Простые нелинейные модели. Многоклассовая классификация

Елена Кантонистова

ПЛАН ЛЕКЦИИ

- Простые нелинейные классификаторы:

- Наивный байесовский классификатор

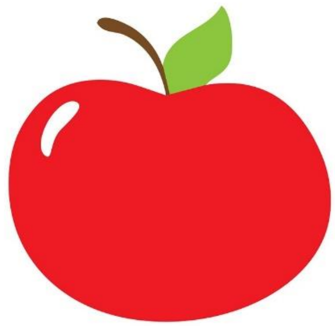
- Метод ближайших соседей

- Многоклассовая классификация

НАИВНЫЙ БАЙЕСОВСКИЙ КЛАССИФИКАТОР

НАИВНЫЙ БАЙЕСОВСКИЙ КЛАССИФИКАТОР

Наивный байесовский классификатор – это алгоритм классификации, основанный на теореме Байеса с допущением о независимости признаков.



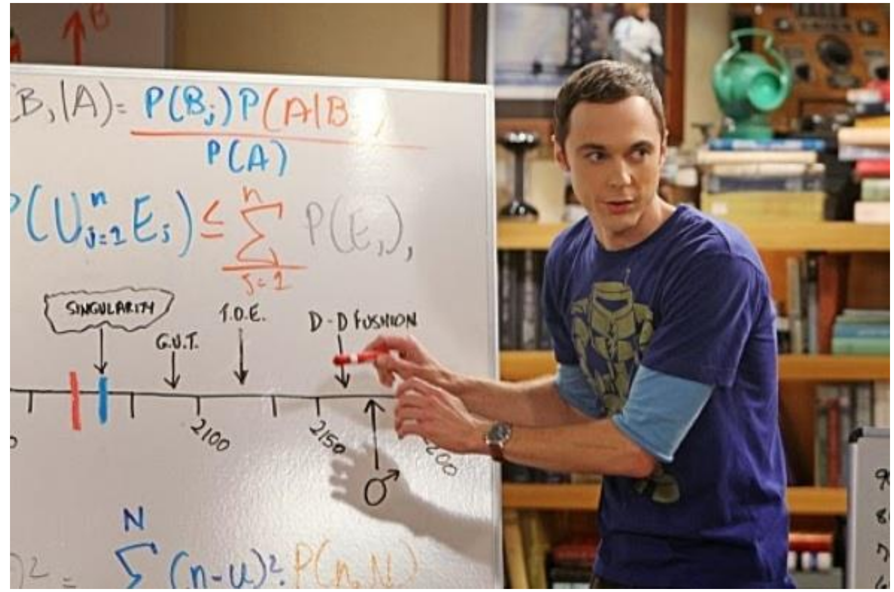
Пример: фрукт может считаться яблоком,

если:

1) он красный
2) круглый
3) его диаметр составляет порядка 8 см

Предполагаем, что признаки вносят независимый вклад в вероятность того, что фрукт является яблоком.

ТЕОРЕМА БАЙЕСА



<u>Теорема Байеса:</u>

$$P(c|x) = \frac{P(x|c) \cdot P(c)}{P(x)}$$

- $\boldsymbol{P(c|x)}$ - вероятность того,

что объект со значением признака x

принадлежит классу c.

- $\boldsymbol{P(c)}$ **–** априорная вероятность класса c.
- $\boldsymbol{P(x|c)}$ - вероятность того, что значение признака равно x при условии, что объект принадлежит классу c.
- $\boldsymbol{P(x)}$ **–** априорная вероятность значения признака x.

ПРИМЕР РАБОТЫ БАЙЕСОВСКОГО АЛГОРИТМА

Пример: на основе данных о погодных условиях необходимо определить, состоится ли матч.

Weather	Play
Sunny	No
Overcast	Yes
Rainy	Yes
Sunny	Yes
Sunny	Yes
Overcast	Yes
Rainy	No
Rainy	No
Sunny	Yes
Rainy	Yes
Sunny	No
Overcast	Yes
Overcast	Yes
Rainy	No

- Преобразуем набор данных в следующую таблицу:

Weather	No	Yes
Overcast	0	4
Rainy	3	2
Sunny	2	3
Grand Total	5	9

ПРИМЕР РАБОТЫ БАЙЕСОВСКОГО АЛГОРИТМА

Решим задачу с помощью теоремы Байеса:

$$P(Yes|Sunny) = P(Sunny|Yes) \cdot P(Yes)/P(Sunny)$$

Таблица частот

Weather	**No**	**Yes**		
Overcast	0	4	**=4/14**	**0.29**
Rainy	3	2	**=5/14**	**0.36**
Sunny	2	3	**=5/14**	**0.36**
Grand Total	5	9		
	=5/14	**=9/14**		
	0.36	**0.64**		

