Линейная классификация

Елена Кантонистова

ПЛАН ЛЕКЦИИ

1) Методы кодирования категориальных признаков
2) Переход от регрессии к бинарной классификации
3) Логистическая регрессия

МЕТОДЫ КОДИРОВАНИЯ КАТЕГОРИАЛЬНЫХ ПРИЗНАКОВ

КОДИРОВАНИЕ КАТЕГОРИАЛЬНЫХ ПРИЗНАКОВ: ONE-HOT ENCODING

- Предположим, категориальный признак $f_j(x)$ принимает m различных значений: $C_1, C_2, \ldots, C_m$.

Пример: еда может быть *горькой, сладкой, солёной или кислой* (4 возможных значения признака).

КОДИРОВАНИЕ КАТЕГОРИАЛЬНЫХ ПРИЗНАКОВ: ONE-HOT ENCODING

- Предположим, категориальный признак $f_j(x)$ принимает m различных значений: $C_1, C_2, \ldots, C_m$.

Пример: еда может быть *горькой, сладкой, солёной или кислой* (4 возможных значения признака).

- Заменим категориальный признак на m бинарных признаков: $b_i(x) = [f_j(x) = C_i]$ (индикатор события).

Тогда One-Hot кодировка для нашего примера будет следующей:

горький = (1,0,0,0), *сладкий* = (0,1,0,0),

солёный = (0,0,1,0), *кислый* = (0,0,0,1).

СЧЁТЧИКИ

Счётчик (*mean target encoding*) – это вероятность получить значение целевой переменной для данного значения категориального признака.

СЧЁТЧИКИ (ПРИМЕР)

	feature	target
0	Moscow	0
1	Moscow	1
2	Moscow	1
3	Moscow	0
4	Moscow	0
5	Tver	1
6	Tver	1
7	Tver	1
8	Tver	0
9	Klin	0
10	Klin	0
11	Tver	1

СЧЁТЧИКИ (ПРИМЕР)

	feature	target
0	Moscow	0
1	Moscow	1
2	Moscow	1
3	Moscow	0
4	Moscow	0
5	Tver	1
6	Tver	1
7	Tver	1
8	Tver	0
9	Klin	0
10	Klin	0
11	Tver	1

	feature	feature_mean	target
0	Moscow	0.4	0
1	Moscow	0.4	1
2	Moscow	0.4	1
3	Moscow	0.4	0
4	Moscow	0.4	0
5	Tver	0.8	1
6	Tver	0.8	1
7	Tver	0.8	1
8	Tver	0.8	0
9	Klin	0.0	0
10	Klin	0.0	0
11	Tver	0.8	1

СЧЁТЧИКИ: ПРИМЕР

city	target	0	1	2
Moscow	1	1/4	1/2	1/4
London	0	1/2	0	1/2
London	2	1/2	0	1/2
Kiev	1	1/2	1/2	0
Moscow	1	1/4	1/2	1/4
Moscow	0	1/4	1/2	1/4
Kiev	0	1/2	1/2	0
Moscow	2	1/4	1/2	1/4

СЧЁТЧИКИ В ЗАДАЧЕ БИНАРНОЙ КЛАССИФИКАЦИИ

В случае бинарной классификации счётчики можно задать формулой:

$$Likelihood = \frac{Goods}{Goods + Bads} = mean(target),$$

где *Goods* – число единиц в столбце target,

Bads – число нулей в столбце target.

СЧЁТЧИКИ (ОБЩАЯ ФОРМУЛА)

- Пусть целевая переменная y принимает значения от 1 до K.
- Закодируем категориальную переменную $f(x)$ следующим способом:

$$counts(u, X) = \sum_{(x,y)\in X} [f(x) = u]$$

$$successes_k(u, X) = \sum_{(x,y)\in X} [f(x) = u][y = k], k = 1, \dots, K$$

Тогда кодировка:

$$\boldsymbol{mean\_target_k(x, X) = \frac{successes_k(f(x), X)}{counts(f(x), X)} \approx p(y = k|f(x))}$$

СЧЁТЧИКИ (ОБЩАЯ ФОРМУЛА)

$$counts(u, X) = \sum_{(x,y)\in X} [f(x) = u]$$

$$successes_k(u, X) = \sum_{(x,y)\in X} [f(x) = u][y = k], k = 1, \dots, K$$

Тогда кодировка:

$$\boldsymbol{mean\_target_k(x, X) = \frac{successes_k(f(x), X)}{counts(f(x), X)}}$$

Недостатки? Когда такой способ кодирования может переобучить наш алгоритм?

СЧЁТЧИКИ: ОПАСНОСТИ

- *Вычисляя счётчики, мы закладываем в признаки информацию о целевой переменной и, тем самым, переобучаемся*
- *Если в данных есть редкие категории, то счетчики на них переобучатся*

РЕШЕНИЕ 1: СЧЁТЧИКИ + СГЛАЖИВАНИЕ

Используем счётчики (mean target encoding) со сглаживанием:

$$\frac{\boldsymbol{mean(target) \cdot n_{rows} + global\ mean \cdot \alpha}}{\boldsymbol{n_{rows} + \alpha}},$$

n_{rows} - количество строк в категории,

α – параметр регуляризации.

