



Лекция 6

Многоклассовая классификация. Ядерные методы. Отбор признаков.

Марк Блуменау

На основе материалов Кантонистовой Е.О.

ВШЭ, 2025

МНОГОКЛАССОВАЯ КЛАССИФИКАЦИЯ

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

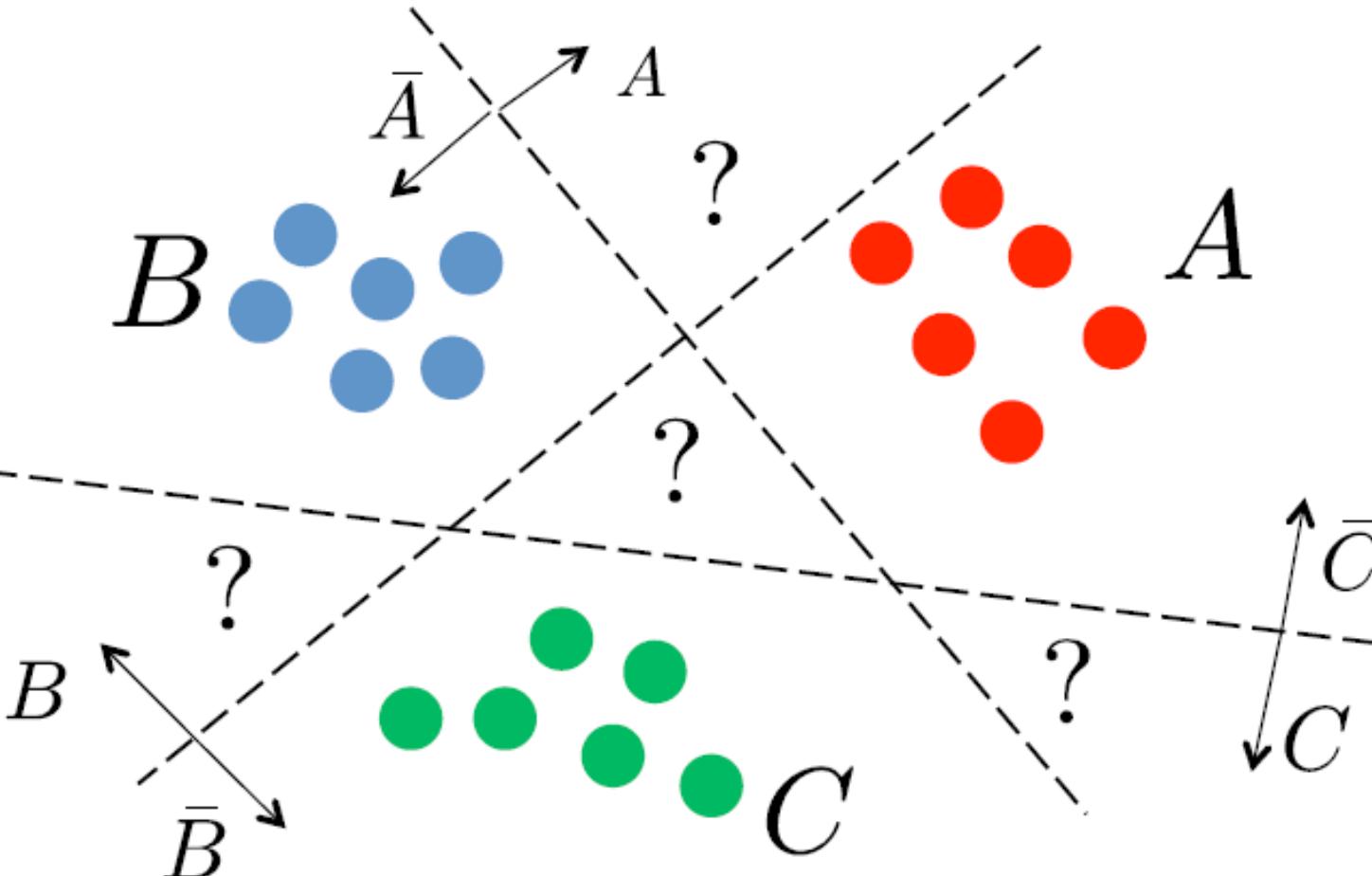
$$b_k(x) = \text{sign}((w_k, x) + w_{0k})$$

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

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

- 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

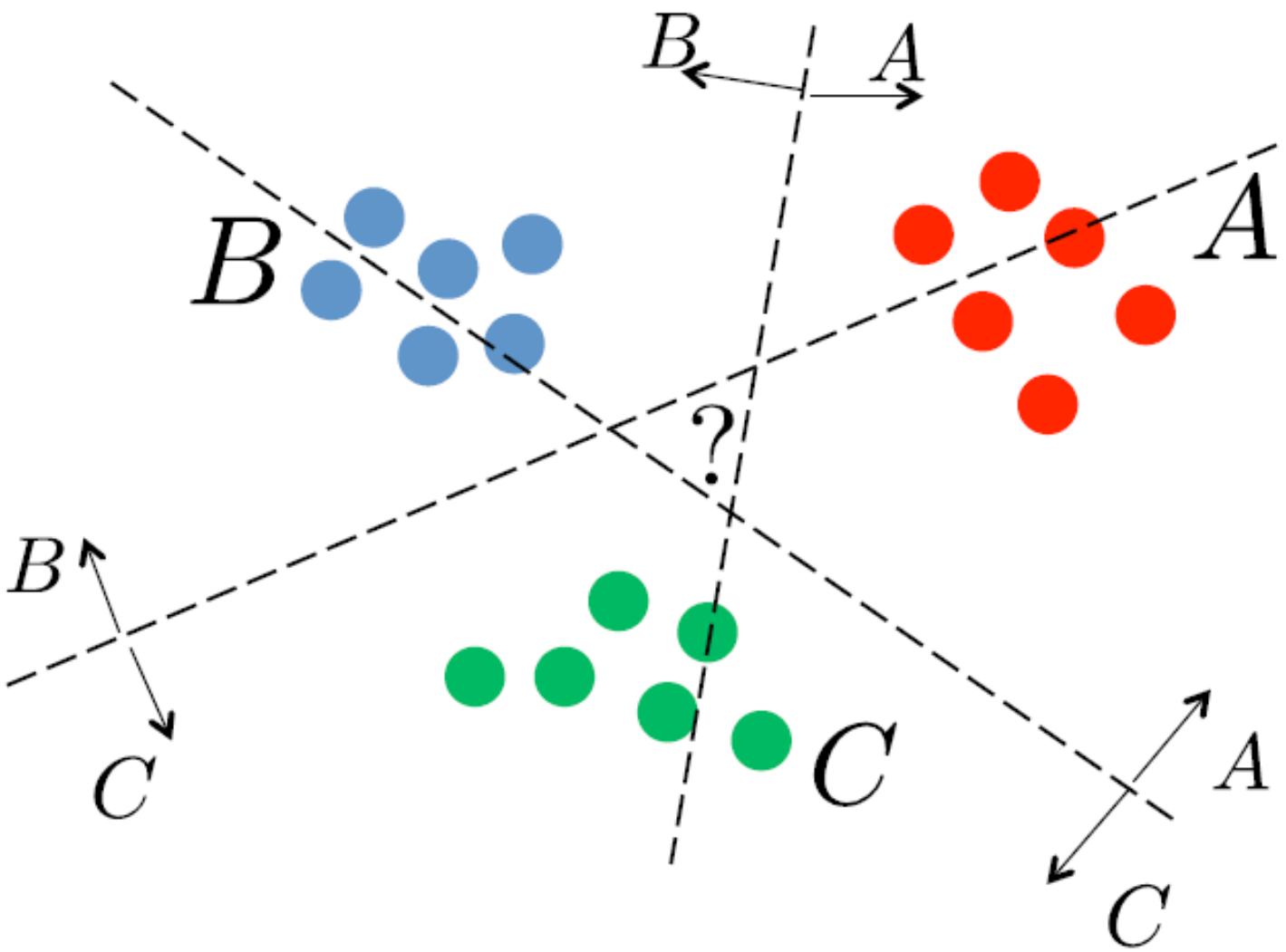
(если всего K классов, то получим C_K^2 классификаторов).