- $P(Sunny|Yes) = \frac{3}{9}, P(Sunny) = \frac{5}{14}, P(Yes) = \frac{9}{14}.$
- $P(Yes|Sunny) = \frac{3}{9} \cdot \frac{9}{14} : \frac{5}{14} = \frac{3}{5} = 0{,}6 \Rightarrow 60\%.$

В СЛУЧАЕ НЕСКОЛЬКИХ ПРИЗНАКОВ

Пусть $x_1, \ldots, x_n$ - признаки объекта, y – целевая переменная.

Тогда теорема Байеса записывается в виде

$$P(y|x_1, \ldots, x_n) = \frac{P(x_1|y)P(x_2|y)\ldots P(x_n|y)P(y)}{P(x_1)P(x_2)\ldots P(x_n)}.$$

Вероятности в правой части формулы вычисляются с помощью частотных таблиц, как и в одномерном случае.

почитать статью про Байесовский классификатор

БАЙЕСОВСКИЙ АЛГОРИТМ ДЛЯ КЛАССИФИКАЦИИ

Плюсы и минусы:

+ классификация быстрая и простая

+ в случае, если выполняется предположение о независимости, классификатор показывает очень высокое качество

- если в тестовых данных присутствует категория, не встречавшаяся в данных для обучения, модель присвоит ей нулевую вероятность

