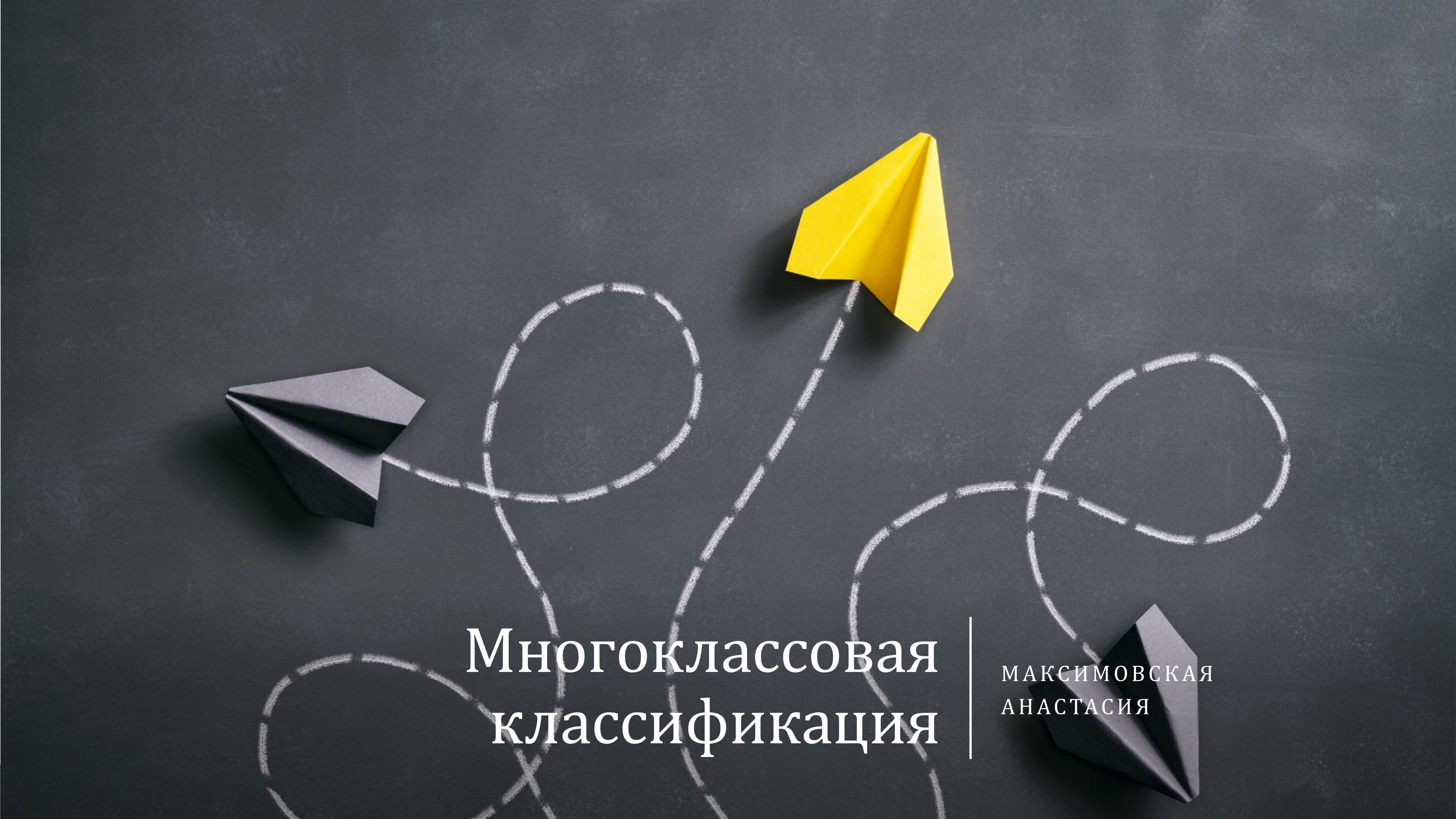


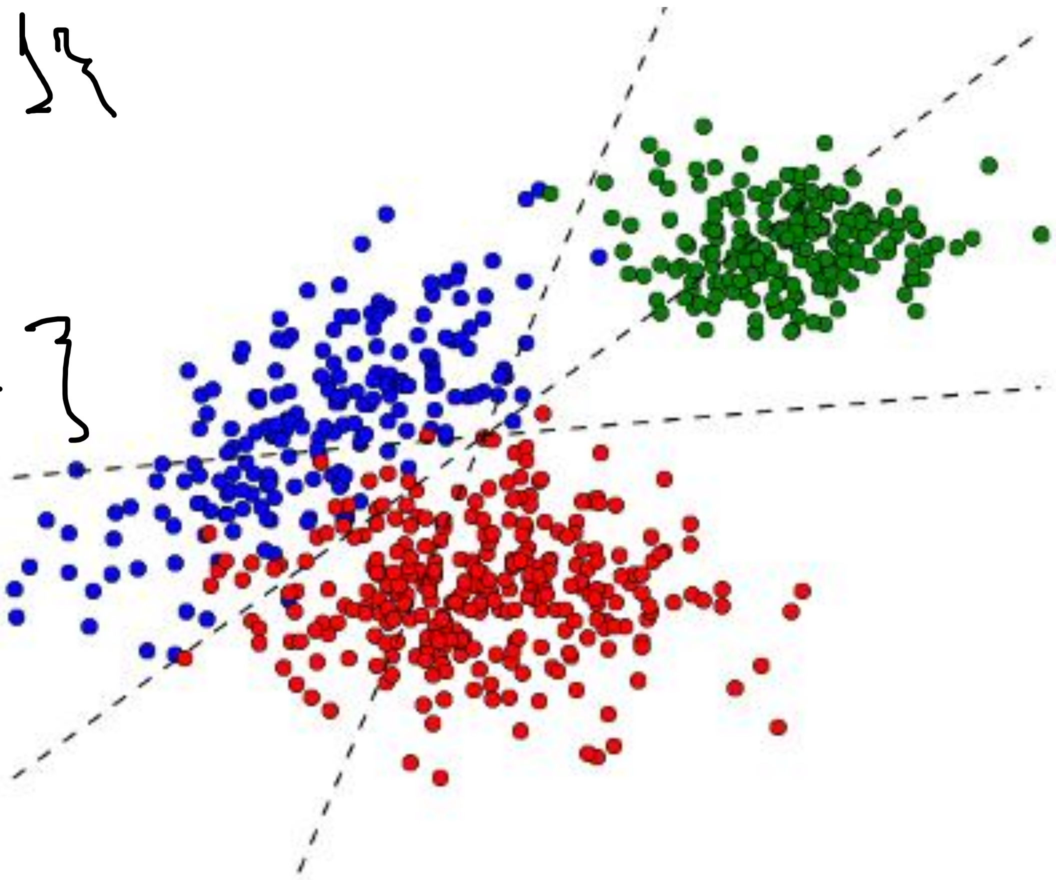
Многоклассовая классификация

МАКСИМОВСКАЯ АНАСТАСИЯ

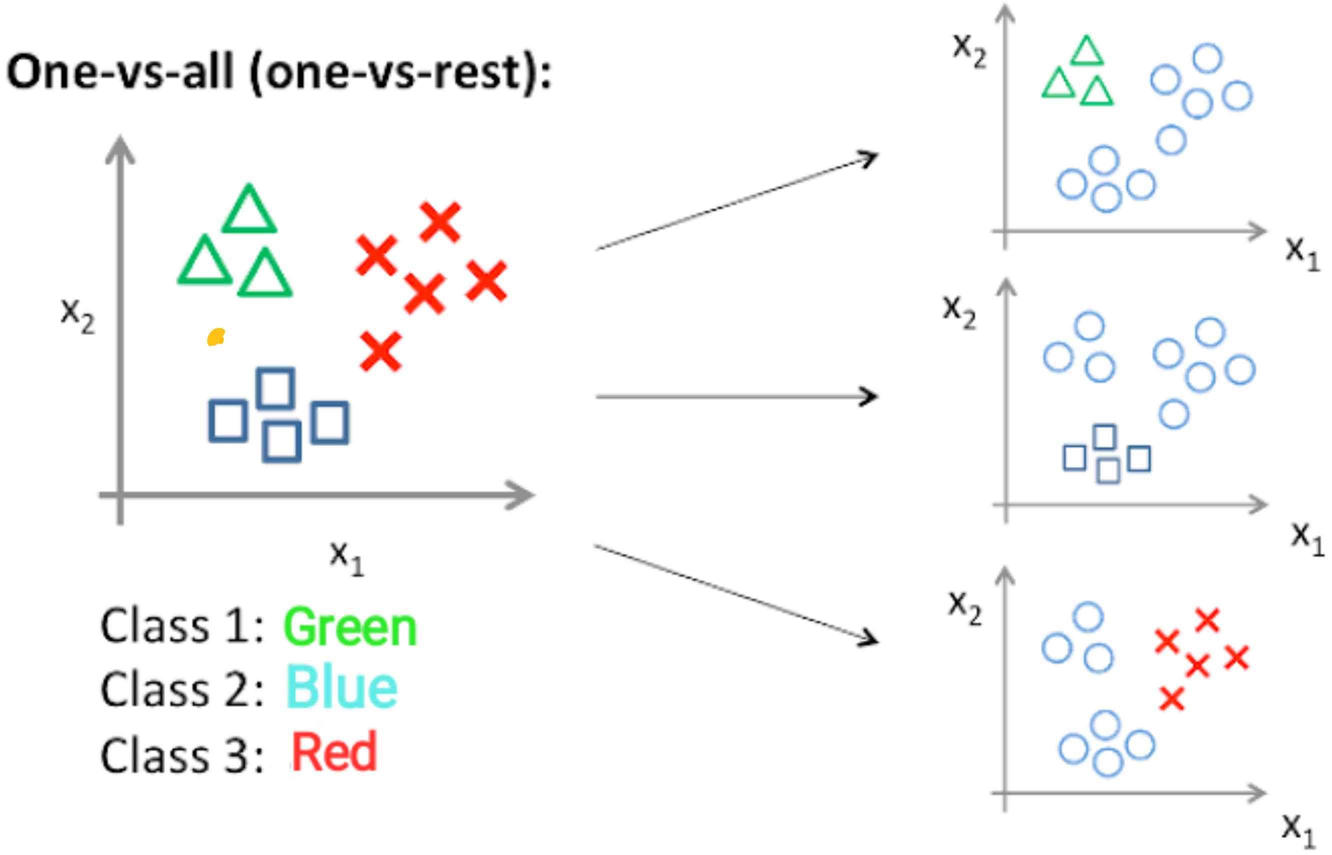
Многоклассовая классификация

$Y = \{+1, -1\}$

$Y = \{1, \ldots, K\}$



Один против всех (one-versus-all)



$b_1(x)$

$b_2(x)$

$b_3(x)$

Один против всех (one-versus-all)

- Обучаем K линейных классификаторов, выдающих оценки принадлежности к классам $1, \dots, K$ соответственно
- Обучаем классификатор b_k по выборке, где целевая переменная равна +1, если это класс k и -1 иначе
- Итоговый классификатор:

$$a(x) = \arg\max_{k \in \{1,\dots,K\}} b_k(x)$$

Один против всех (one-versus-all)

- Каждый классификатор обучается на своей выборке
- Выходы классификаторов будут иметь разные масштабы, сравнивать их некорректно
- Можно нормировать вектора весов, чтобы они выдавали ответы в одной и той же шкале
- Не всегда хорошее решение (например, для SVM)

Все против всех (all-versus-all)

- Обучим C_k^2 классификаторов $a_{ij}(x), i, j = 1, \dots, K, i \neq j$

Напоминание

$k = 3$

$C_3^2 = \frac{3!}{2!\,1!} = 3 = \frac{3\cdot\cancel{2\cdot 1}}{\cancel{2\cdot 1\cdot 1}}$

x_j

$$C_n^k = \frac{n!}{k!(n-k)!}$$

$a_{12}(x) \Rightarrow 1$

$a_{13}(x) \Rightarrow 3$

$a_{23}(x) \Rightarrow 3$

$(x_j; 3)$

Все против всех (all-versus-all)