РЕШЕНИЕ 2: ОТЛОЖЕННАЯ ВЫБОРКА ИЛИ КРОСС-ВАЛИДАЦИЯ

- Можно вычислять счётчики так:

city	target
Moscow	1
London	0
London	2
Kiev	1
Moscow	1
Moscow	0
Kiev	0
Moscow	2

Вычисляем счетчики по этой части

Кодируем признак вычисленными счётчиками и обучаемся по этой части

РЕШЕНИЕ 2: ОТЛОЖЕННАЯ ВЫБОРКА ИЛИ КРОСС-ВАЛИДАЦИЯ

Более продвинутый способ (по кросс-валидации):

1) Разбиваем выборку

на m частей $X_1, \dots, X_m$

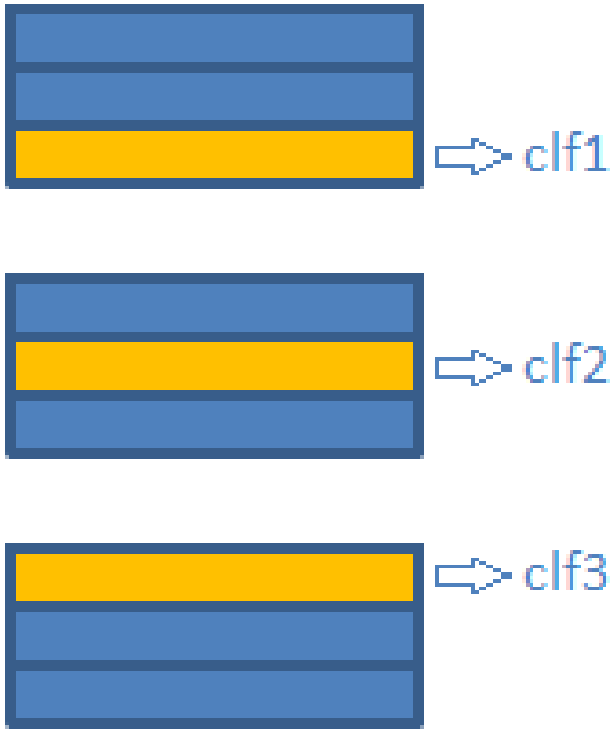
2) На каждой части X_i

значения признаков

вычисляются по

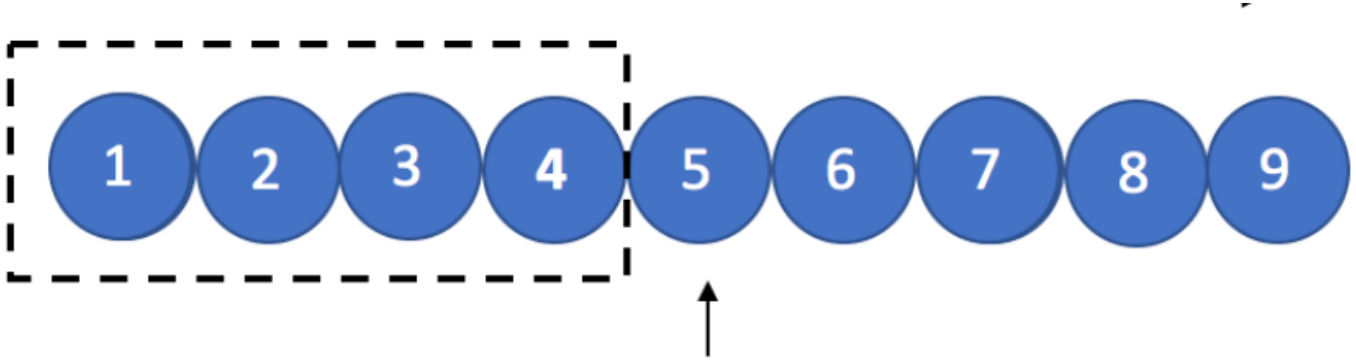
оставшимся частям:

$$x \in X_i \Rightarrow g_k(x) = g_k(x, X \backslash X_i)$$



РЕШЕНИЕ 3: ВЫЧИСЛЕНИЕ СЧЕТЧИКОВ С ПОМОЩЬЮ СХЕМЫ EXPANDING MEAN

Суть схемы заключается в том, чтобы пройти по отсортированному в определенном порядке датасету и для подсчета счетчика для строки *m* использовать строки от *0* до *m-1*.



Running mean calculation.
Numbers are assigned randomly to each observation. Only 1-4 are used to find encoding for 5

БОРЬБА С ПЕРЕОБУЧЕНИЕМ В СЧЁТЧИКАХ

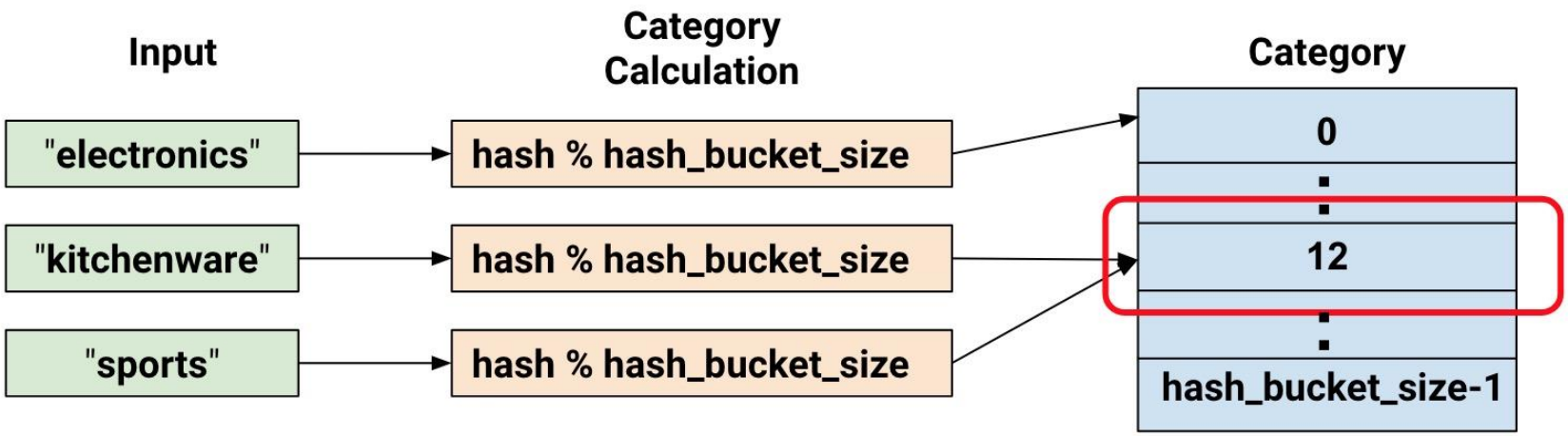
- Вычисление счётчиков по кросс-валидации
- Сглаживание
- Добавление случайных шумов
- Expanding mean