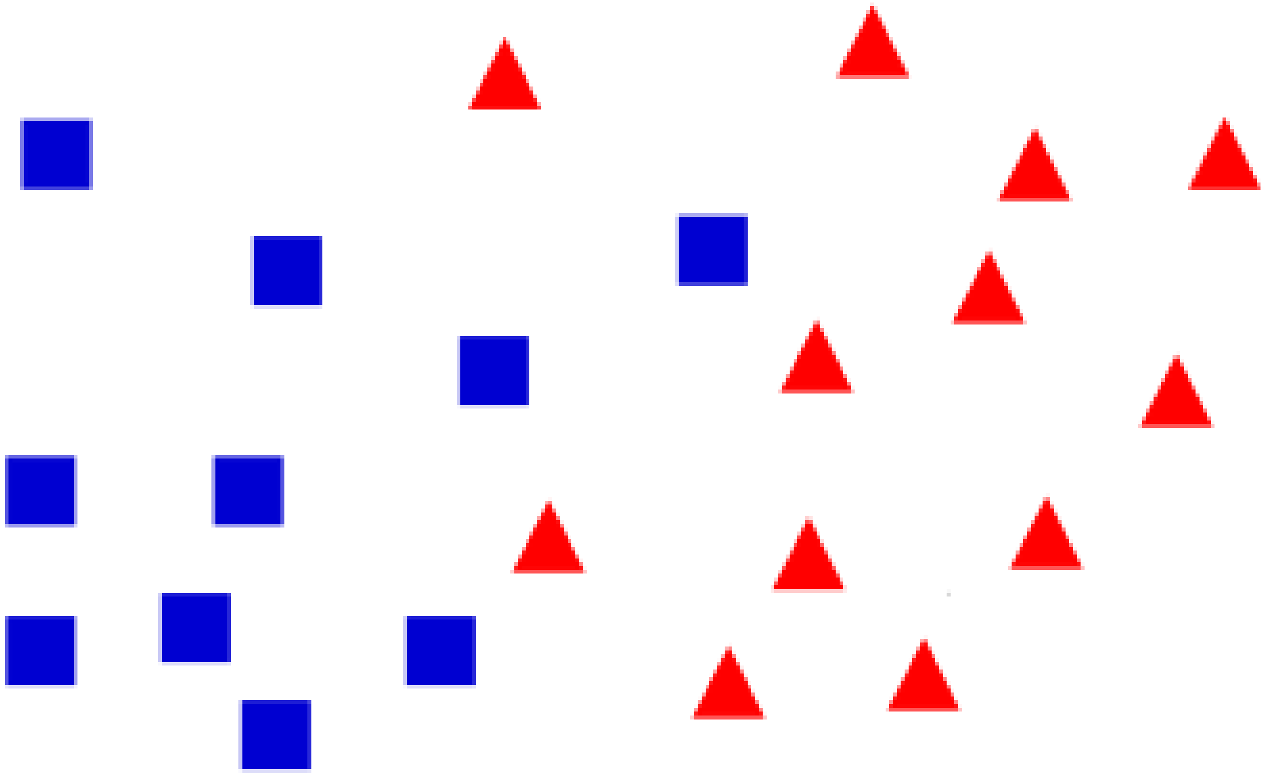
НАИВНЫЙ БАЙЕСОВСКИЙ АЛГОРИТМ

https://scikit-learn.org/stable/modules/naive_bayes.html

МЕТОД БЛИЖАЙШИХ СОСЕДЕЙ

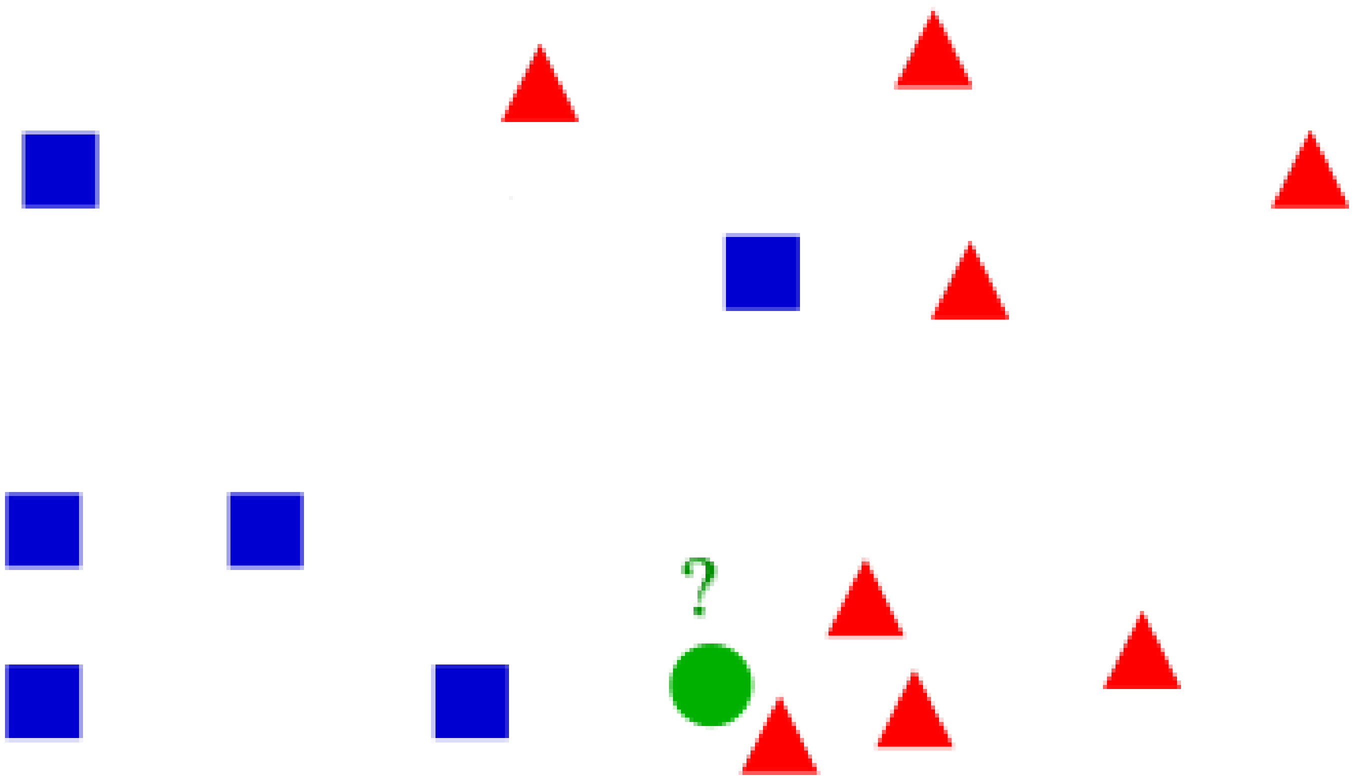
МЕТОД БЛИЖАЙШИХ СОСЕДЕЙ

Идея: схожие объекты находятся близко друг к другу в пространстве признаков.



МЕТОД БЛИЖАЙШИХ СОСЕДЕЙ

Как классифицировать новый объект?



