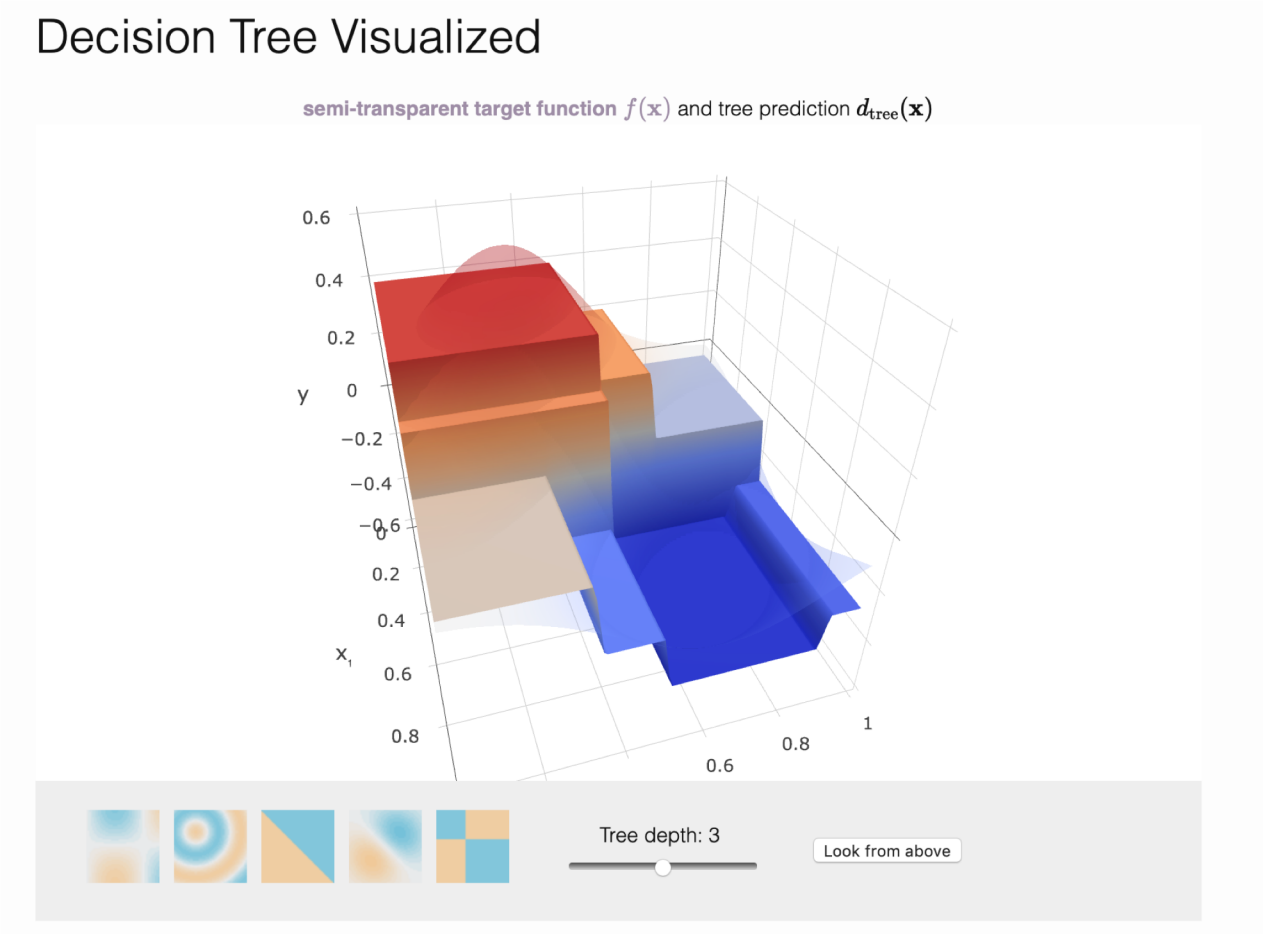
- Внутри узла перебираем не только существующие признаки, но и добавляем новые линейные признаки, построенные по сетке

Модельное дерево



- Поместим в узлы дерева решений другой алгоритм вместо констант

Поиграться с деревьями



http://arogozhnikov.github.io/2016/06/24/gradient_boosting_explained.html

Что делать если не можем создать матрицу признаков?