ХЭШИРОВАНИЕ ПРИЗНАКОВ

- Если у категориального признака слишком много значений, скажем, миллион, то после применения one-hot кодировки мы получим миллион новых столбцов. С такой огромной матрицей тяжело работать.
- Хэширование развивает идею one-hot кодирования, но позволяет получать любое заранее заданное число новых числовых столбцов после кодировки.

АЛГОРИТМ ХЭШИРОВАНИЯ

1) Для каждого значения признака вычисляем значение некоторой функции – хэш-функции (hash)
2) Задаем hash_bucket_size – итоговое количество различных значений категориального признака.
3) Берем остаток: hash % hash_bucket_size – тем самым кодируем каждое значение признака числом от 0 до hash_bucket_size-1.
4) Дальше к полученным числам применяем OHE.

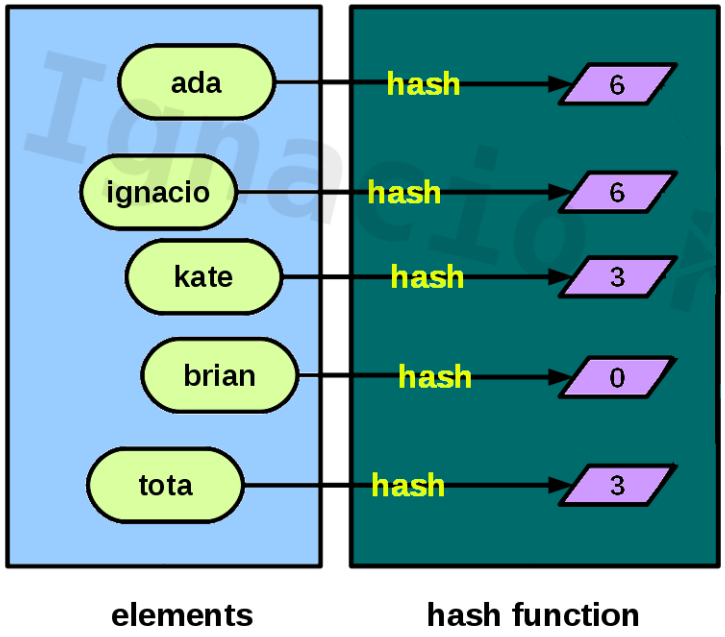


ЧТО ДЕЛАЕТ ХЭШ-ФУНКЦИЯ

Идея: *хэш-функция группирует значения категориального признака:*

- *часто встречающиеся значения признака формируют отдельные группы*
- *редко встречающиеся значения попадают в одну группу при группировке*

ХЭШИРОВАНИЕ ПРИЗНАКОВ: ПРИМЕР



0	1	2	3	4	5	6
0	0	0	0	0	0	1
0	0	0	0	0	0	1
0	0	0	1	0	0	0
1	0	0	0	0	0	0
0	0	0	1	0	0	0

ХЭШИРОВАНИЕ

- Хэширование – это способ кодирования категориальных данных, принимающих множество различных значений, показывающий хорошие результаты на практике.
- **Хэширование позволяет закодировать любое значение категориального признака (в том числе то, которого не было в тренировочной выборке).**

Статья про хэширование:

https://arxiv.org/abs/1509.05472

ЧТО ПОЧИТАТЬ ПРО КОДИРОВАНИЕ КАТЕГОРИАЛЬНЫХ ПРИЗНАКОВ

- Лекция Жени Соколова
- Блог Александра Дьяконова
- Кусочек статьи с Хабра про хеширование

ЛИНЕЙНЫЕ МОДЕЛИ КЛАССИФИКАЦИИ

ОБУЧЕНИЕ ЛИНЕЙНОЙ РЕГРЕССИИ (НАПОМИНАНИЕ)

Обучающая выборка:

пусть $\boldsymbol{x}$ – объект ($x_1, x_2, \dots, x_l$ - его признаки), а y – ответ на объекте (произвольное число), n – количество объектов.