МЕТОД БЛИЖАЙШИХ СОСЕДЕЙ

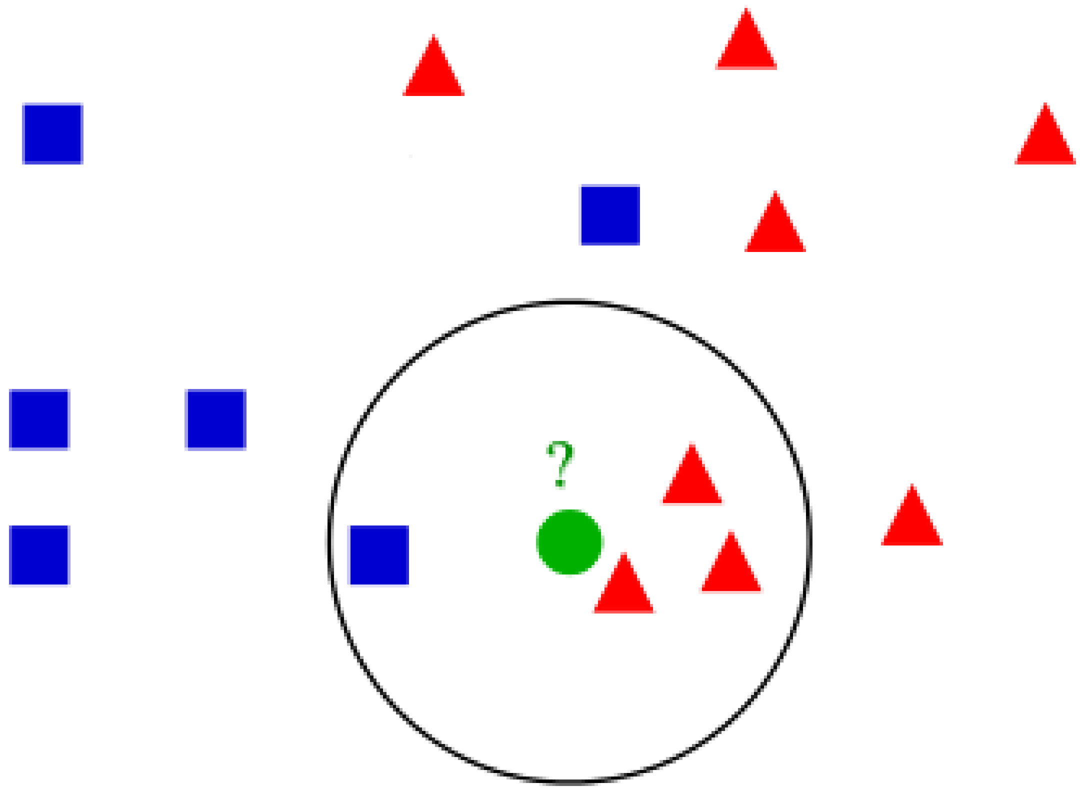
Чтобы классифицировать новый объект, нужно:

- Вычислить расстояние до каждого из объектов обучающей выборки.
- Выбрать k объектов обучающей выборки, расстояние до которых минимально.
- Класс классифицируемого объекта — это класс, наиболее часто встречающийся среди k ближайших соседей.

МЕТОД БЛИЖАЙШИХ СОСЕДЕЙ

Число ближайших соседей k – гиперпараметр метода.

Например, для $k = 4$ получим:



То есть объект будет отнесён к классу *треугольников*.

ФОРМАЛИЗАЦИЯ МЕТОДА

Пусть k – количество соседей. Для каждого объекта u возьмём k ближайших к нему объектов из тренировочной выборки:

$$x_{(1;u)}, x_{(2;u)}, \dots, x_{(k;u)}.$$

Тогда класс объекта u определяется следующим образом:

$$\boldsymbol{a(u) = \mathop{\mathbf{argmax}}_{y \in Y} \sum_{i=1}^{k} \left[y\left(x_{(i;u)}\right) = y \right]}.$$

ФОРМАЛИЗАЦИЯ МЕТОДА

Пусть k – количество соседей. Для каждого объекта u возьмём k ближайших к нему объектов из тренировочной выборки:

$$x_{(1;u)}, x_{(2;u)}, \dots, x_{(k;u)}.$$

Тогда класс объекта u определяется следующим образом:

$$a(u) = \operatorname*{argmax}_{y \in Y} \sum_{i=1}^{k} [y(x_{(i;u)}) = y].$$

Ближайшие объекты – это объекты, расстояние от которых до данного объекта наименьшее по некоторой метрике ρ.

ФОРМАЛИЗАЦИЯ МЕТОДА

Пусть k – количество соседей. Для каждого объекта u возьмём k ближайших к нему объектов из тренировочной выборки:

$$x_{(1;u)}, x_{(2;u)}, \dots, x_{(k;u)}.$$

Тогда класс объекта u определяется следующим образом:

$$a(u) = \underset{y \in Y}{\operatorname{argmax}} \sum_{i=1}^{k} [y(x_{(i;u)}) = y].$$

Ближайшие объекты – это объекты, расстояние от которых до данного объекта наименьшее по некоторой метрике ρ.