- Обучим C_k^2 классификаторов $a_{ij}(x), i, j = 1, \dots, K, i \neq j$
- Классификатор $a_{ij}(x)$ будем обучать на подвыборке, содержащей только объекты классов i и j:

$$X_{ij} = \{(x_n, y_n) \in X \mid [y_n = i] = 1 \text{ или } [y_n = j] = 1\}$$

- Классификатор будет для любого объекта выдавать класс i или j

Все против всех (all-versus-all)

- Чтобы классифицировать новый объект, подадим его на вход каждого из построенных бинарных классификаторов
- Каждый из них проголосует за свой класс, в качестве ответа выберем тот класс, за который наберется больше всего голосов:

$$a(x) = \underset{k \in \{1,\ldots,K\}}{\arg\max} \sum_{i=1}^{K} \sum_{j \neq i} [a_{ij}(x) = k].$$

Многоклассовая логистическая регрессия

$$b_k(x) = \langle w, x \rangle + w_0$$

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

$$(b_1(x), \dots, b_k(x))$$

Многоклассовая логистическая регрессия

- Как преобразовать вектор оценок нескольких классификаторов в вероятности?

$$\mathrm{SoftMax}(z_1, \dots, z_K) = \left(\frac{\exp(z_1)}{\sum_{k=1}^{K} \exp(z_k)}, \dots, \frac{\exp(z_K)}{\sum_{k=1}^{K} \exp(z_k)} \right)$$

- Вероятность класса *k* в таком случае:

$$P(y = k \mid x, w) = \frac{\exp\left(\langle w_k, x \rangle + w_{0k}\right)}{\sum_{j=1}^{K} \exp\left(\langle w_j, x \rangle + w_{0j}\right)}$$

Многоклассовая логистическая регрессия

- Обучаем веса с помощью метода максимального правдоподобия:

$$\sum_{i=1}^{\ell} \log P(y = y_i \mid x_i, w) \to \max_{w_1, \dots, w_K}$$

Метрики

- Есть 2 подхода: микро- и макро-усреднение метрик классификации
- Микро-усреднение: усредняем по всем классам, считаем итоговую

$$\text{precision}(a, X) = \frac{\overline{\text{TP}}}{\overline{\text{TP}} + \overline{\text{FP}}}, \qquad \overline{\text{TP}} = \frac{1}{K}\sum_{k=1}^{K} \text{TP}_k.$$

- Макро-усреднение: вычисляем итоговую метрику для каждого класса, усредняем по всем классам

$$\text{precision}(a, X) = \frac{1}{K}\sum_{k=1}^{K} \text{precision}_k(a, X); \qquad \text{precision}_k(a, X) = \frac{\text{TP}_k}{\text{TP}_k + \text{FP}_k}.$$

Метрики: пример

	Class	Predicted Class	Correct?
1	orange	lemon	0
	orange	lemon	0
	orange	apple	0
	orange	orange	1
	orange	apple	0
2	lemon	lemon	1
	lemon	apple	0
3	apple	apple	1
	apple	apple	1