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

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

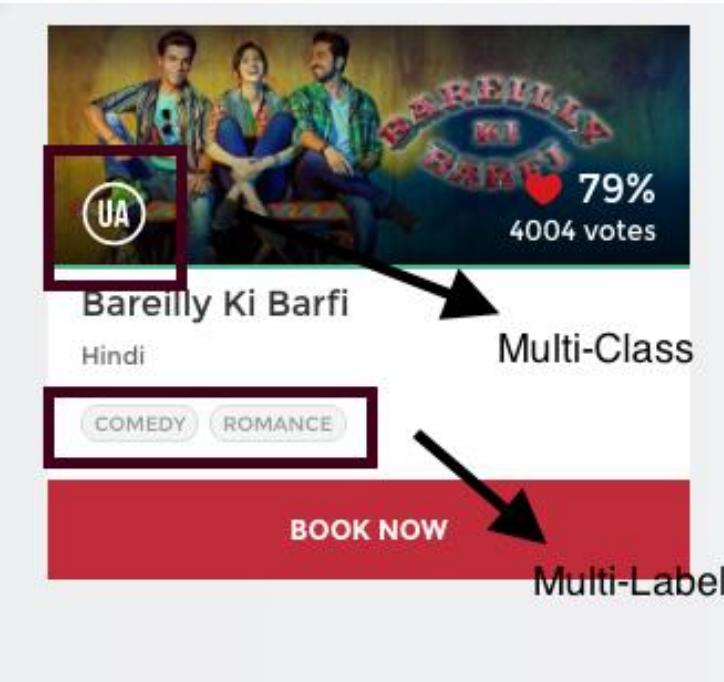
$$a(x) = \operatorname{argmax}_{k \in \{1, \dots, K\}} \sum_{i=1}^K \sum_{j \neq i} [a_{ij}(x) = k]$$

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

Идея: сводим подсчет метрик к бинарному случаю

Подход 1 ([микроусреднение, micro average](#)):

- Вычислим для каждого двухклассового классификатора $a^k(x) = [a(x) = k]$ метрики TP_k, FP_k, FN_k, TN_k
- Усредним каждую характеристику по всем классам, например, $TP = \frac{1}{K} \sum_{k=1}^K TP_k$.

Тогда точность в многоклассовом случае:

$$precision(a, X) = \frac{TP}{TP + FP}$$

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 (img alt="Cat icon" data-bbox="445 375 495 435")/	Fish (img alt="Fish icon" data-bbox="645 375 695 435"/>)	Hen (img alt="Hen icon" data-bbox="845 375 895 435")/)
Predicted	Cat (img alt="Cat icon" data-bbox="115 475 245 535")/	4	6	3
	Fish (img alt="Fish icon" data-bbox="115 575 245 635")/	1	2	0
	Hen (img alt="Hen icon" data-bbox="115 695 245 755")/	1	2	6

МЕТРИКИ КАЧЕСТВА (ПРИМЕР)

		True/Actual		
		Cat (img alt="Cat icon" data-bbox="435 215 485 275})	Fish (img alt="Fish icon" data-bbox="615 215 665 275})	Hen (img alt="Hen icon" data-bbox="805 215 855 275})
Predicted	Cat (img alt="Cat icon" data-bbox="135 310 235 370")	4	6	3
	Fish (img alt="Fish icon" data-bbox="135 420 235 480")	1	2	0
	Hen (img alt="Hen icon" data-bbox="135 530 235 590")	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:

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

- 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):

$$\text{softmax}(\mathbf{b}_1, \dots, \mathbf{b}_K) = \left(\frac{\exp(b_1)}{\sum_{i=1}^K \exp(b_i)}, \frac{\exp(b_2)}{\sum_{i=1}^K \exp(b_i)}, \dots, \frac{\exp(b_K)}{\sum_{i=1}^K \exp(b_i)} \right)$$