- В качестве метрики ρ как правило используют евклидово расстояние, но можно использовать и другие метрики.
- Перед использованием метода необходимо масштабировать данные, иначе признаки с большими числовыми значениями будут доминировать при вычислении расстояний.

МНОГОКЛАССОВАЯ КЛАССИФИКАЦИЯ

ПОДХОД ONE-VS-ALL

Решаем задачу классификации на K классов.

- Обучим K бинарных классификаторов $b_1(x), \ldots, b_K(x)$, каждый из которых решает задачу: *принадлежит объект* x *к классу* k_i *или не принадлежит*?

Например, линейные классификаторы будут иметь вид

$$b_k(x) = sign((w_k, x) + w_{0k})$$

ПОДХОД ONE-VS-ALL

Решаем задачу классификации на K классов.

- Обучим K бинарных классификаторов $b_1(x), \dots, b_K(x)$, каждый из которых решает задачу: *принадлежит объект x к классу k_i или не принадлежит?*

Например, линейные классификаторы будут иметь вид

$$b_k(x) = sign((w_k, x) + w_{0k})$$

- Тогда в качестве итогового предсказания будем выдавать класс самого уверенного классификатора:

$$a(x) = \underset{k \in \{1,\dots,K\}}{arg\max}((w_k, x) + w_{0k})$$

ПОДХОД ONE-VS-ALL

Решаем задачу классификации на K классов.

- Обучим K бинарных классификаторов $b_1(x), \dots, b_k(x)$, каждый из которых решает задачу: *принадлежит объект x к классу k_i или не принадлежит?*