$1 = 1/5 = 0.2$

$2 = 1/2 = 0.5$

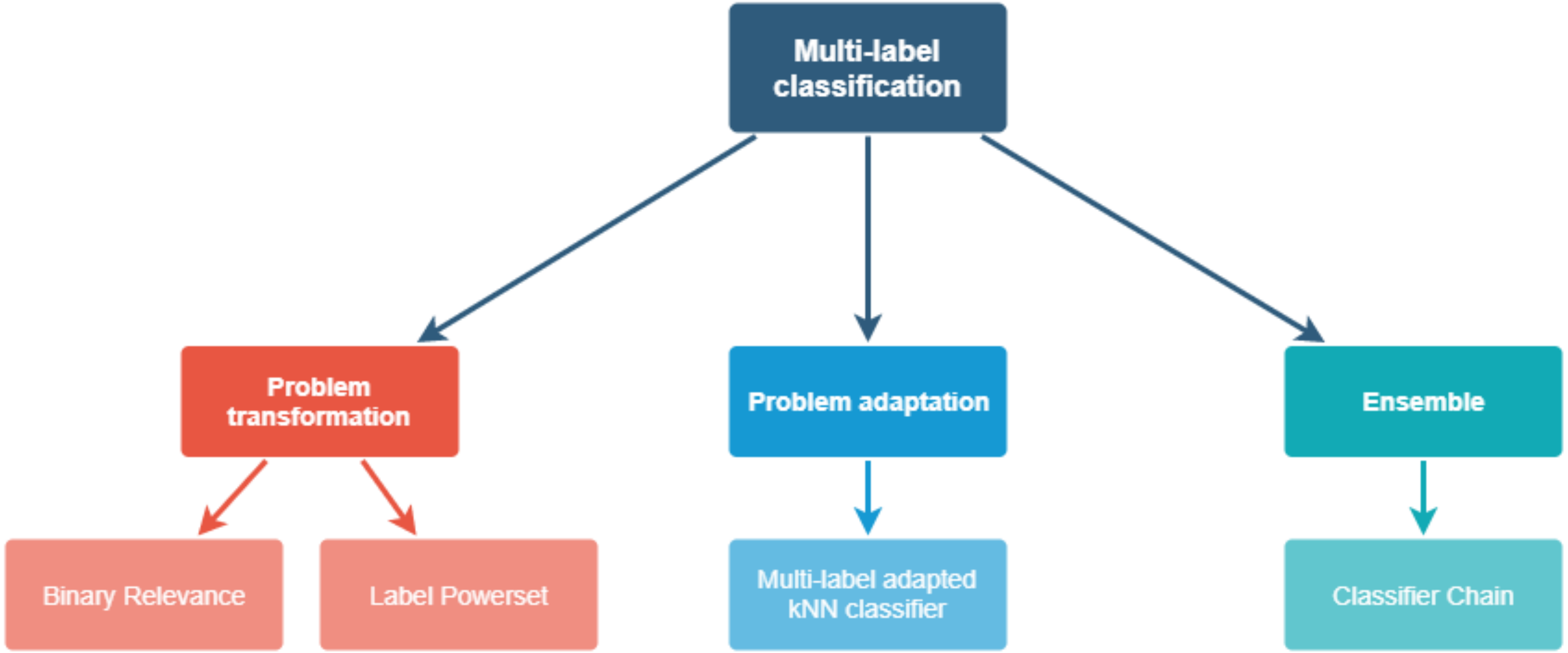
$3 = 2/2 = 1$

Macro: $\frac{1.7}{3} = 0.57$

Micro: $\frac{4}{9} = 0.4$

Пересекающиеся классы

Пересекающиеся классы



Binary relevance

- Предполагаем, что все классы независимы, и определяем принадлежность к каждому отдельным классификатором
- Проблема – не учитываем возможные связи между классами

$x_j \to h_1(x)$ 0,3 −

$\to h_2(x)$ 0,6 +

$\to h_3(x)$ 0,67 +

$x_j \to 2, 3$

Стекинг классификаторов

x_j $b_1(x)$ 0.7
$b_2(x)$ 0.3
$b_3(x)$ 0.6

$x_j = (0.7, 0.3, 0.6)$

- Разделим выборку X на 2 части: X_1 и X_2
- На первой X_1 обучим K независимых классификаторов $b_k(x)$
- Для каждого объекта из второй выборки сформируем признаковое описание из прогнозов наших классификаторов:

$$x'_{ik} = b_k(x_i), \quad x_i \in X_2$$

- Обучим на ***полученной*** выборке новый набор классификаторов $a_k(x)$,каждый из которых определяет принадлежность объекта к одному из классов