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

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

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

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

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

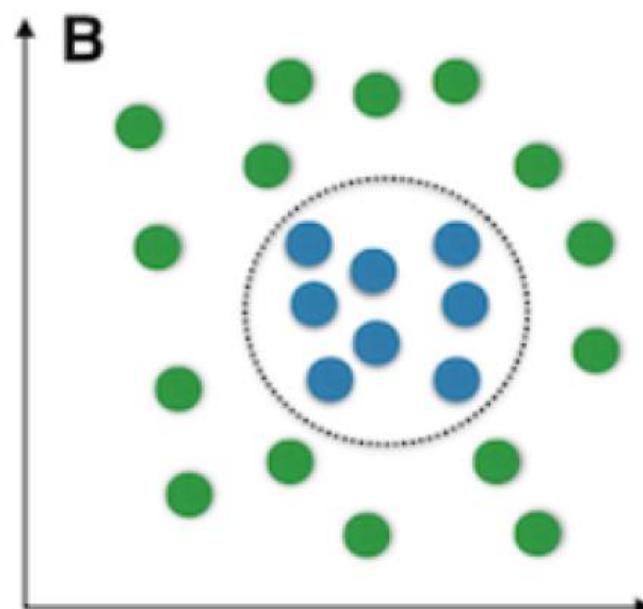
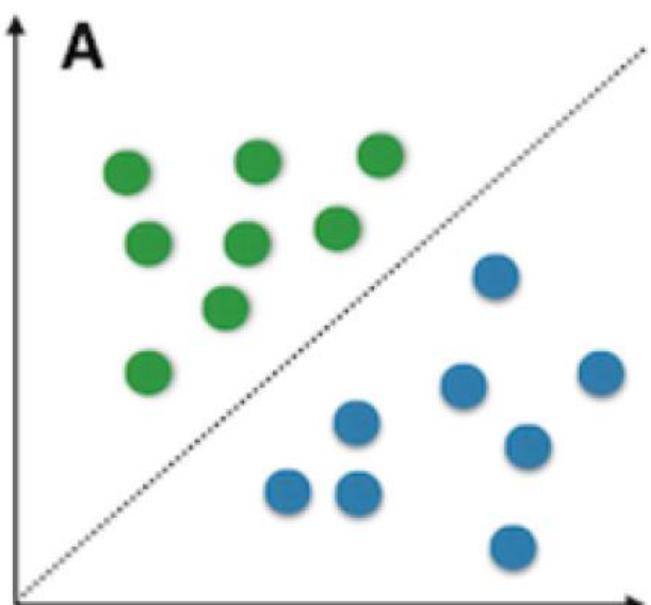
То есть в итоге обучаем одну модель (а не K моделей)

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

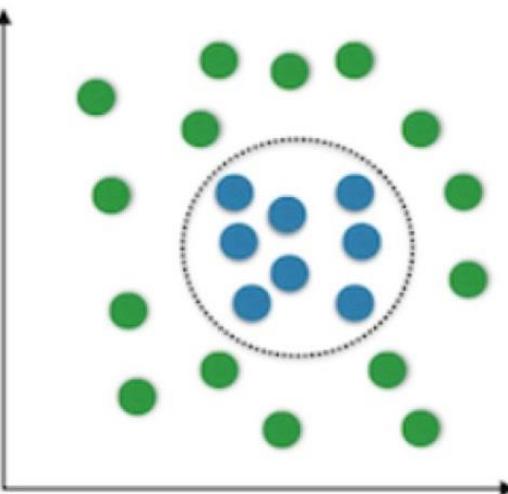
ЯДРОВЫЕ МЕТОДЫ

НЕЛИНЕЙНЫЕ ЗАДАЧИ КЛАССИФИКАЦИИ

Linear vs. nonlinear problems



НЕЛИНЕЙНЫЕ ЗАДАЧИ КЛАССИФИКАЦИИ

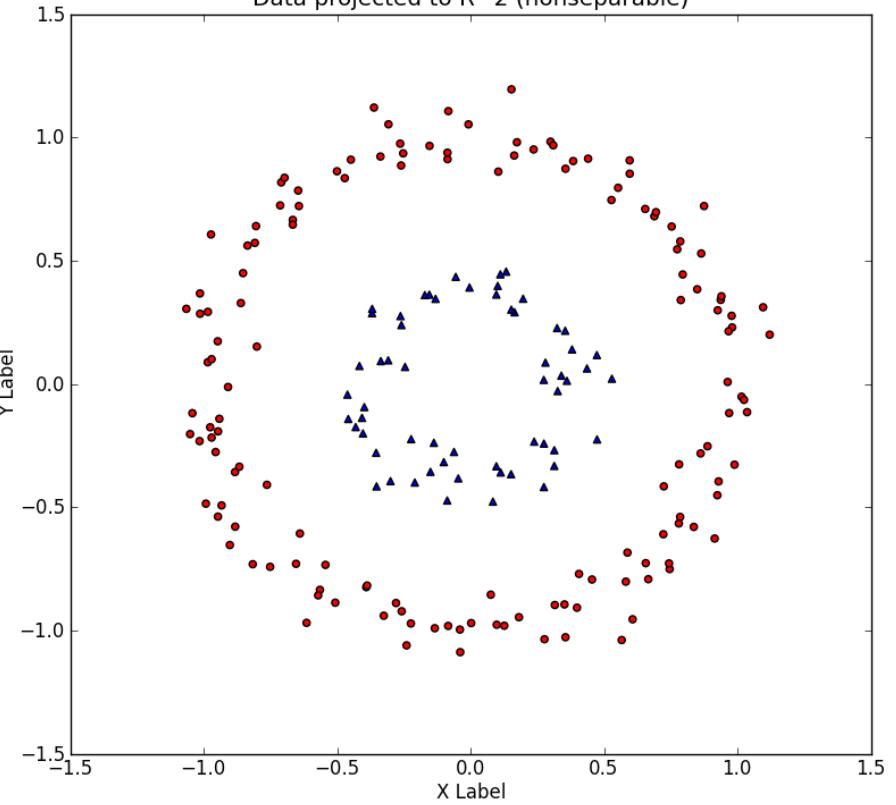


$$(x_1, x_2) \rightarrow (x_1, x_2, x_3 = x_1^2 + x_2^2)$$

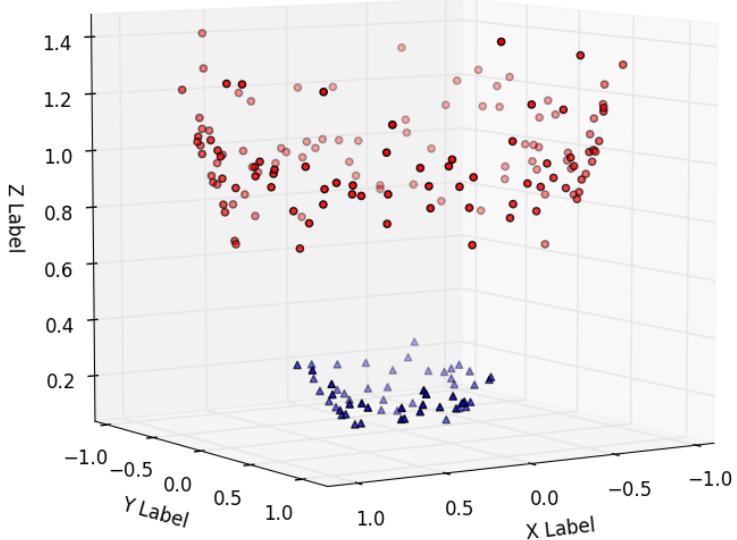
- В новом пространстве признаков выборка идеально разделяется гиперплоскостью.
- ***В новом признаковом пространстве*** классификатор имеет вид $a(x) = sign(w, x) = sign(w_0 + w_1x_1 + w_2x_2 + w_3x_3)$, то есть ***разделяющая поверхность $w_0 + w_1x_1 + w_2x_2 + w_3x_3 = 0$ – линейная.***