Модель линейной регрессии:

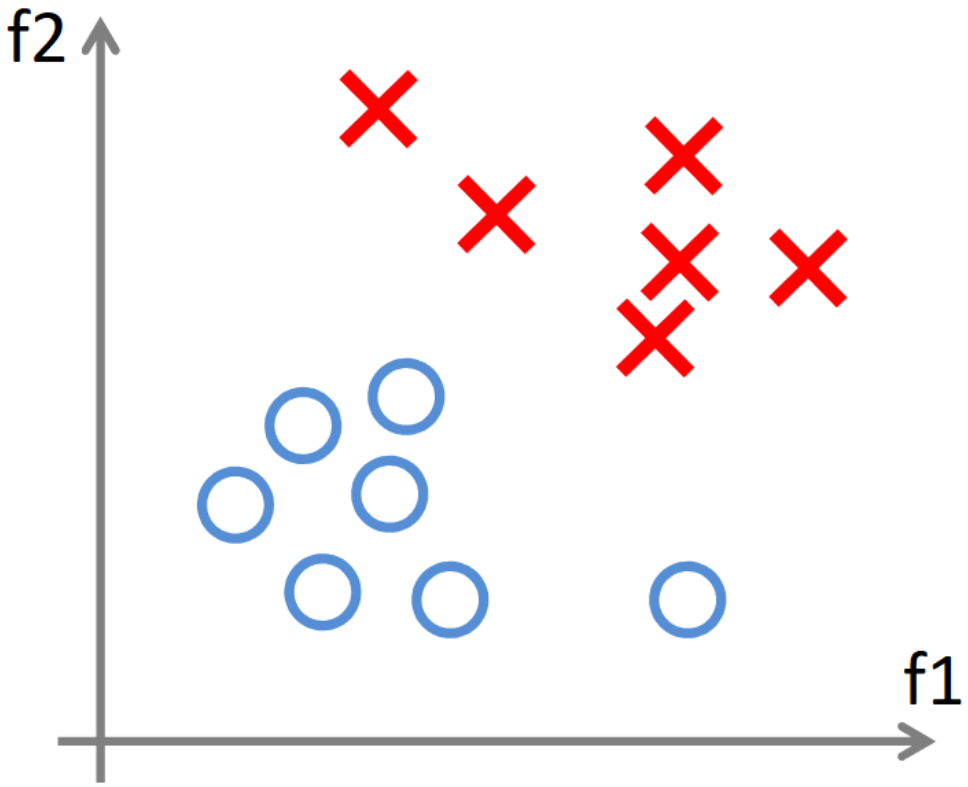
$$a(\boldsymbol{x}, w) = \sum_{i=1}^{l} w_j x_j$$

- Метод обучения – метод наименьших квадратов (***минимизируем разность между предсказанием и правильным ответом***):

$$Q(w) = \sum_{i=1}^{n} (a(\boldsymbol{x_i}, w) - y_i)^2 \to \min_w$$

БИНАРНАЯ КЛАССИФИКАЦИЯ

$y_1, y_2, \ldots, y_n$ - ответы (***+1 или -1***).



Как выглядит модель линейного классификатора: $\boldsymbol{a}(\boldsymbol{x}, \boldsymbol{w}) = ?$

БИНАРНАЯ КЛАССИФИКАЦИЯ

Модель линейного классификатора:

$$a(x, w) = sign(\sum_{j=1}^{l} w_j x_j)$$

БИНАРНАЯ КЛАССИФИКАЦИЯ

Модель линейного классификатора:

$$\boldsymbol{a(x,w) = sign(\sum_{j=1}^{l} w_j x_j)}$$

- если $\sum_{j=1}^{l} w_j x_j > 0$, то $sign(\sum_{j=1}^{l} w_j x_j) = +1$, то есть объект отнесён к положительному классу
- если $\sum_{j=1}^{l} w_j x_j < 0$, то $sign(\sum_{j=1}^{l} w_j x_j) = -1$, то есть объект отнесён к отрицательному классу

БИНАРНАЯ КЛАССИФИКАЦИЯ

Модель линейного классификатора:

$$\boldsymbol{a(x, w) = sign(\sum_{j=1}^{l} w_j x_j)}$$

- если $\sum_{j=1}^{l} w_j x_j > 0$, то $sign(\sum_{j=1}^{l} w_j x_j) = +1$, то есть объект отнесён к положительному классу
- если $\sum_{j=1}^{l} w_j x_j < 0$, то $sign(\sum_{j=1}^{l} w_j x_j) = -1$, то есть объект отнесён к отрицательному классу
- значит, $\sum_{j=1}^{l} w_j x_j = 0$ – *уравнение разделяющей границы* между классами. *Это уравнение плоскости* (или прямой в двумерном случае), поэтому *классификатор является линейным*.

БИНАРНАЯ КЛАССИФИКАЦИЯ

Модель линейного классификатора:

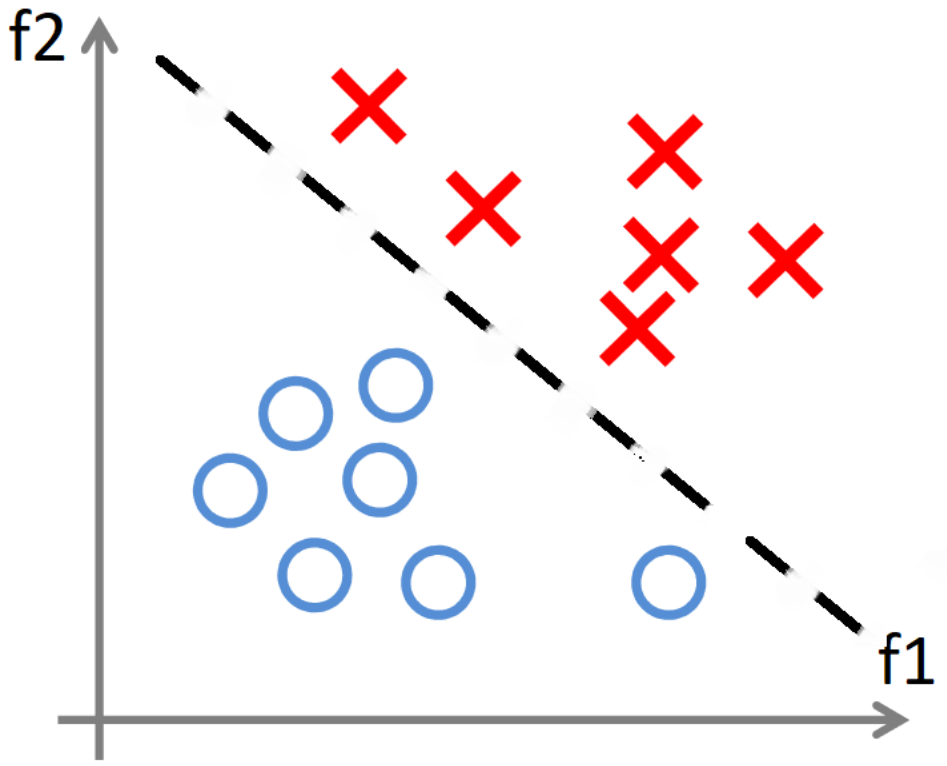
$$a(x, w) = sign(\sum_{j=1}^{l} w_j x_j)$$

Уравнение

$\sum_{j=1}^{l} w_j x_j = 0$

– уравнение плоскости

(или прямой).



ОБУЧЕНИЕ КЛАССИФИКАТОРА

Как обучить линейный классификатор?

ОБУЧЕНИЕ КЛАССИФИКАТОРА

- Обучение - минимизация доли ошибок классификатора:

$$Q(a, X) = \frac{1}{n} \sum_{i=1}^{n} [\boldsymbol{a(x_i) \neq y_i}] \rightarrow min\,,$$

где $[a(\boldsymbol{x_i}) \neq y_i] = 1$, если предсказание на объекте неверное, то есть $a(\boldsymbol{x_i}) \neq y_i$, и 0 иначе.

ОБУЧЕНИЕ КЛАССИФИКАТОРА

- Обучение - минимизация доли ошибок классификатора:

$$Q(a, X) = \frac{1}{n}\sum_{i=1}^{n} [\boldsymbol{a(x_i) \neq y_i}] \rightarrow min \ (*),$$

где $[a(\boldsymbol{x_i}) \neq y_i] = 1$, если предсказание на объекте неверное, то есть $a(\boldsymbol{x_i}) \neq y_i$, и 0 иначе.

- Обозначим $M_i = y_i \cdot (w, \boldsymbol{x_i})$ - ***отступ*** *на i-м объекте*.

ОБУЧЕНИЕ КЛАССИФИКАТОРА

- Обучение - минимизация доли ошибок классификатора:

$$Q(a, X) = \frac{1}{n}\sum_{i=1}^{n} [\boldsymbol{a(x_i) \neq y_i}] \rightarrow min \ (*),$$

где $[a(\boldsymbol{x_i}) \neq y_i] = 1$, если предсказание на объекте неверное, то есть $a(\boldsymbol{x_i}) \neq y_i$, и 0 иначе.

- Обозначим $M_i = y_i \cdot (w, \boldsymbol{x_i})$ - ***отступ*** *на i-м объекте*.

Утверждение. Решение задачи (*) эквивалентно решению задачи

$$Q(a, X) = \frac{1}{n}\sum_{i=1}^{n} [\boldsymbol{M_i < 0}] \rightarrow min$$

ДОКАЗАТЕЛЬСТВО УТВЕРЖДЕНИЯ

- Обучение - минимизация доли ошибок классификатора:

$$Q(a, X) = \frac{1}{n}\sum_{i=1}^{n}[a(x_i) \neq y_i] = \frac{1}{n}\sum_{i=1}^{n}[sign(w, x_i) \neq y_i] \to min$$

Функционал Q можно переписать в виде:

$$Q(a, X) = \frac{1}{n}\sum_{i=1}^{n}[y_i \cdot (w, x_i) < 0] = \frac{1}{n}\sum_{i=1}^{n}[M_i < 0] \to min$$

- $M_i = y_i \cdot (w, x_i)$ - ***отступ***

ОТСТУП (MARGIN)

Знак отступа $M = y \cdot (w, x)$ говорит о корректности классификации на объекте:

ОТСТУП (MARGIN)

Знак отступа $M = y \cdot (w, x)$ говорит о корректности классификации на объекте:

Случаи неверной классификации (предсказание не совпадает с правильным ответом):

- Если $(w, x) > 0$ (то есть объект отнесён к классу $+1$), а $y = -1$, то $M = y \cdot (w, x) < 0$.

ОТСТУП (MARGIN)

Знак отступа $M = y \cdot (w, x)$ говорит о корректности классификации на объекте:

Случаи неверной классификации (предсказание не совпадает с правильным ответом):

- Если $(w, x) > 0$ (то есть объект отнесён к классу $+1$), а $y = -1$, то $M = \mathrm{y} \cdot (w, x) < \mathbf{0}$.
- Аналогично, если $(w, x) < 0$, а $y = +1$, то

$M = \mathrm{y} \cdot (w, x) < \mathbf{0}$.

