

Курсы по машинному обучению

Тема 7. Ошибки классификации и метрические алгоритмы

- Матрица ошибок (Confusion matrix)

	$y_{true} = 1$	$y_{true} = 0$
$y_{pred} = 1$	True Positive (TP)	False Positive (FP)
$y_{pred} = 0$	False Negative (FN)	True Negative (TN)

- Матрица ошибок (Confusion matrix)

	$y_{true} = 1$	$y_{true} = 0$
$y_{pred} = 1$	True Positive (TP)	Ошибка 2 рода (FP)
$y_{pred} = 0$	Ошибка 1 рода (FN)	True Negative (TN)

- Матрица ошибок (Confusion matrix)

	$y_{true} = 1$	$y_{true} = 0$
$y_{pred} = 1$	True Positive (TP)	False Positive (FP)
$y_{pred} = 0$	False Negative (FN)	True Negative (TN)

- Accuracy

$$accuracy = \frac{TP+TN}{TP+TN+FP+FN}$$

- Матрица ошибок (Confusion matrix)

Пусть 10 писем спам и 100 писем не спам

	$y_{true} = 1$	$y_{true} = 0$
$y_{pred} = 1$	5 (TP)	10 (FP)
$y_{pred} = 0$	5 (FN)	90 (TN)

- Accuracy

$$accuracy = \frac{TP+TN}{TP+TN+FP+FN} = \frac{5+90}{5+90+10+5} = 86.4$$

- Матрица ошибок (Confusion matrix)

Пусть 10 писем спам и 100 писем не спам

	$y_{true} = 1$	$y_{true} = 0$
$y_{pred} = 1$	0 (TP)	0 (FP)
$y_{pred} = 0$	10 (FN)	100 (TN)

- Accuracy

$$accuracy = \frac{TP+TN}{TP+TN+FP+FN} = \frac{5+90}{5+90+10+5} = 86.4$$

Если предсказывать всегда не спам:

$$accuracy = \frac{TP+TN}{TP+TN+FP+FN} = \frac{0+100}{0+100+0+10} = 90.9$$

- Матрица ошибок (Confusion matrix)

	$y_{true} = 1$	$y_{true} = 0$
$y_{pred} = 1$	True Positive (TP)	False Positive (FP)
$y_{pred} = 0$	False Negative (FN)	True Negative (TN)

- Precision (точность)

$$precision = \frac{TP}{TP+FP}$$

- Recall (полнота)

$$recall = \frac{TP}{TP+FN}$$

- F-мера

$$f_score = 2 \frac{precision * recall}{precision + recall} = (\beta^2 + 1) \frac{precision * recall}{\beta^2 precision + recall}$$

- Матрица ошибок (Confusion matrix)

Пусть 10 писем спам и 100 писем не спам

	$y_{true} = 1$	$y_{true} = 0$
$y_{pred} = 1$	5 (TP)	10 (FP)
$y_{pred} = 0$	5 (FN)	90 (TN)

$$precision = \frac{TP}{TP+FP} = \frac{5}{5+10} = 0.33$$

$$recall = \frac{TP}{TP+FN} = \frac{5}{5+5} = 0.5$$

$$f_score = 2 \frac{precision*recall}{precision+recall} = 2 \frac{0.33*0.5}{0.33+0.5} = 0.39$$

- Матрица ошибок (Confusion matrix)

Пусть 10 писем спам и 100 писем не спам

	$y_{true} = 1$	$y_{true} = 0$
$y_{pred} = 1$	5 (TP)	10 (FP)
$y_{pred} = 0$	5 (FN)	90 (TN)