Задано:

- Графы
- Фотографии лиц
- Структуры белков

Что делать если не можем создать матрицу признаков?

Задано:

- Графы
- Фотографии лиц
- Структуры белков

+

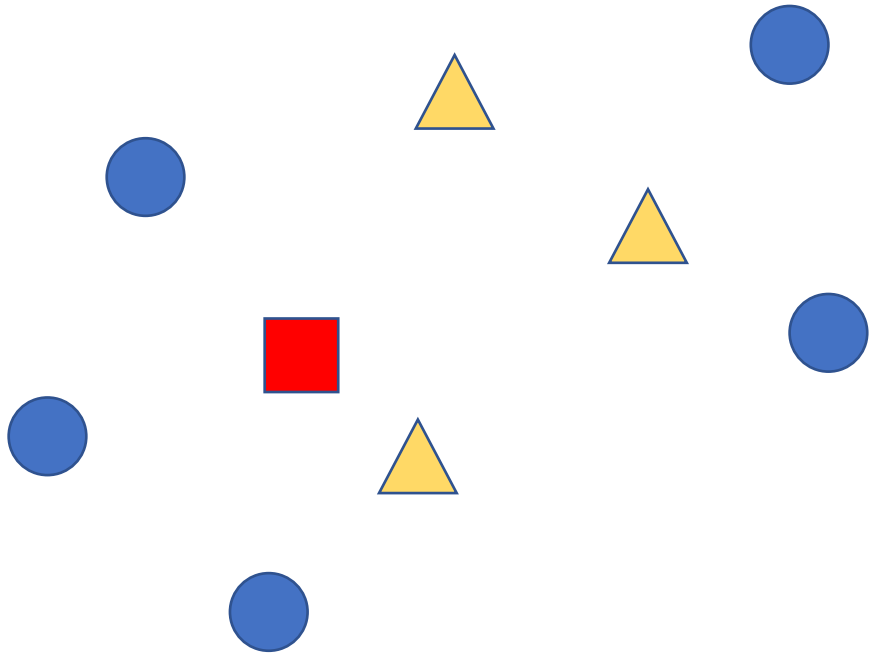
Умеем сравнивать объекты между собой

Введем меру сходства объектов

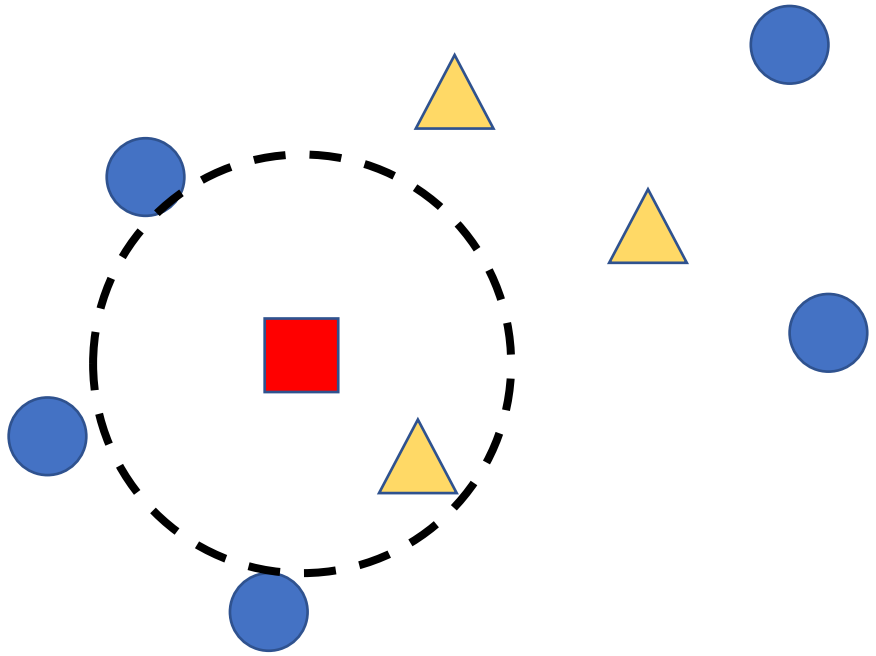
Метрика

Пусть на множестве объектов X задана функция расстояния ρ: $X \times X \longrightarrow [0; \infty)$
Существует целевая зависимость f: $X \longrightarrow Y$, значения которой известны только на объектах обучающей выборки $X_{train} = (x_i; y_i)_{i=1}^{train\ size}, y_i = f(x_i)$.
Множество классов Y конечно. Требуется построить алгоритм классификации p: $X \longrightarrow Y$ a : X → Y , аппроксимирующий целевую зависимость y∗(x) на всём множестве X.