ОТСТУП (MARGIN)

Знак отступа $M = y \cdot (w, x)$ говорит о корректности классификации на объекте:

Случаи неверной классификации (предсказание не совпадает с правильным ответом):

- Если $(w, x) > 0$ (то есть объект отнесён к классу $+1$), а $y = -1$, то $M = y \cdot (w, x) < 0$.
- Аналогично, если $(w, x) < 0$, а $y = +1$, то

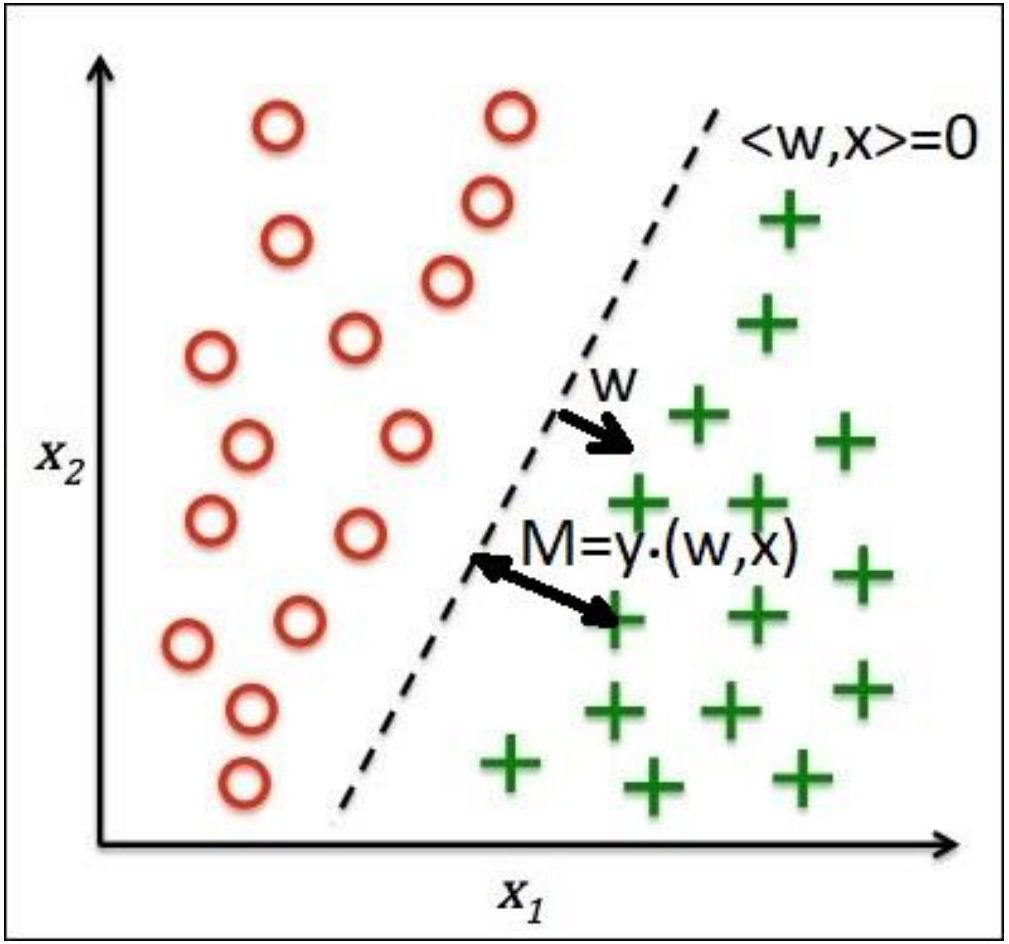
$M = y \cdot (w, x) < 0$.

Случаи верной классификации:

- Если $(w, x) > 0$ и $y = +1$ или $(w, x) < 0$ и $y = -1$ получаем $M = y \cdot (w, x) > 0$.

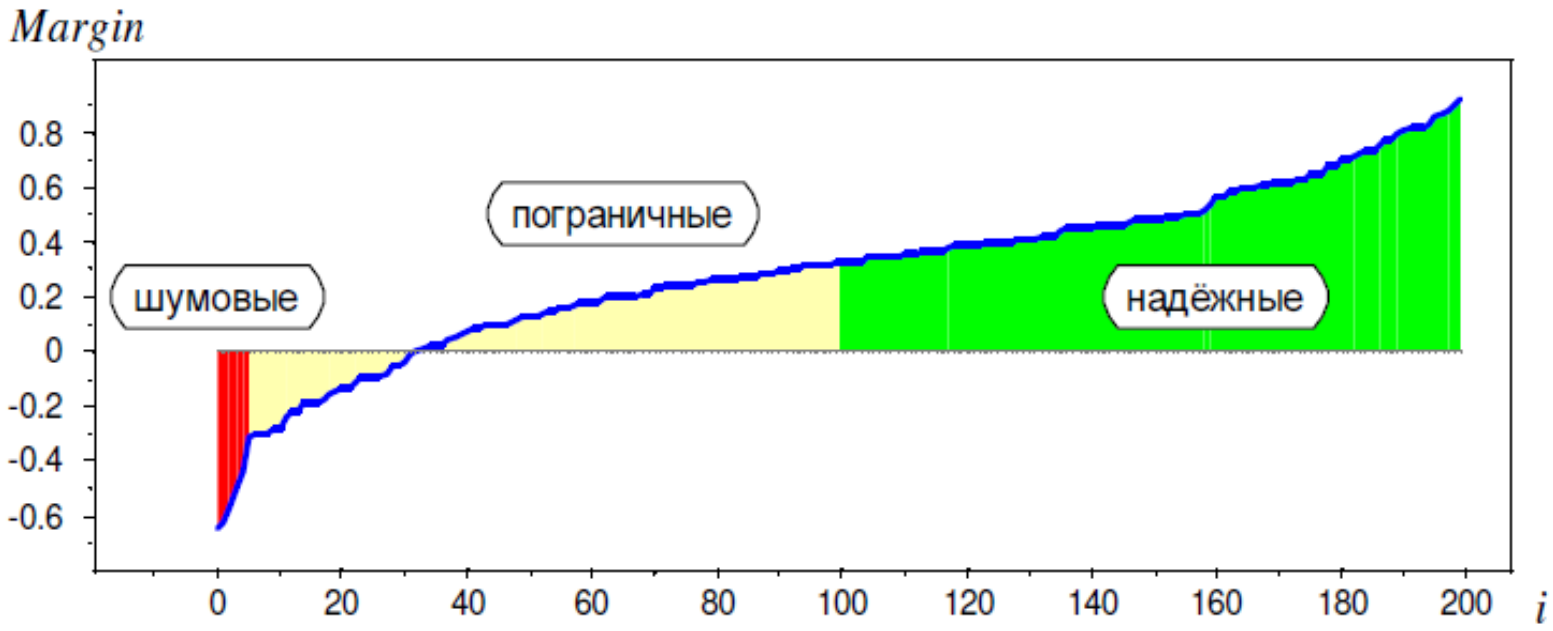
ОТСТУП (MARGIN)