- True Positive Rate

$$TPR = \frac{TP}{TP+FN} = \frac{5}{5+5} = 0.5$$

- False Positive Rate

$$FPR = \frac{FP}{FP+TN} = \frac{10}{10+90} = 0.1$$

ID	Вероятность
1	0.99
2	0.9
3	0.51
4	0.49
5	0.44

- True Positive Rate

$$TPR = \frac{TP}{TP+FN}$$

- False Positive Rate

$$FPR = \frac{FP}{FP+TN}$$

ID	Вероятность
1	0.99
2	0.9
3	0.51
4	0.49
5	0.44

ID	Вероятность
1	0.99
2	0.9
3	0.51
4	0.49
5	0.44

ID	Вероятность
1	0.99
2	0.9
3	0.51
4	0.49
5	0.44

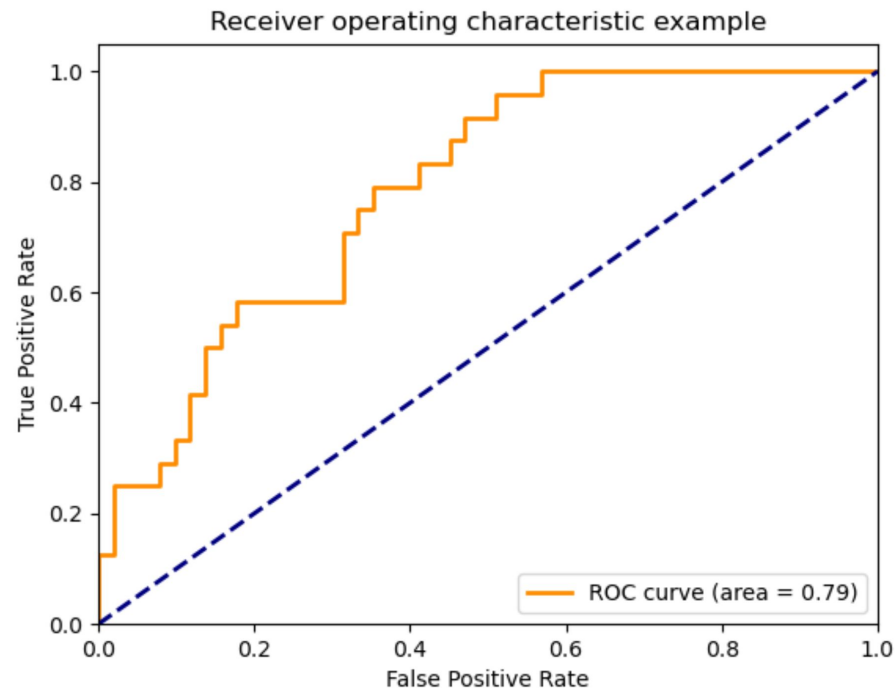
- True Positive Rate

$$TPR = \frac{TP}{TP+FN}$$

- False Positive Rate

$$FPR = \frac{FP}{FP+TN}$$

- Построение ROC AUC
 - а. Для каждого порога посчитать TPR и FPR
 - б. Построить график в осях TPR/FPR
- Кривая на графике: ROC curve
- Площадь под кривой: ROC AUC



Что делать если классов несколько?

- micro-метрики
 - Считаем общую матрицу ошибок
 - По ним вычисляем метрики
- macro-метрики
 - Вычисляем метрики на каждом классе “1 против всех”
 - Берем среднее
- weighted-метрики
 - Вычисляем метрики на каждом классе “1 против всех”
 - Берем средневзвешенное по классам
 - **Проблема:** F-мера может не лежать между Precision и Recall

Что делать если не можем создать матрицу признаков?

Задано:

- Графы
- Фотографии лиц
- Подписи
- Временные ряды
- Структуры белков

Что делать если не можем создать матрицу признаков?

Задано:

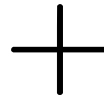
- Графы
- Фотографии лиц
- Подписи
- Временные ряды
- Структуры белков

Умеем сравнивать
объекты между собой

Что делать если не можем создать матрицу признаков?

Задано:

- Графы
- Фотографии лиц
- Подписи
- Временные ряды
- Структуры белков



Умеем сравнивать
объекты между собой

Введем метрику сходства объектов

Неотрицательная функция называется метрикой, если:

1. $d(x, y) = 0 \Leftrightarrow x = y$

2. $d(x, y) = d(y, x)$

3. $d(x, z) \leq d(x, y) + d(y, z)$

Неотрицательная функция называется метрикой, если:

1. $d(x, y) = 0 \Leftrightarrow x = y$
2. $d(x, y) = d(y, x)$
3. $d(x, z) \leq d(x, y) + d(y, z)$