ПЕРЕХОД К НОВЫМ ПРИЗНАКАМ

Data projected to R^2 (nonseparable)



Data in R^3 (separable)



ПЕРЕХОД К НОВЫМ ПРИЗНАКАМ

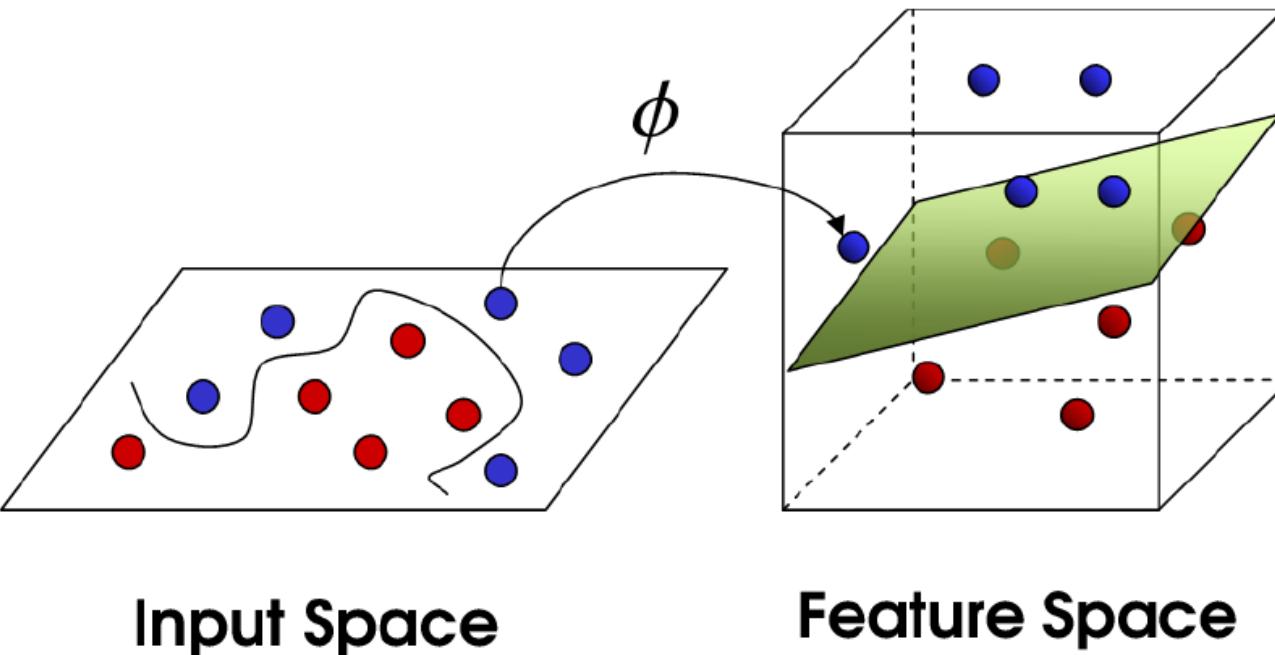
Переход к новым признакам

$$x = (x_1, x_2, \dots, x_d) \rightarrow \varphi_1(x), \varphi_2(x), \dots, \varphi_m(x)$$

Позволяет построить классификатор:

- являющийся *нелинейным в исходном пространстве признаков x_1, x_2, \dots, x_d* (и строящий нелинейную разделяющую поверхность)
- являющийся *линейным в новом пространстве признаков $\varphi_1(x), \varphi_2(x), \dots, \varphi_m(x)$* – спрямляющем пространстве

ПЕРЕХОД К НОВЫМ ПРИЗНАКАМ



ПЕРЕХОД К НОВЫМ ПРИЗНАКАМ

Линейный классификатор:

$$a(x) = \text{sign}((w, \mathbf{x}) + w_0)$$

Классификатор в новом признаковом пространстве

$$a(x) = \text{sign}((w, \varphi(\mathbf{x})) + w_0)$$

ПРОБЛЕМА

При переходе к новым признакам

$$x = (x_1, \dots, x_n) \rightarrow \varphi_1(x), \varphi_2(x), \dots \varphi_N(x)$$

новых признаков может потребоваться довольно много, чтобы линейно разделить выборку в новом пространстве.

Поэтому **сильно возрастает вычислительная сложность алгоритма, а также объем памяти, нужный для хранения всех данных.**

ЯДРОВОЙ ТРЮК (KERNEL TRICK)

При переходе к новым признакам

$$x = (x_1, \dots, x_n) \rightarrow \varphi_1(x), \varphi_2(x), \dots \varphi_N(x)$$

новых признаков может потребоваться довольно много, чтобы линейно разделить выборку в новом пространстве.

Поэтому **сильно возрастает вычислительная сложность алгоритма, а также объем памяти, нужный для хранения всех данных.**

- Существует подход к решению этой проблемы под названием ***kernel trick (ядровой трюк)***. Ядровой трюк позволяет перейти в спрямляющее пространство без увеличения вычислительной сложности и требуемой памяти.

ЯДРО

Ядро – это функция $K(x, z)$, представимая в виде скалярного произведения $K(x, z) = (\varphi(x), \varphi(z))$, где $\varphi: X \rightarrow H$ – отображение из исходного признакового пространства X в некоторое спрямляющее пространство H .

Теорема (Мерсер). Функция $K(x, z)$ является ядром тогда и только тогда, когда:

1) $K(x, z) = K(z, x)$

2) Для любой конечной выборки (x_1, \dots, x_l) матрица $K =$

$$\left(K(x_i, x_j) \right)_{i,j=1}^l \text{ неотрицательно определена}$$

Из теоремы Мерсера следует, что ядро $K(x, z)$ задаёт скалярное произведение объектов x и z .

KERNEL TRICK

- Идея ядрового трюка состоит в том, что некоторые модели машинного обучения (в частности, линейную регрессию и SVM) можно записать в таком виде, чтобы и модель, и функционал ошибки зависели **только от скалярных произведений объектов (а не от самих объектов)**.

Пример – SVM: *переходим к новым признакам $\varphi(x)$*

- Исходная модель: $a(x, w) = \text{sign}((w, \varphi(x)) + w_0)$
- Модель можно записать в виде (двойственная запись):

$$a(x, \lambda) = \text{sign}\left(\sum_{i=1}^l \lambda_i y_i K(x_i, x) - w_0\right)$$

- То есть для вычисления предсказания не нужно вычислять значения новых признаков $\varphi(x)$, а достаточно уметь вычислять ядро $K(x_i, x)$.
- Таким образом, мы можем изначально задать только ядро и не задавать (даже не знать!) явный вид преобразования $\varphi(x)$ и тем самым решить проблему размерности.