Абсолютная величина отступа M обозначает степень уверенности классификатора в ответе (чем ближе M к нулю, тем меньше уверенность в ответе)



ОТСТУП (MARGIN)

Ранжирование объектов по возрастанию отступа:



ПОРОГОВАЯ ФУНКЦИЯ ПОТЕРЬ

Ранее мы показали, что обучение классификатора – это минимизация ***пороговой функции потерь***:

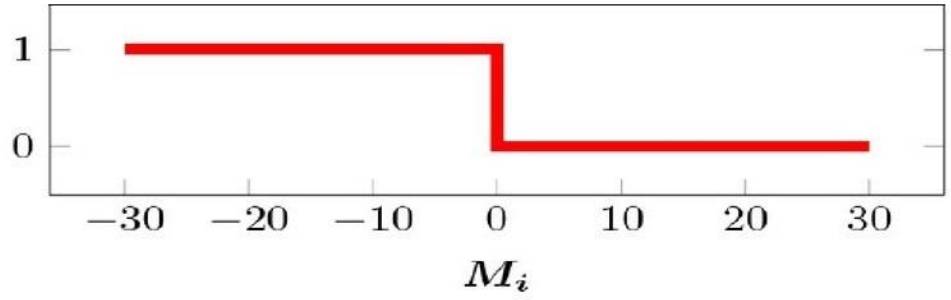
$$Q(a, X) = \frac{1}{n} \sum_{i=1}^{n} [\boldsymbol{M_i < 0}] \rightarrow min$$

ПОРОГОВАЯ ФУНКЦИЯ ПОТЕРЬ

Ранее мы показали, что обучение классификатора – это минимизация ***пороговой функции потерь***:

$$Q(a, X) = \frac{1}{n} \sum_{i=1}^{n} [\boldsymbol{M_i < 0}] \rightarrow min$$

- Пороговая функция потерь ***разрывна***, и этот факт сильно затрудняет процесс минимизации.

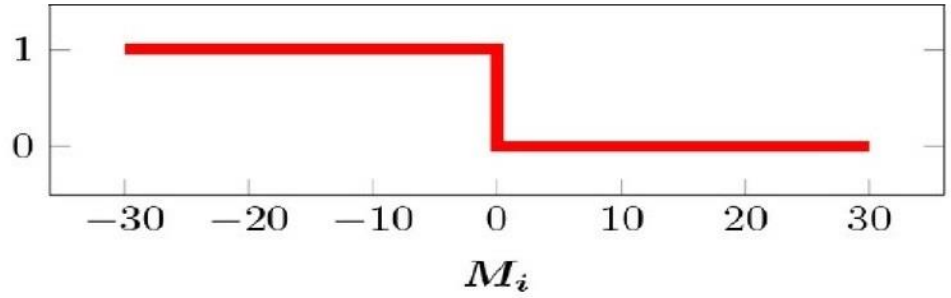


ПОРОГОВАЯ ФУНКЦИЯ ПОТЕРЬ

Ранее мы показали, что обучение классификатора – это минимизация ***пороговой функции потерь***:

$$Q(a, X) = \frac{1}{n}\sum_{i=1}^{n}[\boldsymbol{M_i < 0}] \to min$$

- Пороговая функция потерь разрывна, и этот факт сильно затрудняет процесс минимизации.



- Для решения этой проблемы используют ***другие функции потерь – непрерывные или гладкие, как правило, являющиеся верхними оценками пороговой функции***.

ПОРОГОВАЯ ФУНКЦИЯ ПОТЕРЬ

Ранее мы показали, что обучение классификатора – это минимизация ***пороговой функции потерь***:

$$Q(a, X) = \frac{1}{n}\sum_{i=1}^{n} [\boldsymbol{M_i < 0}] \rightarrow min$$

- Пороговая функция потерь разрывна, и этот факт сильно затрудняет процесс минимизации.
- Для решения этой проблемы используют другие функции потерь – непрерывные или гладкие, как правило, являющиеся верхними оценками пороговой функции.
- Задача минимизации некоторой функции потерь называется ***минимизацией эмпирического риска*** (сама функция потерь – эмпирический риск).