Примеры:

$$d(x, y) = |x - y|$$

Числа

$$d(x, y) = \sqrt{\sum (x_i - y_i)^2}$$

Вектора

$$d(f, g) = \sup |f(x) - g(x)|$$

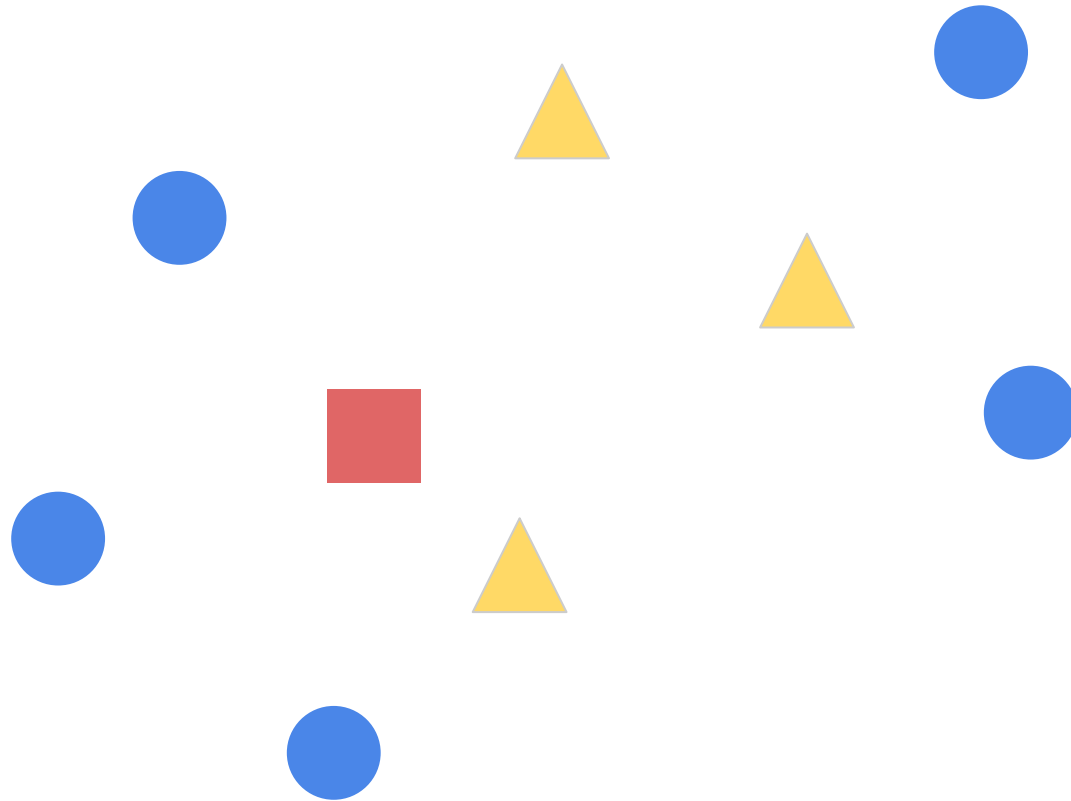
Функции

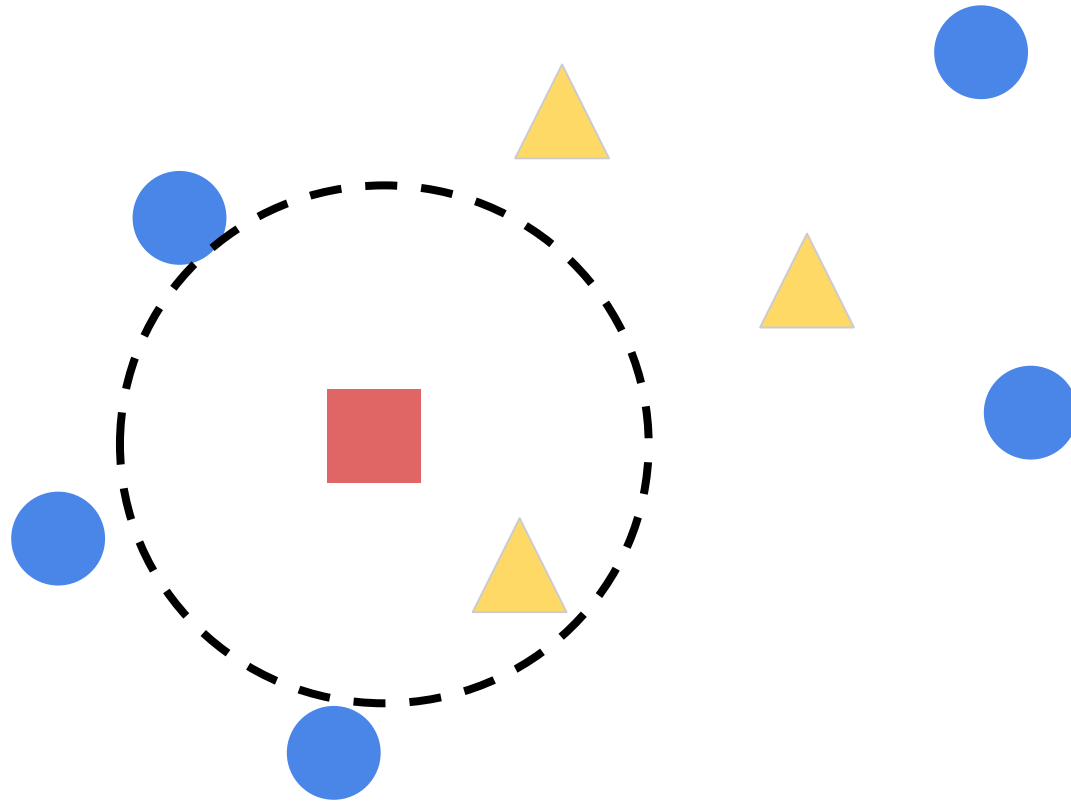
Гипотеза компактности — в задачах классификации предположение о том, что схожие объекты гораздо чаще лежат в одном классе, чем в разных

Другими словами, что классы образуют компактно локализованные подмножества в пространстве объектов.

Это также означает, что граница между классами имеет достаточно простую форму.

* Компактные множества тут не при чем





Определяем самый ближний элемент и класс нашего объекта будет совпадать с ближайшим