МЕТОДЫ ПОСТРОЕНИЯ ЯДЕР

Теорема 1. Пусть $K_1(x, z)$ и $K_2(x, z)$ - ядра, заданные на множестве X . Тогда следующие функции являются ядрами:

$$1) K(x, z) = K_1(x, z) + K_2(x, z)$$

$$2) K(x, z) = \alpha K_1(x, z), \alpha > 0$$

$$3) K(x, z) = K_1(x, z)K_2(x, z)$$

$$4) K(x, z) = f(x)f(z), f(x) - вещественная функция на X$$

$$5) K(x, z) = K_3(\varphi(x), \varphi(z)), \varphi: X \rightarrow \mathbb{R}^n -$$

векторная функция на X, K_3 – ядро, заданное на \mathbb{R}^n .

МЕТОДЫ ПОСТРОЕНИЯ ЯДЕР

Теорема 2. Пусть $K_1(x, z), K_2(x, z), \dots$ - последовательность ядер, причем предел

$$K(x, z) = \lim_{n \rightarrow \infty} K_n(x, z)$$

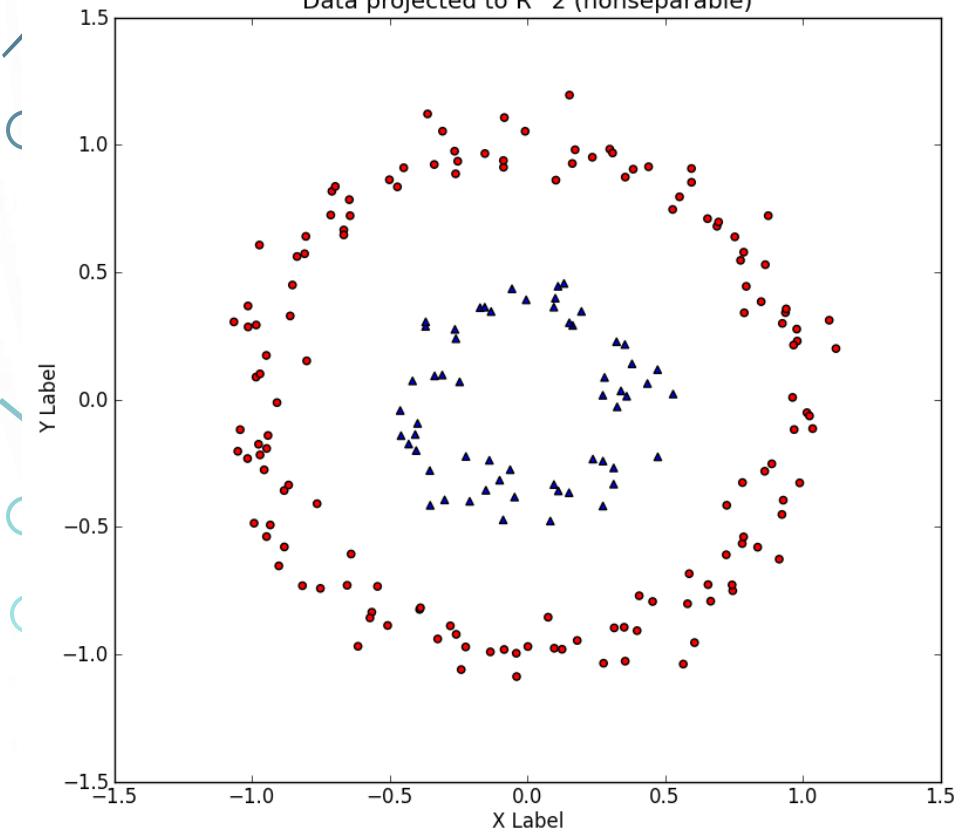
Существует для всех x и z . Тогда $K(x, z)$ ядро.

ПРИМЕРЫ ЯДЕР

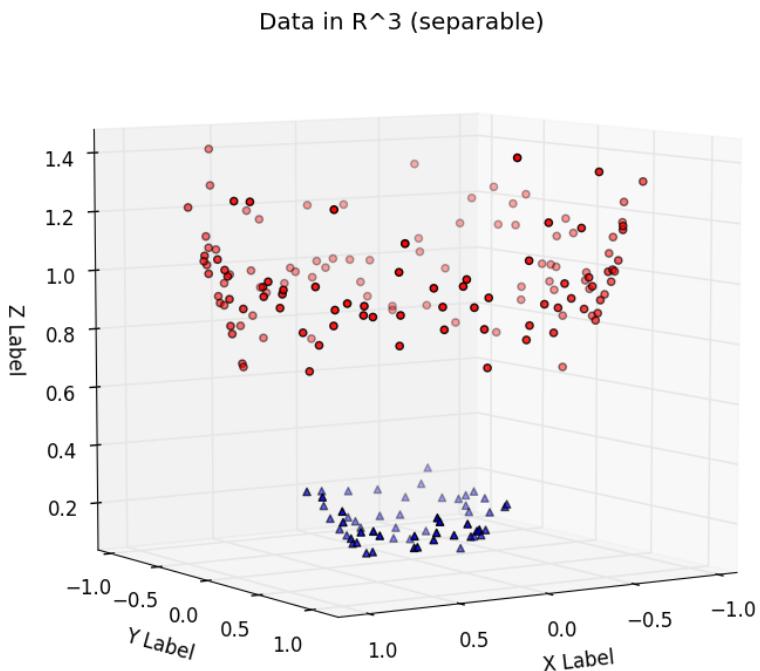
- $K(x, z) = 1$
- $K(x, z) = (x, z)$ – скалярное произведение
- $K(x, z) = (x, z)^2$, где $x = (x_1, x_2), z = (z_1, z_2)$
- $K(x, z) = \exp(-\gamma|x - y|^2)$ – гауссовское или радиальное ядро (RBF-ядро).
- $K(x, z) = p((x, z))$, где p – многочлен с положительными коэффициентами
- $K(x, z) = ((x, z) + R)^d, R > 0$ – полиномиальное ядро

РАДИАЛЬНОЕ ЯДРО

Data projected to R^2 (nonseparable)

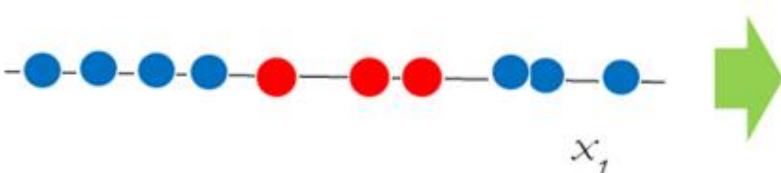


Data in R^3 (separable)