Стекинг классификаторов

- Таким образом, новые классификаторы $a_k(x)$ опираются на прогнозы классификаторов с первого этапа $b_k(x)$
- Благодаря этому они могут обнаружить связи между классами
- Обучать $b_k(x)$ и $a_k(x)$ на одной выборке – плохая идея, т.к. может привести к переобучению

Трансформация пространства ответов

- Хотим учитывать связи между классами в рамках одной модели
- Идея: трансформировать пространство ответов так, чтобы классы стали менее зависимыми
- Используем SVD разложение для вектора ответов Y

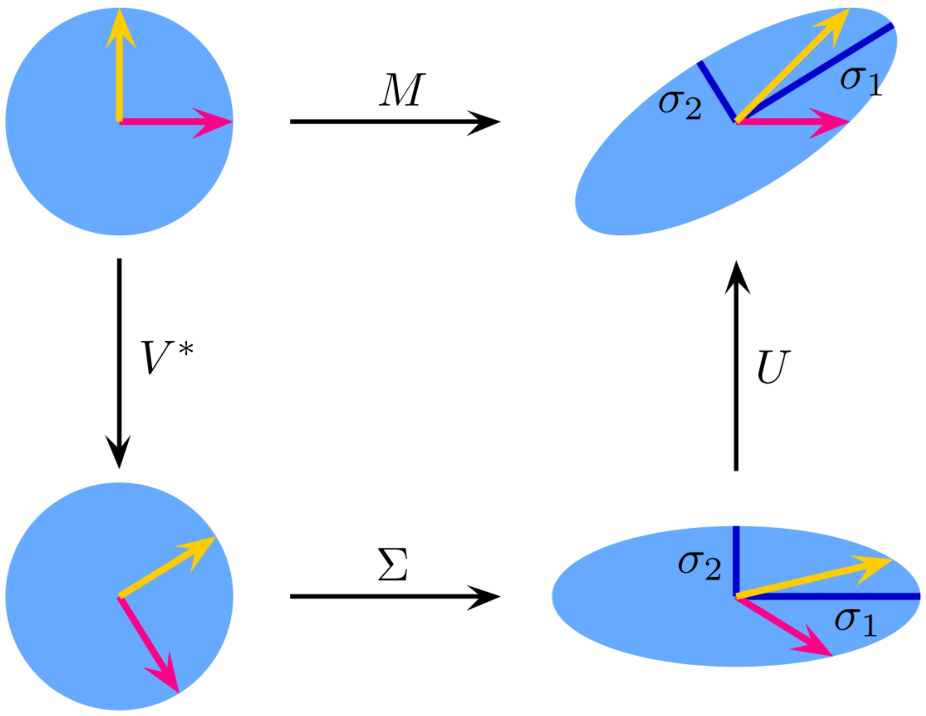
$x_1 \; \{1,2,3,4\} \to y_1$

$\{1,2,5\} \to z_1$

$Y|Z = \underbrace{\{3,4\}}_{2}$

$Z|Y = \underbrace{\{5\}}_{1}$

SVD



$$M = U \cdot \Sigma \cdot V^*$$

Трансформация пространства ответов

- Обозначим через V_M матрицу, состоящую из тех M столбцов матрицы V , которые соответствуют наибольшим сингулярным числам
- Спроецируем с её помощью матрицы Y :

$$YV_M = Y' \in \mathbb{R}^{\ell \times M}$$

- Настроим на новые метки Y' независимые модели
- Получим матрицу прогнозов A' и переведем ее в исходное пространство:

$$A = A'V_M^T$$

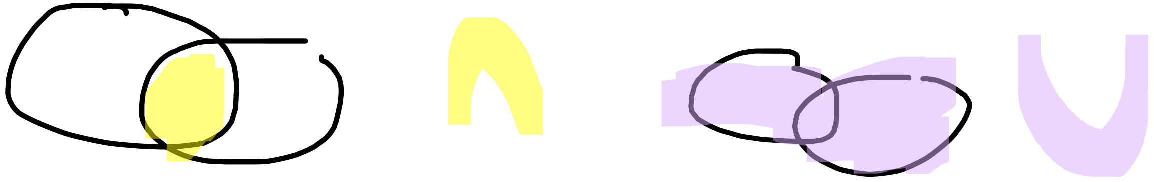
Метрики качества для multi-label

- Z_i – множество классов, к котором можно отнести объект
- Хемминговo расстояние доля классов, факт принадлежности которым угадан неверно:

$$\text{hamming}(a, X) = \frac{1}{\ell} \sum_{i=1}^{\ell} \boxed{\frac{\overbrace{|Y_i \setminus Z_i|}^{2} + \overbrace{|Z_i \setminus Y_i|}^{1}}{K}}$$

$3/5 = 0.6$

- Хотим минимизировать



Метрики качества для multi-label

$Y = \{1, 2, 3, 4\}$ $\quad Z = \{1, 2, 5\}$

$$\text{accuracy}(a, X) = \frac{1}{\ell} \sum_{i=1}^{\ell} \frac{|Y_i \cap Z_i|}{|Y_i \cup Z_i|}, \quad = \frac{|\{1,2\}|}{|\{1, \ldots, 5\}|} = \frac{2}{5} = 0$$

$$\text{precision}(a, X) = \frac{1}{\ell} \sum_{i=1}^{\ell} \frac{|Y_i \cap Z_i|}{|Z_i|}, \quad = \frac{2}{3} = 0{,}67$$

$$\text{recall}(a, X) = \frac{1}{\ell} \sum_{i=1}^{\ell} \frac{|Y_i \cap Z_i|}{|Y_i|}. \quad = \frac{2}{4} = 0{,}5$$

Категориальные признаки

- Помним: One-Hot кодирование

$x \quad f(x) \qquad U = \{u_1, u_2, \dots, u_n\}$

animal	cat	dog	~~rabbit~~
cat	1	0	~~0~~
dog	0	1	~~0~~
Rabbit	0	0	~~1~~

Категориальные признаки

- Помним: One-Hot кодирование
- Бинарное кодирование с хэшированием

Бинарное кодирование с хэшированием

- Выберем хэш-функцию $h : U \to \{1, 2, \ldots, B\}$
- После этого бинарные признаки можно индексировать значениями хэш-функции:

$$g_j(x) = [h(f(x)) = j], \quad j = 1, \ldots, B.$$

https://contrib.scikit-learn.org/category_encoders/hashing.html – Hashing Encoder,
https://docs.python.org/3/library/hashlib.html – hashlib

Бинарное кодирование с хэшированием

Преимущества:

- Отпадает необходимость в хранении соответствий между значениями категориального признака и индексами бинарных признаков
- Позволяет понизить количество признаков (как правило, без существенной потери качества)

Категориальные признаки

- Помним: One-Hot кодирование (и другие кодирования – см. лекции 1-2)
- Бинарное кодирование с хэшированием
- Счетчики

Счетчики

cat	y
1	1
2	0
3	1
1	0
2	1

$U \in \{1, 2, 3\}$

$c_1 = 2 \quad c_2 = 2 \quad c_3 = 1$

$s_{10} = 1 \quad s_{11} = 1$

$s_{20} = 1 \quad s_{21} = 1$

$s_{30} = 0 \quad s_{31} = 1$

где
b от
U

K+1:

$$\text{counts}(u, X) = \sum_{(x,y)\in X} [f(x) = u],$$ – сколько раз u_i

$$\text{successes}_{k}(u, X) = \sum_{(x,y)\in X} [f(x) = u][y = k], \quad k = 1, \ldots, K.$$

– сколько раз быр. u_i and k класс k

c1 c2 c3
1 3 13
2 3 23
1 3 13
2 4 24

Счетчики

Заменим наш категориальный признак на K вещественных $g_1(x), \dots, g_k(x)$

$$g_k(x, X) = \frac{\text{successes}_k(f(x), X) + c_k}{\text{counts}(f(x), X) + \sum_{m=1}^{K} c_m}, \quad k = 1, \dots, K.$$

, 2

$$\downarrow$$

$$p(y = k \mid f(x))$$