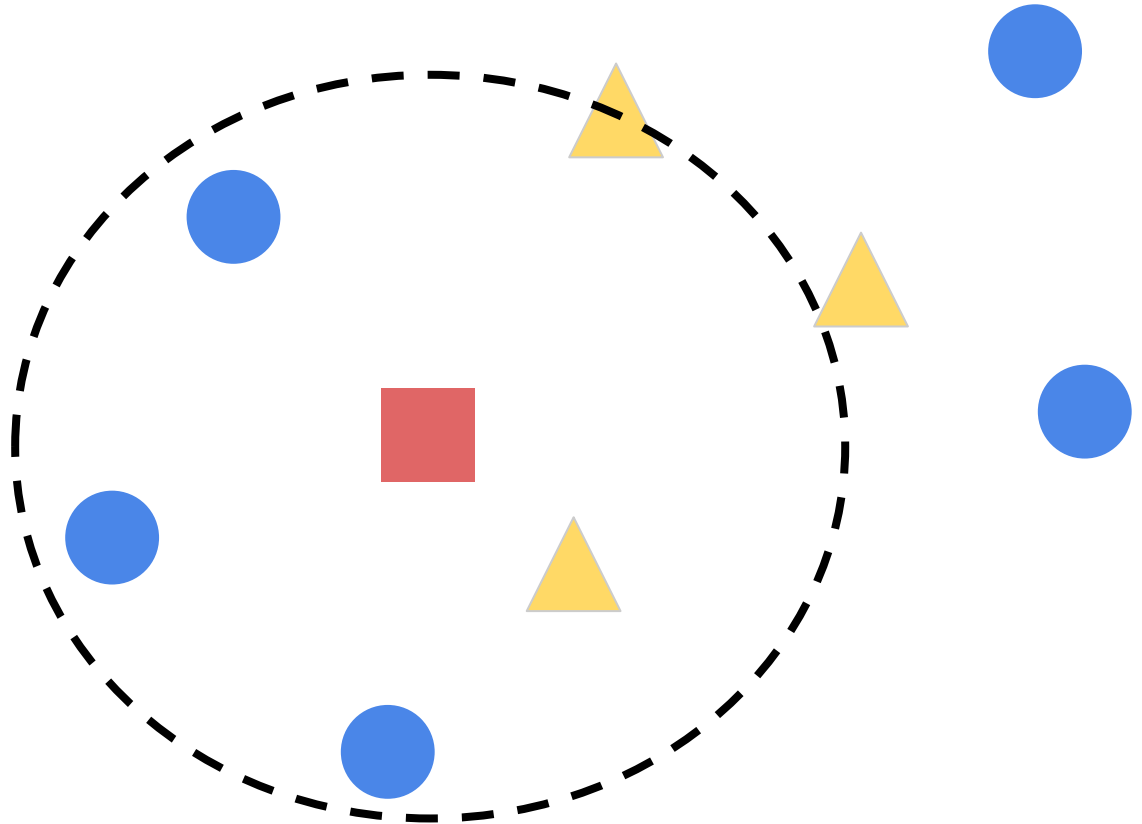
Плюсы

- Простая реализация
- Интерпретируемость

Минусы

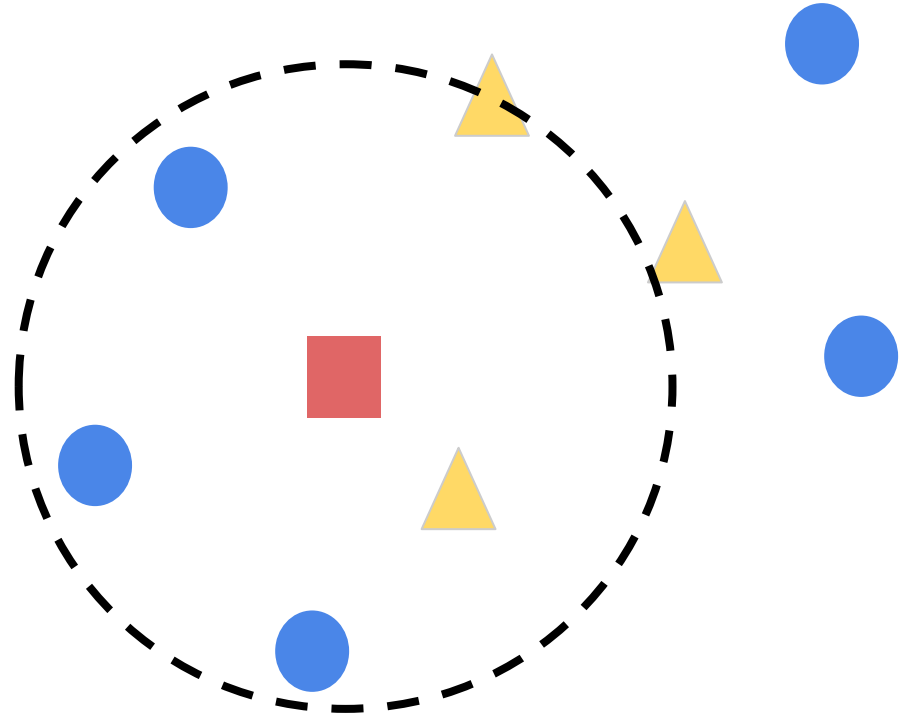
- Неустойчивость к выбросам
- Мало гиперпараметров
- Полная зависимость от метрики
- Низкое качество

$$\arg \max_{y \in Y} \sum_{i=1}^k [y_u^{(i)} = y]$$



Определяем k самых ближайших элементов и класс нашего объекта будет выбран с помощью голосования, аналогично голосованию в случайном лесу