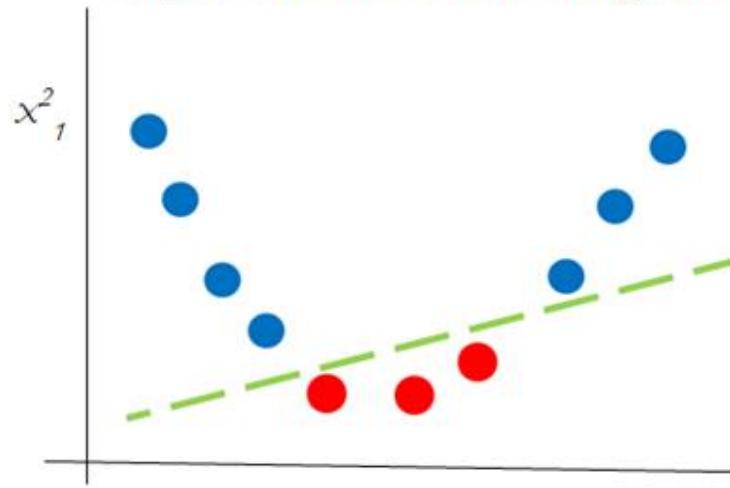


ПОЛИНОМИАЛЬНОЕ ЯДРО

*1-Dimensional Linearly
Inseparable Classes*

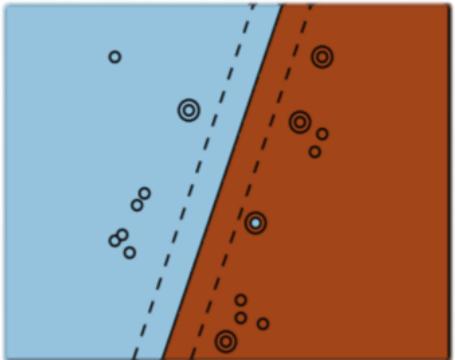


*1-Dimensional Linearly
Inseparable Classes transformed with
Polynomial Kernel of Degree 2*



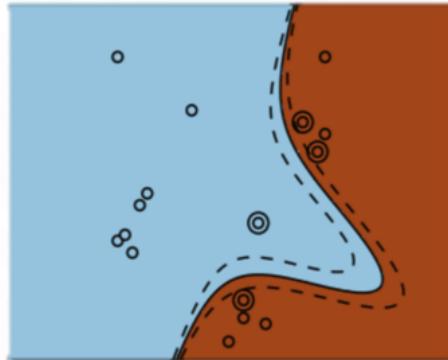
ПРИМЕР: SVM С РАЗЛИЧНЫМИ ЯДРАМИ (ДВА КЛАССА)

Linear Kernel



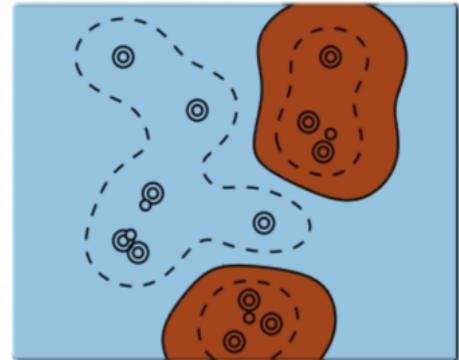
C hyperparameter

Polynomial Kernel



C plus gamma, degree and coefficient hyperparameters

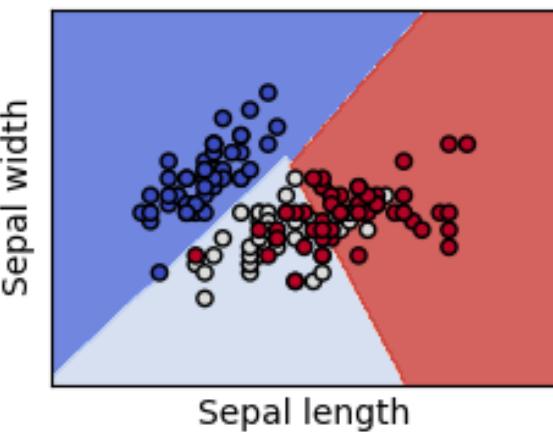
RBF Kernel



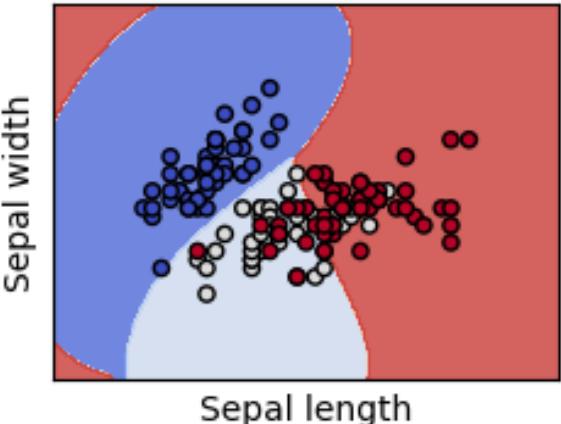
C plus gamma hyperparameter

ПРИМЕР: SVM С РАЗЛИЧНЫМИ ЯДРАМИ (ТРИ КЛАССА)

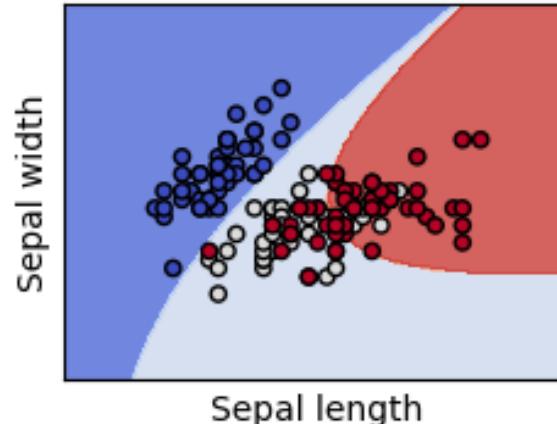
SVC with linear kernel



SVC with RBF kernel



SVC with polynomial (degree 3) kernel



ОТБОР ПРИЗНАКОВ

VARIANCE THRESHOLD

- Можем удалить признаки, которые имеют очень маленькую дисперсию, т.е. практически константы.

ОТБОР ПРИЗНАКОВ ПО КОРРЕЛЯЦИИ С ЦЕЛЕВОЙ ПЕРЕМЕННОЙ

- Для каждого признака вычислим его корреляцию с целевой переменной. Будем выкидывать признаки, имеющие маленькую корреляцию.