Например, линейные классификаторы будут иметь вид

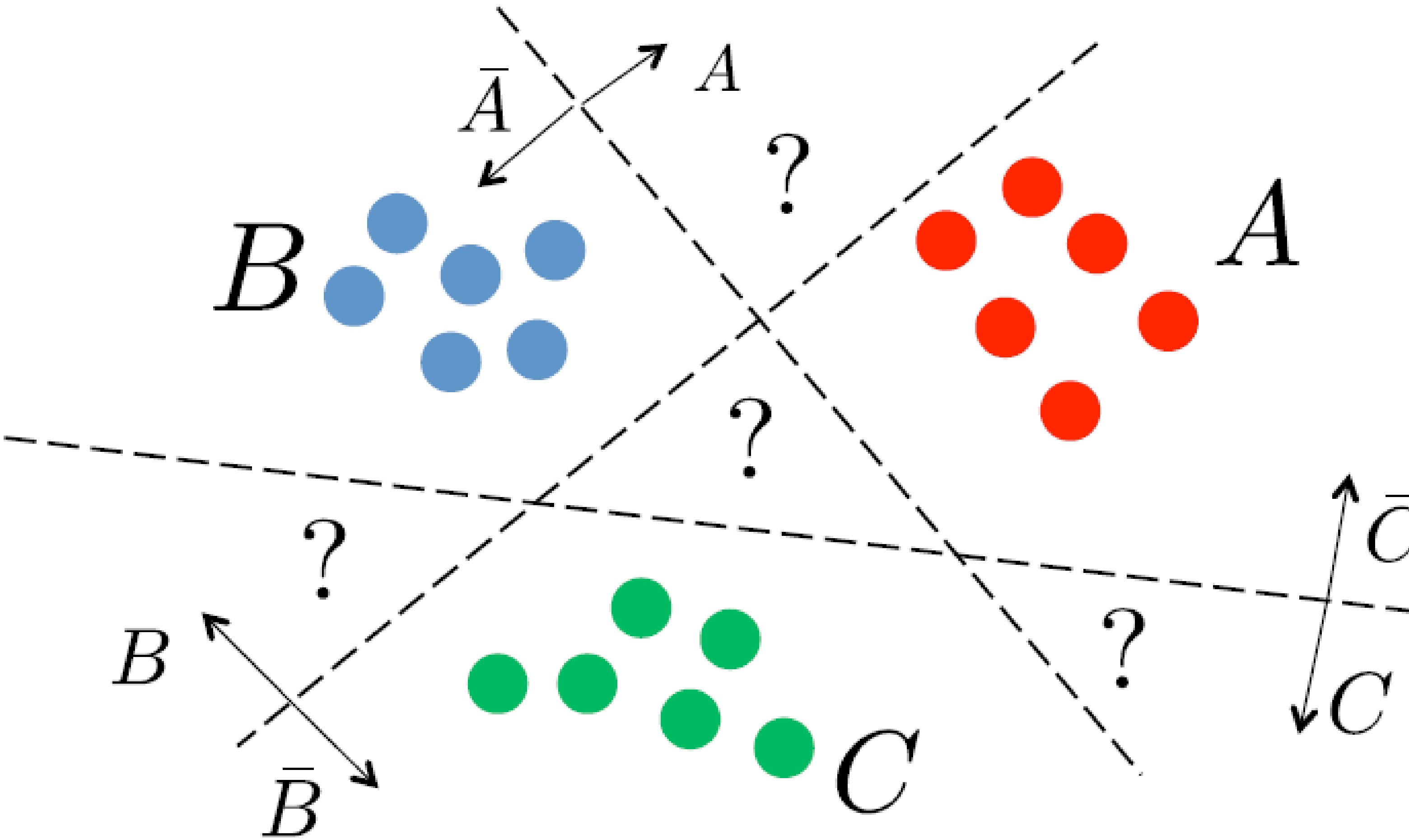
$$b_k(x) = sign((w_k, x) + w_{0k})$$

- Тогда в качестве итогового предсказания будем выдавать класс самого уверенного классификатора:

$$a(x) = \underset{k \in \{1,\dots,K\}}{arg\max}((w_k, x) + w_{0k})$$

- Предсказания классификаторов могут иметь разные масштабы, поэтому сравнивать их некорректно.

ПОДХОД ONE-VS-ALL



ПОДХОД ALL-VS-ALL

- Для каждой пары классов i и j обучим бинарный классификатор $a_{ij}(x)$, который будет предсказывать класс i или j

(если всего K классов, то получим C_K^2 классификаторов).

Каждый такой классификатор будем обучать только на объектах классов i и j.

ПОДХОД ALL-VS-ALL

- Для каждой пары классов i и j обучим бинарный классификатор $a_{ij}(x)$, который будет предсказывать класс i или j

(если всего K классов, то получим C_K^2 классификаторов).

Каждый такой классификатор будем обучать только на объектах классов i и j.

- В качестве итогового предсказания выдадим класс, который предсказало наибольшее число алгоритмов:

$$a(x) = \underset{k \in \{1,\dots,K\}}{\operatorname{argmax}} \sum_{i=1}^{K} \sum_{j \neq i} [a_{ij}(x) = k]$$

ПОДХОД ALL-VS-ALL

- Для каждой пары классов i и j обучим бинарный классификатор $a_{ij}(x)$, который будет предсказывать класс i или j

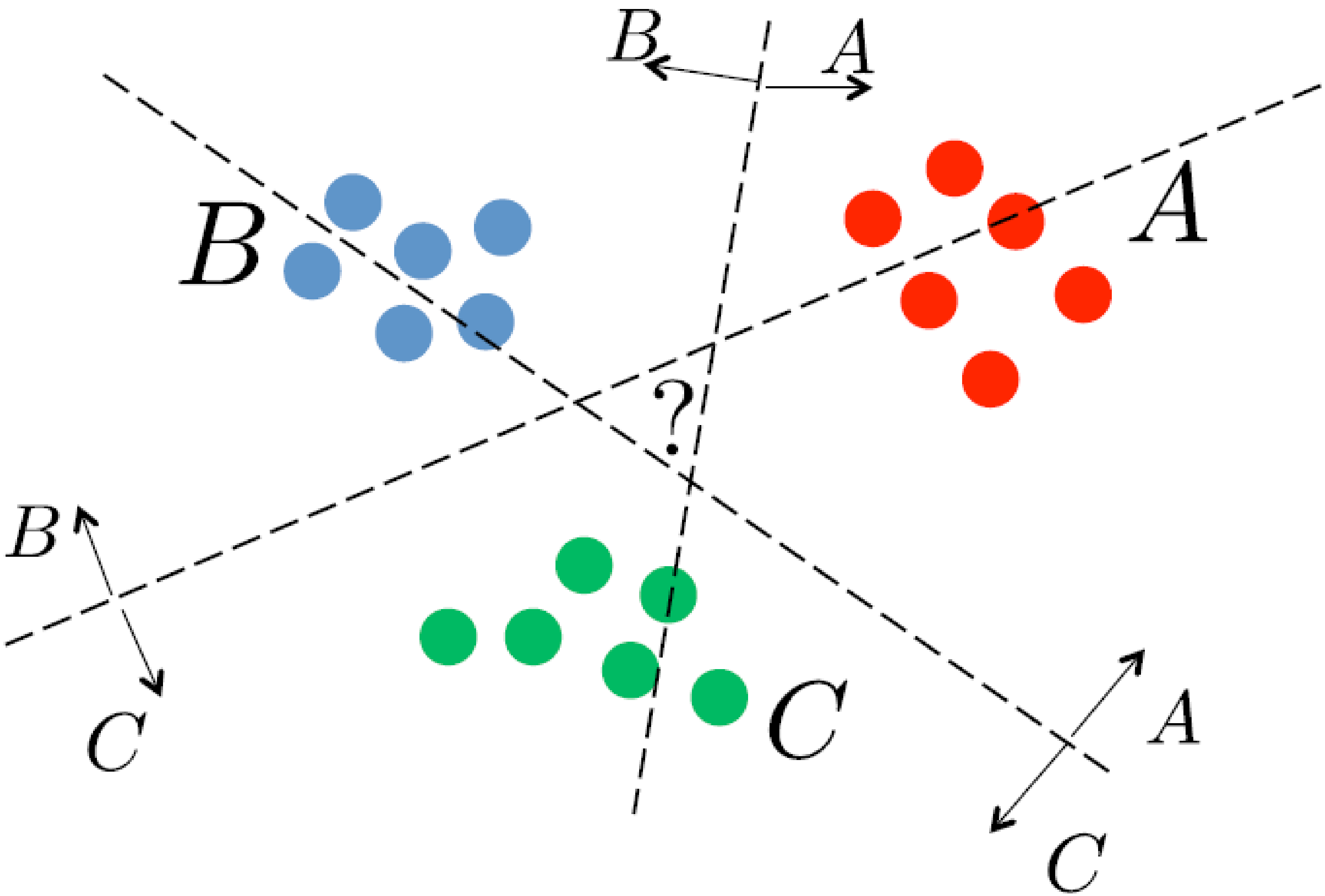
(если всего K классов, то получим **C_K^2 классификаторов**).