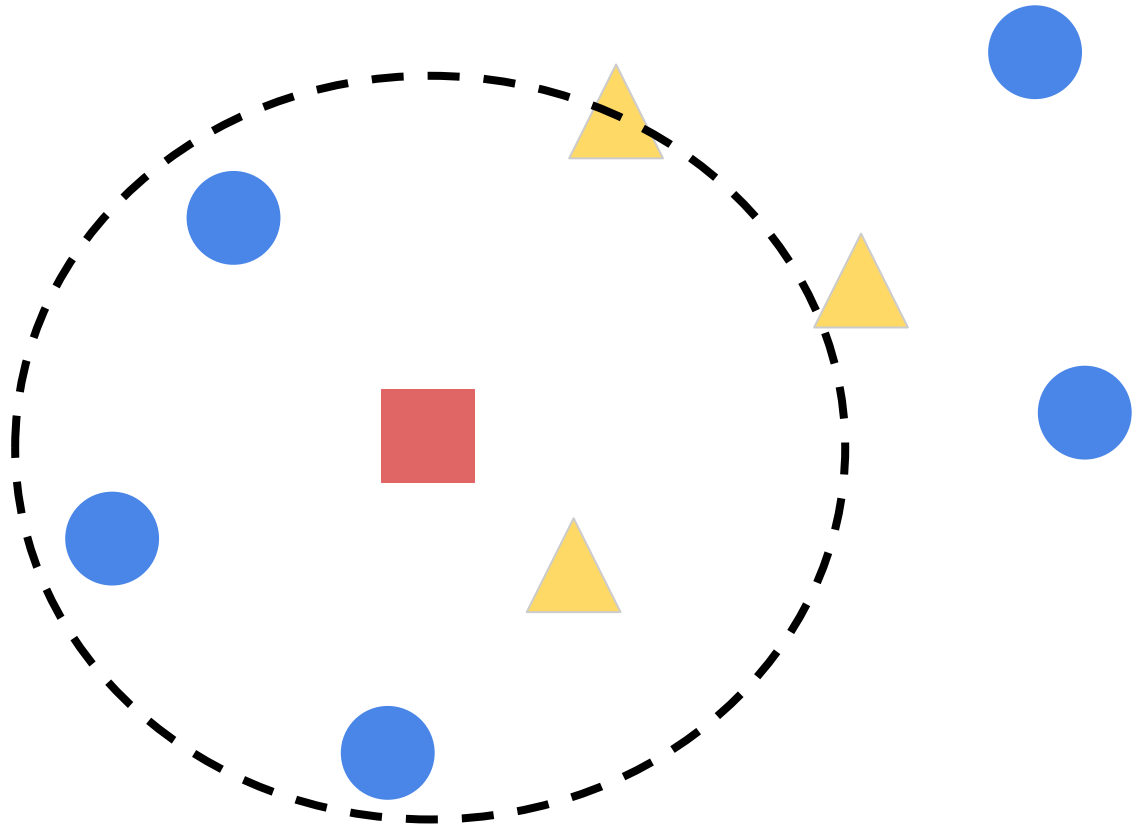
- Выбор класса производится голосованием
- $k=1$ -> метод ближайшего соседа
- $k=L$ -> метод константа (для всех объектов будет выбран преобладающий класс)
- Подбор параметра по критерию скользящего