Метод ближайшего соседа



Метод ближайшего соседа



Метод ближайшего соседа

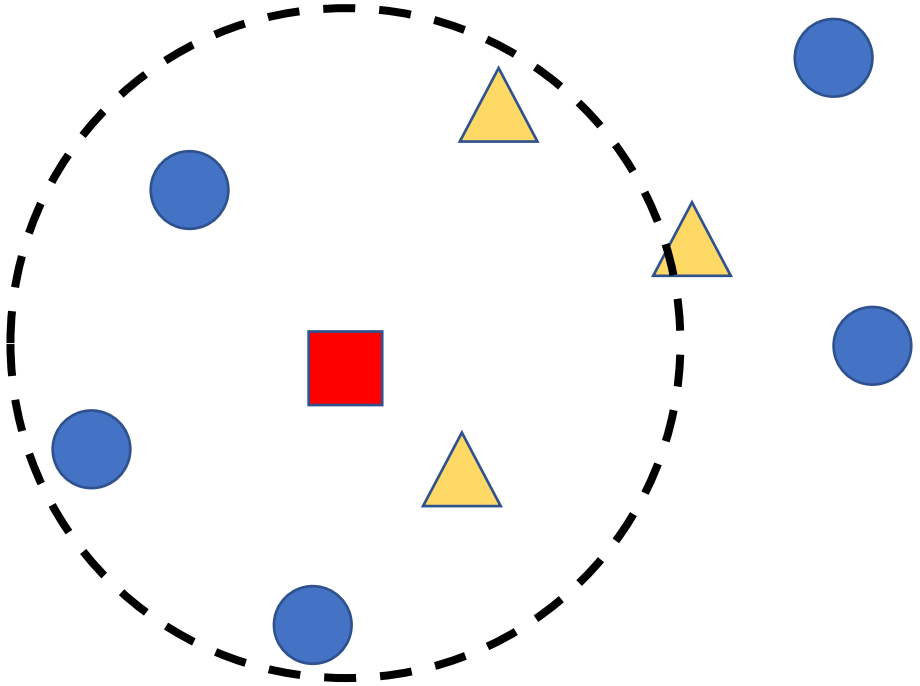
Плюсы

- Простая реализация
- Интерпретируемость

Минусы

- Неустойчивость к выбросам
- Мало гиперпараметров
- Низкое качество
- Надо хранить обучающую выборку