ВЕРХНИЕ ОЦЕНКИ ЭМПИРИЧЕСКОГО РИСКА

- $L(a, y) = L(M) = [M < 0]$ – разрывная функция потерь

Оценим

$\boldsymbol{L(M) \leq \tilde{L}(M)}$, где $\tilde{L}(M)$ - непрерывная или гладкая функция потерь.

- Тогда

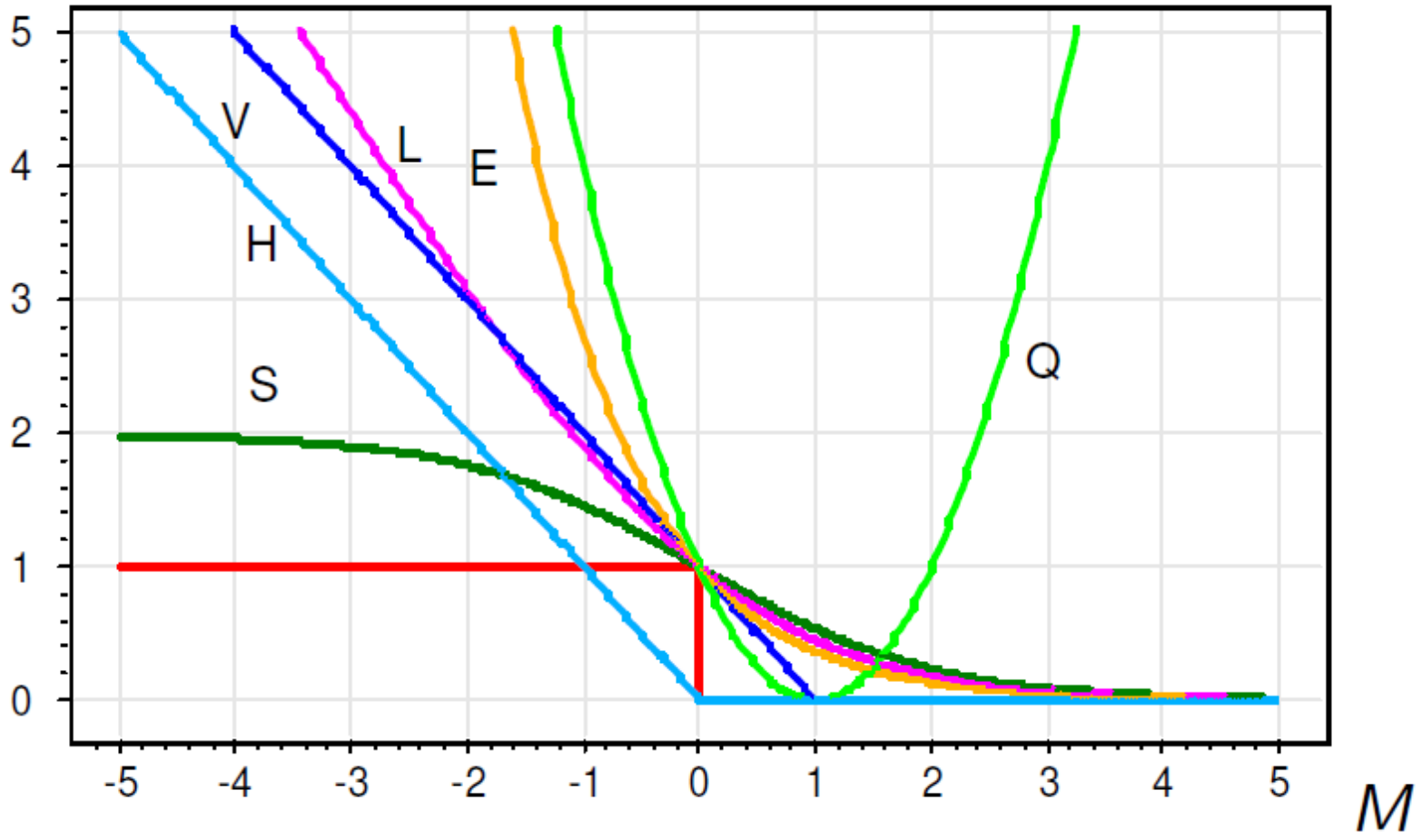
$$Q(a, X) = \frac{1}{n}\sum_{i=1}^{n} L(y_i \cdot (w, x_i)) \leq \frac{1}{n}\sum_{i=1}^{n} \tilde{L}\big(y_i \cdot (w, x_i)\big) \to min$$

ФУНКЦИИ ПОТЕРЬ

Минимизируя различные функции потерь, получаем разные результаты. Поэтому разные функции потерь определяют различные классификаторы.

- $L(M) = \log(1 + e^{-M})$ – логистическая функция потерь
- $\mathrm{V}(M) = (1 - \mathrm{M})_+ = \max(0,1 - M)$ – кусочно-линейная функция потерь (метод опорных векторов)
- $H(M) = (-\mathrm{M})_+ = \max(0, -M)$ – кусочно-линейная функция потерь (персептрон)
- $E(M) = e^{-M}$ - экспоненциальная функция потерь
- $S(M) = \frac{2}{1+e^{-M}}$ - сигмоидная функция потерь
- $[M < 0]$ – пороговая функция потерь

ФУНКЦИИ ПОТЕРЬ



ОПТИМИЗАЦИЯ ФУНКЦИОНАЛА ПОТЕРЬ

- Нахождение минимума функции потерь $\boldsymbol{Q}$ происходит с помощью метода градиентного спуска:

$$\boldsymbol{w^{(k)} = w^{(k-1)} - \eta \cdot \nabla Q(w^{(k-1)})}$$