контроля с исключением объектов по одному (leave-one-out)
- Также можно сравнивать средние расстояния до объектов другого класса



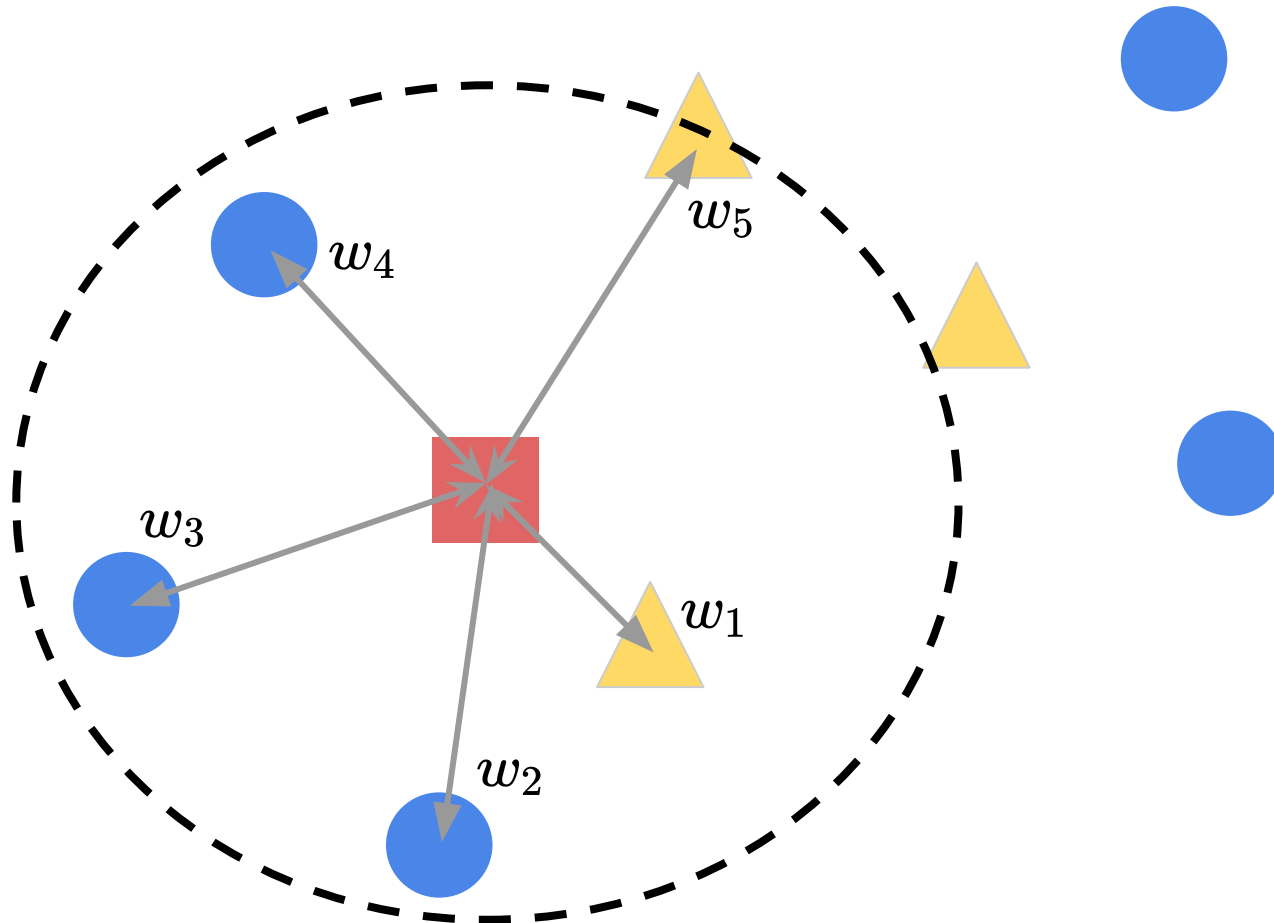
Что делать если голосование не выявило лидирующий класс?