БОЛЕЕ СЛОЖНЫЕ МЕТОДЫ

- Filtration methods (фильтрационные методы)
- Wrapping methods (оберточные методы)
- Model selection (встроенный в модель отбор признаков)

1. ФИЛЬТРАЦИОННЫЕ МЕТОДЫ

- **Фильтрационные методы - это отбор признаков по различным статистическим тестам.** Идея метода состоит в вычислении влияния каждого признака в отдельности на целевую переменную (с помощью вычисления некоторой статистики).

Очевидный плюс метода: скорость, так как мы вычисляем значения N статистик, где N - количество признаков.

1. ФИЛЬТРАЦИОННЫЕ МЕТОДЫ

В `sklearn` есть сразу несколько методов, использующих отбор по статистическим критериям. Среди них выделим следующие:

- **SelectKBest** - оставляет k признаков с наибольшим значением выбранной статистики
- **SelectPercentile** - оставляет признаки со значениями выбранной статистики, попавшими в заданную пользователем квантиль

I. СТАТИСТИЧЕСКИЕ ТЕСТЫ ДЛЯ ОТБОРА ПРИЗНАКОВ (ПРИМЕР)

- Тест χ^2 используется в статистике для проверки независимости двух событий.
- Поскольку χ^2 проверяет степень независимости между двумя переменными, а мы хотим сохранить только признаки, наиболее зависимые от метки, то будем вычислять χ^2 между каждым признаком и меткой, сохраняя только признаки с наибольшими значениями.
- Критерий χ^2 можем применять только для бинарных или порядковых признаков.

1. СТАТИСТИЧЕСКИЕ ТЕСТЫ ДЛЯ ОТБОРА ПРИЗНАКОВ

- Статистика χ^2 вычисляется по формуле

$$\chi^2(X; Y) = \sum_{i,j} \frac{(O_{ij} - E_{ij})^2}{E_{ij}},$$

где O_{ij} - наблюдаемая частота, E_{ij} - ожидаемая частота.

Пример: хотим выявить влияние курения на гипертонию:

	Артериальная гипертония есть (1)	Артериальной гипертонии нет (0)	Всего
Курящие (1)	40	30	70
Некурящие (0)	32	48	80
Всего	72	78	150

Вычисляем χ^2 : $\chi^2 = (40-33.6)^2/33.6 + (30-36.4)^2/36.4 + (32-38.4)^2/38.4 + (48-41.6)^2/41.6 = 4.396$.

При отборе признаков оставляем k (или заданную квантиль) признаков с наибольшим значением χ^2 .

I. СТАТИСТИЧЕСКИЕ ТЕСТЫ ДЛЯ ОТБОРА ПРИЗНАКОВ

- mutual information:

для векторов X и Y статистика вычисляется по формуле

$$I(X; Y) = \sum_{y \in Y} \sum_{x \in X} p(x, y) \log\left(\frac{p(x, y)}{p(x)p(y)}\right)$$

- хи-квадрат:

$$\chi^2(X; Y) = \sum_{i,j} \frac{(O_{ij} - E_{ij})^2}{E_{ij}},$$

где O_{ij} - наблюдаемая частота, E_{ij} - ожидаемая частота.

2. ОБЕРТОЧНЫЕ МЕТОДЫ

Оберточные методы используют **жадный отбор признаков**, т.е. последовательно выкидывают наименее подходящие по мнению методов признаки.

В sklearn есть оберточный метод - Recursive Feature Elimination (RFE).

Параметры метода:

- a) алгоритм, используемый для отбора признаков (например, RandomForest)
- b) число признаков, которое мы хотим оставить.

2. ЖАДНЫЙ ОТБОР ПРИЗНАКОВ

1 шаг: Перебираем все признаки и убираем тот, удаление которого сильнее всего уменьшает ошибку

2 шаг: Из оставшихся признаков убираем тот, удаление которого сильнее всего уменьшает ошибку

И т.д.

3. ВСТРОЕННЫЕ В МОДЕЛЬ МЕТОДЫ

Напоминание: L_1 -регуляризация умеет отбирать признаки.

$$Q(w) + \alpha \sum_{i=1}^d |w_j| \rightarrow \min_w$$

Рассмотрим другой вариант регуляризации, которая тоже умеет отбирать признаки (L_0 -регуляризация):

$$Q(w) + \alpha \sum_{i=1}^d [w_j \neq 0] \rightarrow \min_w$$

3. ИНФОРМАЦИОННЫЕ КРИТЕРИИ

- Информационный критерий - мера качества модели, учитывающая степень «подгонки» модели под данные с корректировкой (штрафом) на используемое количество параметров.
- Информационные критерии основаны на **компромиссе между точностью и сложностью модели**. Критерии различаются тем, как они обеспечивают этот баланс.

3. КРИТЕРИЙ AIC

Критерий Акаике (AIC, Akaike Information Criterion)

- Дополнительно предполагаем, что модель a – линейная.

$$AIC(a, X) = Q(a, X) + \frac{2\hat{\sigma}^2}{l} n \rightarrow \min$$

Q – функционал ошибки

$\hat{\sigma}^2$ - оценка дисперсии ошибки $D(y_i - a(x_i))$

n – количество используемых признаков

l – число объектов

- Если Q – среднеквадратичная ошибка для линейной регрессии, и шумы нормально распределены, то

$$AIC = -\ln P + n$$

3. КРИТЕРИЙ ВІС

Критерий Шварца (BIC, Bayesian Information Criterion)

$$BIC(a, X) = \frac{l}{\hat{\sigma}^2} (Q(a, X) + \frac{\hat{\sigma}^2 l n l}{l} n) \rightarrow \min$$

- Если Q – среднеквадратичная ошибка для линейной регрессии, и шумы нормально распределены, то

$$BIC = -\ln P + \frac{n}{2} \ln l$$

3. ОТБОР ПРИЗНАКОВ С ПОМОЩЬЮ ИНФОРМАЦИОННЫХ КРИТЕРИЕВ

- Если в модели k признаков (регрессоров), то существует 2^k всевозможных моделей
- В идеале необходимо построить все 2^k моделей, для каждой посчитать значение критерия качества (AIC, BIC) и выбрать модель, лучшую по этому критерию
- При большом количестве регрессоров используют метод включений-исключений (жадный отбор признаков)

3. ПРИМЕР

Задача предсказания уровня преступности в разных штатах по следующим признакам:

Регрессор

Нулевой коэффициент

Возраст

Южный штат(да/нет)

Образование

Расходы

Труд

Количество мужчин

Численность населения

Безработные (14-24)

Безработные (25-39)

Доход

3. ПРИМЕР: ОТБОР ПРИЗНАКОВ ПО AIC

- Мы решаем задачу линейной регрессии с предположением, что ошибки нормально распределены, поэтому $AIC = \ln P(a, X) - n \rightarrow \max$.