Каждый такой классификатор будем обучать только на объектах классов i и j.

- В качестве итогового предсказания выдадим класс, который предсказало наибольшее число алгоритмов:

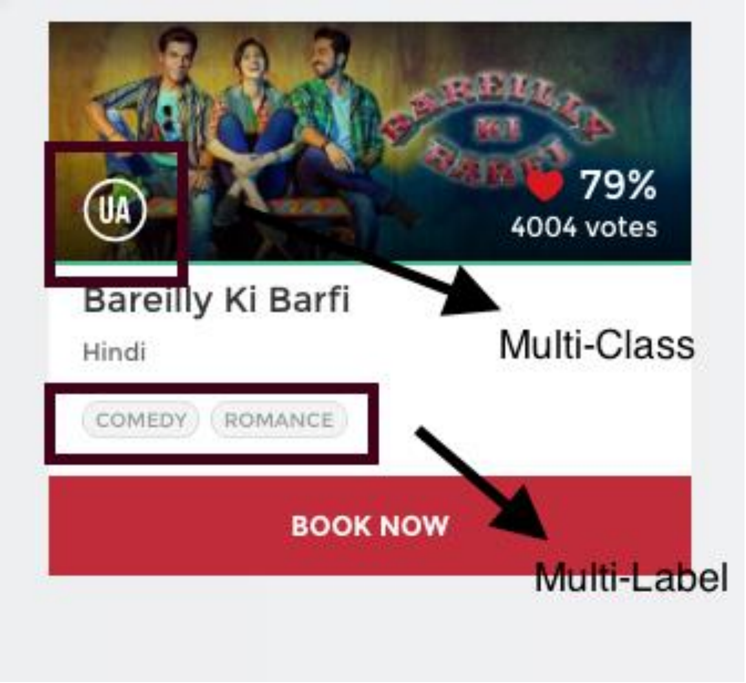
$$a(x) = \underset{k \in \{1,\dots,K\}}{\operatorname{argmax}} \sum_{i=1}^{K} \sum_{j \neq i} [a_{ij}(x) = k]$$

ПОДХОД ALL-VS-ALL



MULTICLASS AND MULTI-LABEL CLASSIFICATION

- Если каждый объект может принадлежать только одному классу, то решаем задачу multiclass классификации
- Если каждый объект может принадлежать нескольким классам (задача классификации с пересекающимися классами), то решаем задачу multi-label классификации.



МЕТРИКИ КАЧЕСТВА

Объяснение на Stepik

Подход 1 (*микроусреднение, micro average*):

- В этом подходе мы вычисляем значения TP, TN, FP, FN по всей матрице ошибок сразу, исходя из их определения. Затем по полученным числам вычисляем выбранные метрики.