- Для двух классов брать нечетные k
- Использовать веса
- Выбирать случайно

$$\arg \max_{y \in Y} \sum_{i=1}^k \left[y_u^{(i)} = y \right] w_i$$



Определяем k самых ближайших элементов и класс нашего объекта будет выбран с помощью голосования, но с учетом веса каждого элемента

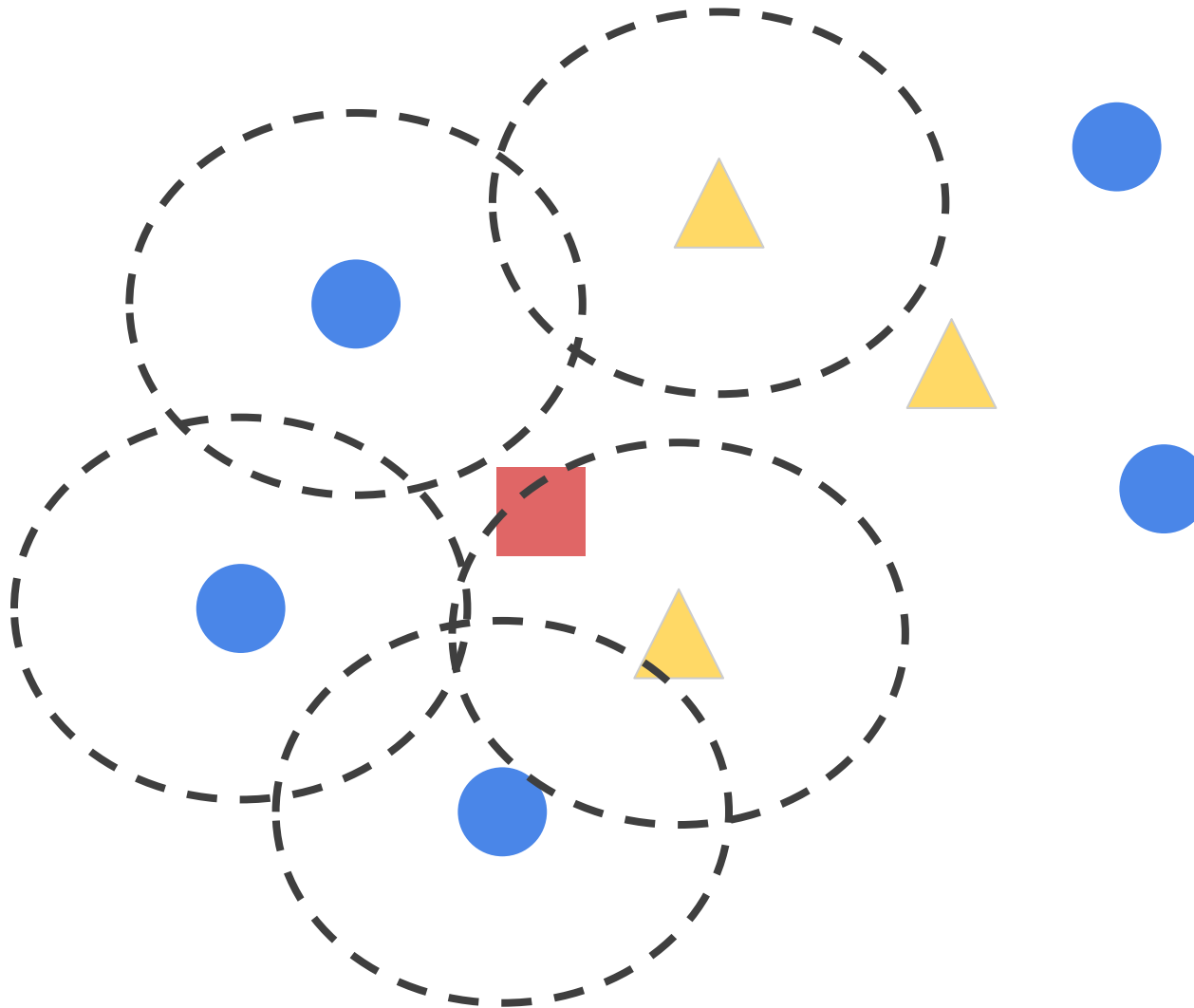


- $\rho(u, x_u^{(i)})$ – функция расстояния до соседа
- Функция ядра $K(z)$ – не возрастает на $[0, \infty)$
- Постоянная ширина окна

$$a(u; X^l, h, K) = \arg \max_{y \in Y} \sum_{i=1}^l [y_u^{(i)} = y] K \left(\frac{\rho(u, x_u^{(i)})}{h} \right)$$

- Переменная ширина окна (неравномерное распределение)

$$a(u; X^l, k, K) = \arg \max_{y \in Y} \sum_{i=1}^k [y_u^{(i)} = y] K \left(\frac{\rho(u, x_u^{(i)})}{\rho(u, x_u^{(k+1)})} \right)$$



- Выбор числа соседей k
- Отсев шума
- Большие выборки
- При погрешностях (в данных или метрике) снижается точность по границам классов
- Поиск ближайшего соседа = сравнение с каждым элементом выборки
- Проблема выбора метрики
- Проклятие размерности
- Малое число параметров