В модели с полным набором регрессоров $AIC = -310.37$. В порядке убывания AIC при удалении каждой из переменных равен:

Численность населения ($AIC = -308$), Труд ($AIC = -309$), Южный штат ($AIC = -309$), Доход ($AIC = -309$), Количество мужчин ($AIC = -310$), Безработные I ($AIC = -310$), Образование ($AIC = -312$), Безработные II ($AIC = -314$), Возраст ($AIC = -315$), Расходы ($AIC = -324$).

Таким образом, имеет смысл удалить переменную “Население”.

3. ПРИМЕР: ОТБОР ПРИЗНАКОВ ПО AIC

Южный штат (AIC = -308), Труд (AIC = -308), Доход (AIC = -308), Количество мужчин (AIC = -309), Безработные I (AIC = -309), Образование (AIC = -310), Безработные II (AIC = -313), Возраст (AIC = -313), Расходы (AIC = -329).

Удаляем переменные до тех пор, пока не удастся больше получить увеличения AIC.

Уровень преступности = 1.2 Возраст + 0.75 Образование + 0.87
Расходы + 0.34 Количество мужчин – 0.86 Безработные I + 2.31
Безработные II.

КАЛИБРОВКА ВЕРОЯТНОСТЕЙ

КАЛИБРОВКА ВЕРОЯТНОСТЕЙ

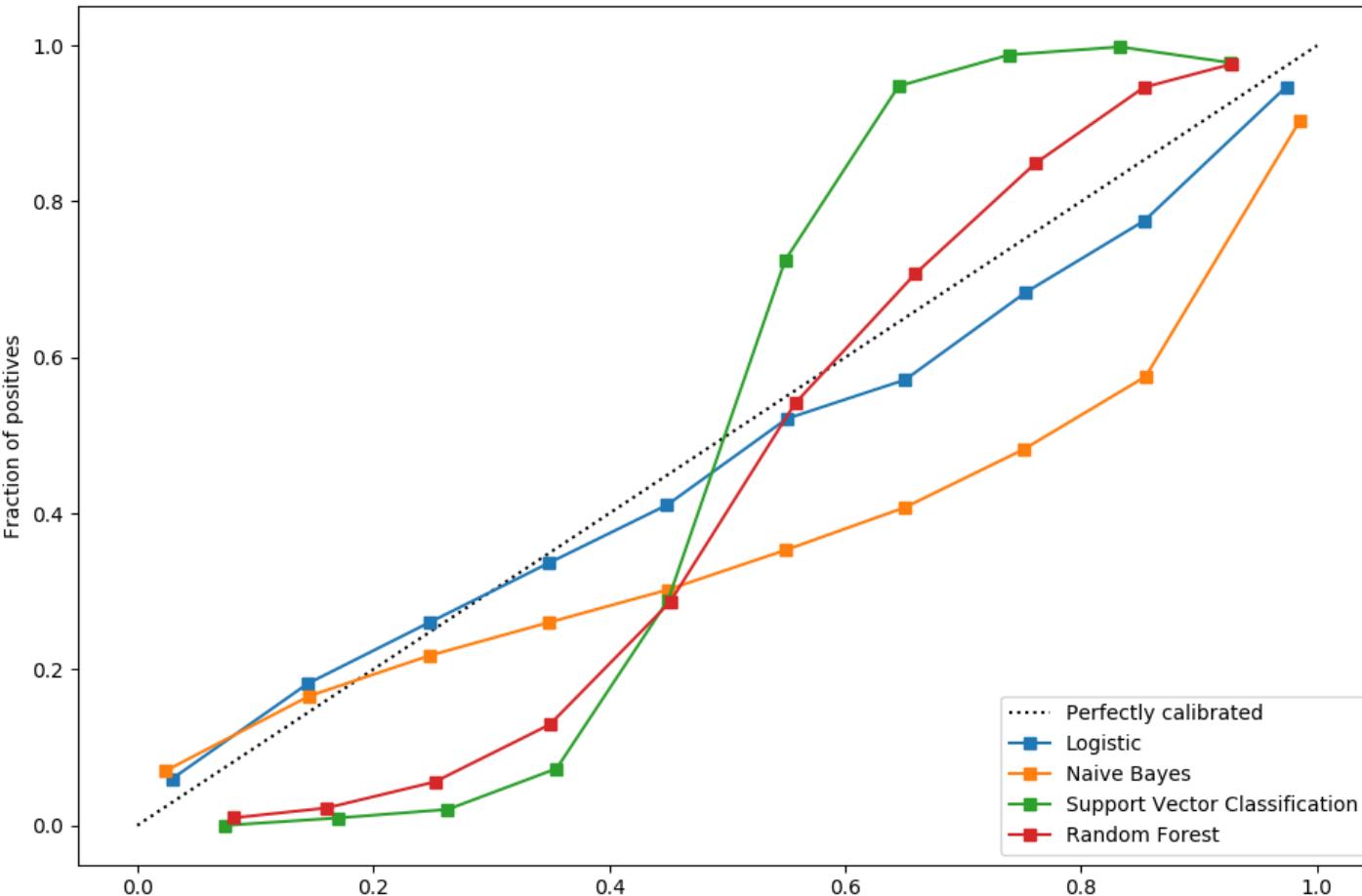
Калибровка вероятностей - приведение ответов алгоритма к значениям, близким к вероятностям объектов принадлежать конкретному классу.

Зачем это нужно?

- Вероятности гораздо проще интерпретировать
- Вероятности могут дать дополнительную информацию о результатах работы алгоритма

ПРИМЕР ИЗ SKLEARN

Calibration plots (reliability curve)



КАЛИБРОВКА ПЛАТТА

- Пусть есть два класса, $Y = \{+1, -1\}$

Задача: для классификатора $a(x)$, предсказывающего значения из отрезка $[0, 1]$, либо предсказывающего класс (+1 или -1), сделать калибровку, чтобы предсказания были вероятностями $p(y = +1|x)$.

Идея: *обучаем логистическую регрессию на ответах классификатора $a(x)$.*

- $\pi(x; \alpha; \beta) = \sigma(\alpha \cdot a(x) + \beta) = \frac{1}{1+e^{-(\alpha \cdot a(x) + \beta)}}$
- Находим α и β , минимизируя логистическую функцию потерь (*то есть обучаем логистическую регрессию*):

$$-\sum_{y_i=-1} \log(1 - \pi(x; \alpha; \beta)) - \sum_{y_i=+1} \log(\pi(x; \alpha; \beta)) \rightarrow \min_{\alpha, \beta}$$