В случае микроусреднения все метрики (precision, recall, f1) совпадают с accuracy.

МЕТРИКИ КАЧЕСТВА

Идея: сводим подсчет метрик к бинарному случаю

Подход 2 (*макроусреднение, macro average*):

- Вычислим для каждого двухклассового классификатора $a^k(x) = [a(x) = k]$ метрики TP_k, FP_k, FN_k, TN_k
- Вычислим итоговую метрику для каждого класса в отдельности: $precision_k(a, X) = \frac{TP_k}{TP_k + FP_k}$

Тогда точность в многоклассовом случае:

$$precision(a, X) = \frac{1}{K} \sum_{k=1}^{K} precision_k(a, X)$$

МЕТРИКИ КАЧЕСТВА (ПРИМЕР)

Результаты некоторого классификатора:

		True/Actual		
		Cat (🐱)	Fish (🐠)	Hen (🐔)
Predicted	Cat (🐱)	4	6	3
	Fish (🐠)	1	2	0
	Hen (🐔)	1	2	6

МЕТРИКИ КАЧЕСТВА (ПРИМЕР)

		True/Actual		
		Cat	Fish	Hen
Predicted	Cat	4	6	3
	Fish	1	2	0
	Hen	1	2	6

	precision	recall	f1-score	support
Cat	0.308	0.667	0.421	6
Fish	0.667	0.200	0.308	10
Hen	0.667	0.667	0.667	9
micro avg	0.480	0.480	0.480	25
macro avg	0.547	0.511	0.465	25
weighted avg	0.581	0.480	0.464	25

МНОГОКЛАССОВАЯ ЛОГИСТИЧЕСКАЯ РЕГРЕССИЯ

- Бинарная лог.регрессия предсказывает вероятность класса 1:

$$(w, x) \to a(x) = \frac{1}{1 + \mathrm{e}^{-(w,x)}} = \frac{e^{(w,x)}}{1 + e^{(w,x)}}$$

- Предположим, у нас есть K линейных моделей, каждая из которых дает оценку принадлежности выбранному классу: $b_k(x) = (w_k, x)$.

- Преобразуем вектор предсказаний в вектор вероятностей (softmax-преобразование):

$$\boldsymbol{softmax(b_1, \dots, b_K) = \left(\frac{exp(b_1)}{\sum_{i=1}^{K} exp(b_i)}, \frac{exp(b_2)}{\sum_{i=1}^{K} exp(b_i)}, \dots, \frac{exp(b_K)}{\sum_{i=1}^{K} exp(b_i)}\right)}$$

Тогда вероятность класса k:

$$P(y = k | x, w) = \frac{\exp((w_k, x))}{\sum_{i=1}^{K} \exp((w_i, x))}$$

ОБУЧЕНИЕ ВЕСОВ МОДЕЛИ

$$a_j(x) = P(y = j|x, w) = \frac{\exp\left(b_j(x)\right)}{\sum_{i=1}^{K} \exp(b_i(x))}$$

Обучение – по методу максимального правдоподобия (аналогично бинарной классификации):

$$\Pi = \prod_{i=1}^{n} a_1(x_i)^{[y_i=1]} \cdot a_2(x_i)^{[y_i=2]} \cdot \ldots a_K(x_i)^{[y_i=K]} =$$

$$= \prod_{i=1}^{n} \prod_{j=1}^{K} a_j(x_i)^{[y_i=j]} \to \max_{w_1,\ldots,w_K}$$

$$-\sum_{i=1}^{n} \sum_{j=1}^{K} [y_i = j]\, logP(y = j|x_i, w) \to \min_{w_1,\ldots,w_K}$$

ОБУЧЕНИЕ ВЕСОВ МОДЕЛИ

$$a_j(x) = P(y = j|x, w) = \frac{\exp(b_j(x))}{\sum_{i=1}^{K} \exp(b_i(x))}$$

Обучение – по методу максимального правдоподобия (аналогично бинарной классификации):

$$\Pi = \prod_{i=1}^{n} a_1(x_i)^{[y_i=1]} \cdot a_2(x_i)^{[y_i=2]} \cdot \ldots a_K(x_i)^{[y_i=K]} =$$

$$= \prod_{i=1}^{n} \prod_{j=1}^{K} a_j(x_i)^{[y_i=j]} \to \max_{w_1,\ldots,w_K}$$

То есть в итоге обучаем одну модель (а не K моделей)

$$-\sum_{i=1}^{n} \sum_{j=1}^{K} [y_i = j]\, log P(y = j|x_i, w) \to \min_{w_1,\ldots,w_K}$$