- Отступ объекта $x_i \in X^i$ относительно алгоритма классификации, имеющего вид $a(u) = \arg \max_{y \in Y} \Gamma_y(u)$, называется величина

$$M(x_i) = \Gamma_{y_i}(x_i) - \max_{y \in Y \setminus y_i} \Gamma_y(x_i)$$

Эталонные	$M \gg 0$	По ним можно классифицировать
Неинформативные	$M > 0$	Не несут важной информации
Пограничные	$M \sim 0$	Чувствительны к изменениям
Ошибочные	$M < 0$	Ошибка в метрике
Шумовые (выбросы)	$M \ll 0$	Ошибка в данных

- В n-мерном шаре весь объем сосредоточен на сфере
- Пример. Рассмотрим сферу в 20-мерном пространстве.
Формула для объема шара в 20-мерном пространстве:

$$V = \frac{\pi^{10} \pi}{10!} R^{20}$$

Найдем отношение объема шара радиуса 1 и радиуса 0.9:

$$\frac{V_{0,9}}{V_1} = \frac{0,9^{20}}{1} = 0.12$$

- 88% точек лежит на сфере, а значит почти до всех ближайших соседей расстояние одинаковое

- Также данные крайне разреженные
- 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 10^{20} достижения той же степени покрытия потребуется уже точек. То есть, по сравнению с одномерным пространством, требуется в 10^{18} раз больше точек.

Решение - снижение размерности.
Например, PCA - метод главных компонент