Метод ближайших соседей



Метод ближайших соседей

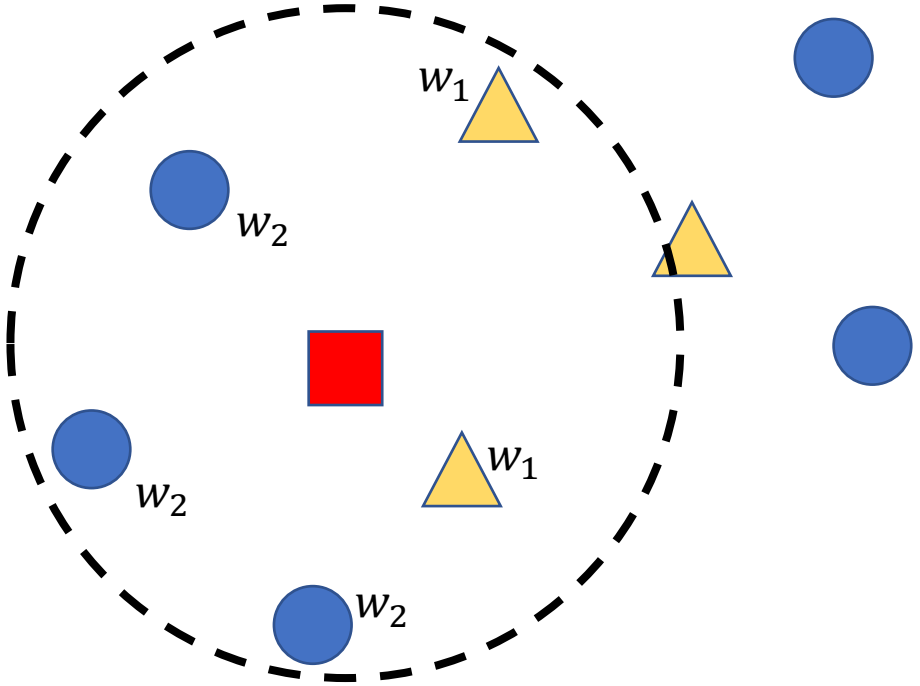
Плюсы

- Простая реализация
- Интерпретируемость

Минусы

- Проблемы с выбором класса
- Проблемы с несбалансированной выборкой
- Большой расход памяти
- Потенциально долгий перебор объектов

Метод ближайших соседей (с весами)



Метод ближайшего соседа (с весами)

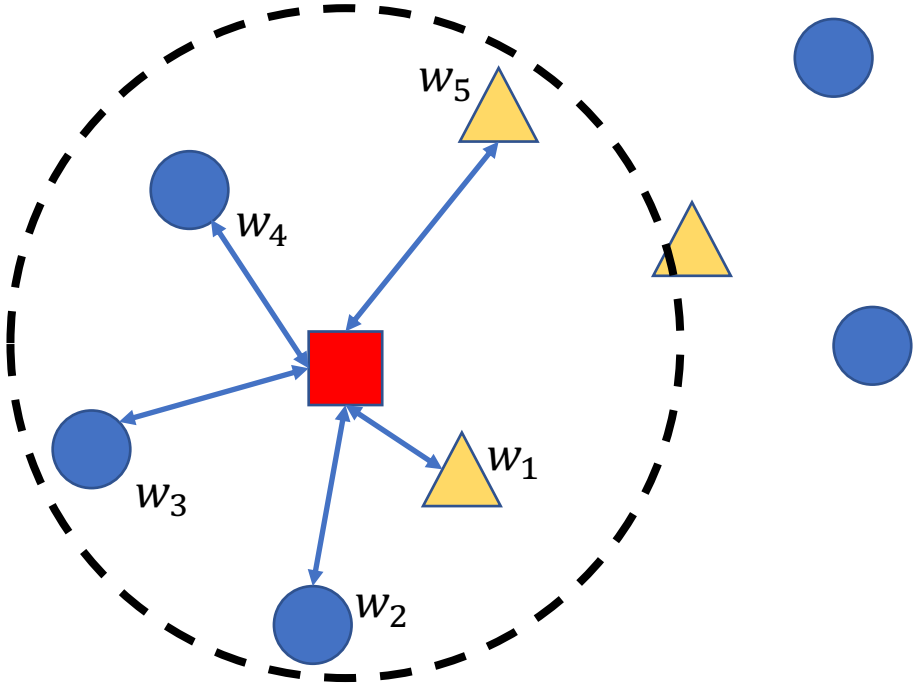
Плюсы

- Простая реализация
- Интерпретируемость

Минусы

- Большой расход памяти
- Потенциально долгий перебор объектов

Метод парзеновского окна



Метод парзеновского окна

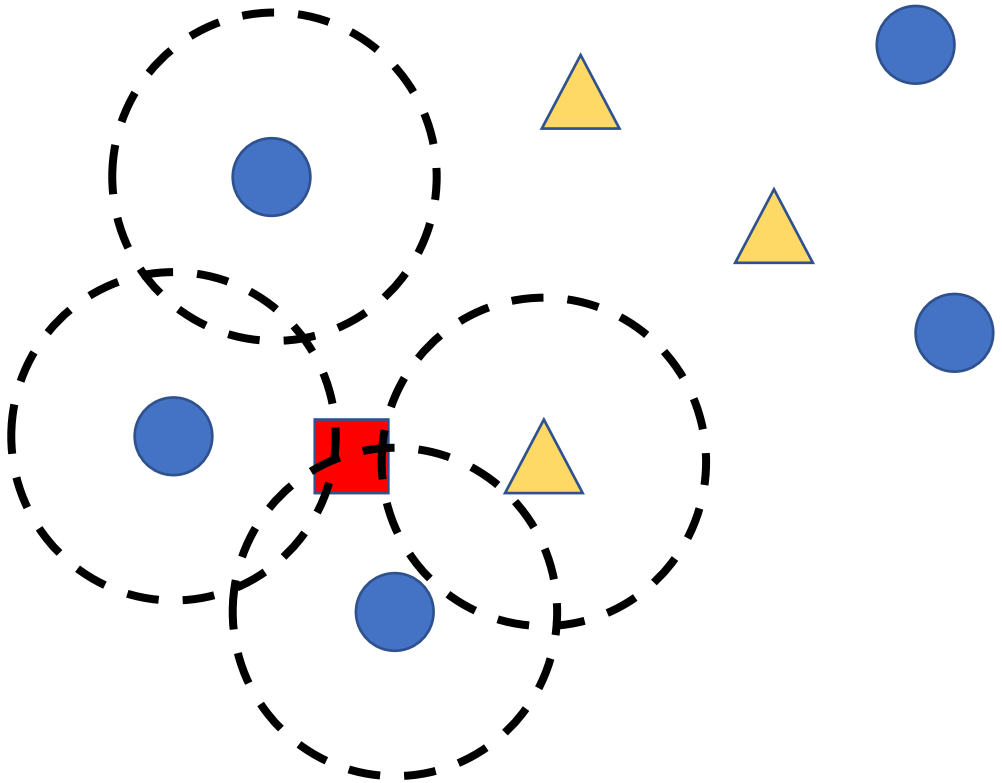
Окно постоянной ширины:

$$p(u; X_{train}; h; K) = \arg\max_{y \in Y} \sum_{i=1}^{train\ size} \left[y_{\blacksquare}^{(i)} = y \right] K\left(\frac{\rho(\blacksquare;\ x_{\blacksquare}^{(i)})}{h}\right)$$

Окно переменной ширины:

$$p(u; X_{train}; k; K) = \arg\max_{y \in Y} \sum_{i=1}^{train\ size} \left[y_{\blacksquare}^{(i)} = y \right] K\left(\frac{\rho(\blacksquare;\ x_{\blacksquare}^{(i)})}{\rho(\blacksquare;\ x_{\blacksquare}^{(k+1)})}\right)$$

Метод потенциальных функций



Проблемы метрических алгоритмов

Теорема. В n-мерном шаре весь объем сосредоточен на сфере при $n \to \infty$

Пример. Рассмотрим сферу в 20-мерном пространстве. Формула для объема сфера в 20-мерном пространстве: $V = \frac{\pi^{10}}{10!} R^{20}$. Найдем отношение объема шара радиуса 1 и радиуса 0.9: $\frac{V_{0.9}}{V_1} = \frac{0.9^{20}}{1} = 0.12$

Проблемы метрических алгоритмов

Пример 1. Рассмотрим единичный интервал [0,1]. 100 равномерно разбросанных точек будет достаточно, чтобы покрыть этот интервал с частотой не менее 0,01.

Пример 2. Теперь рассмотрим 10-мерный куб. Для достижения той же степени покрытия потребуется уже 10^{20} точек. То есть, по сравнению с одномерным пространством, требуется в 10^{18} раз больше точек.

Отступ объекта

Отступом объекта $x_i \in \mathrm{X}_{train}$ относительно алгоритма классификации, имеющего вид $p(u) = \arg\max_{y \in \mathrm{Y}} \Phi_y(u)$, называется величина:

$$M(x_i) = \Phi_{y_i}(x_i) - \max_{y \in \mathrm{Y} \setminus y_i} \Phi_y(x_i)$$

• Эталонные	$M \gg 0$	Все слишком хорошо
• Неинформативные	$M > 0$	Стандартная ситуация
• Пограничные	$M \sim 0$	Плохая метрика
• Ошибочные	$M < 0$	Плохой алгоритм
• Шумовые	$M \ll 0$	Плохая выборка

Виды алгоритмов классификации

Метрические алгоритмы

- KNN
- Ядерные алгоритмы

Алгоритмы, построенные на деревьях

- Деревья решений
- Random Forest
- Бустинг

Линейные алгоритмы

- Линейные модели
- Логистическая регрессия

Сложный случай для метрических алгоритмов

X1	X2	X3	X4	X5	Y
2	11	12	17	17	2
1	10	14	6	5	1
1	16	10	15	9	1
3	2	19	17	18	3

- Есть признак X1, который очень хорошо аппроксимирует целевую функцию
- Остальные признаки являются шумом
- При всех стандартных метриках будет большая ошибка

Классная работа

bit.ly/2IQem5C

Скачать лекцию