Лекция 5 Линейная классификация.

Кантонистова Е.О.

ВШЭ, 2022

ПЛАН ЛЕКЦИИ

1) Линейная классификация
2) Логистическая регрессия
3) Персептрон

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:

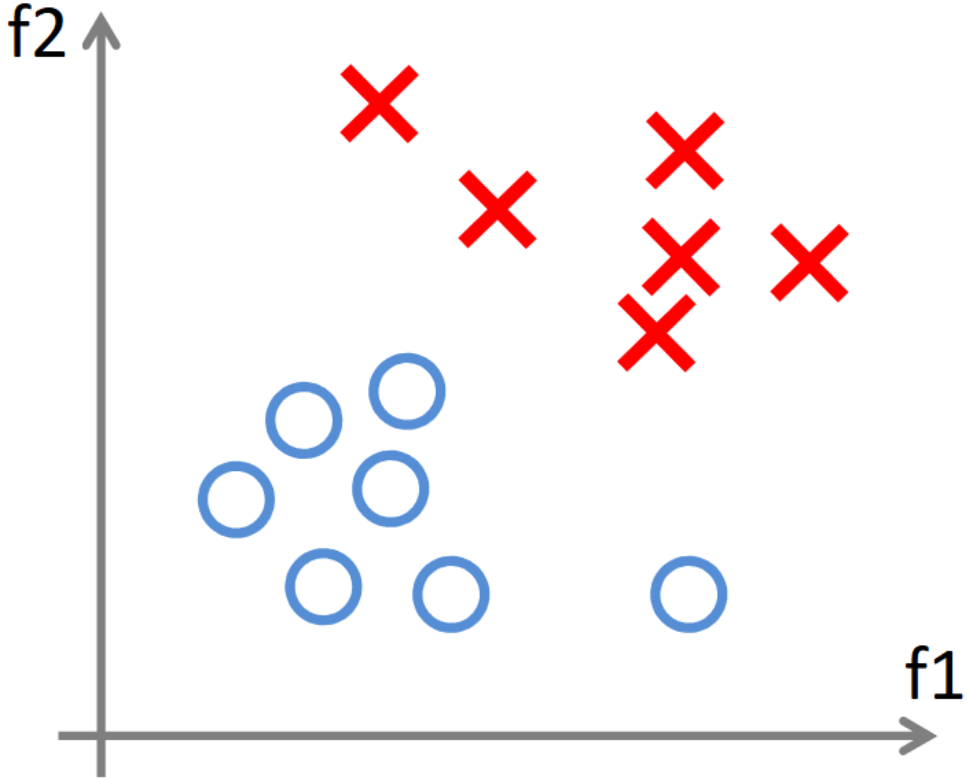
$$a(\boldsymbol{x}, w) = \sum_{i=1}^{l} w_j x_j$$

- 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***).



БИНАРНАЯ КЛАССИФИКАЦИЯ

Модель линейного классификатора:

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

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
- 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*.

ОБУЧЕНИЕ КЛАССИФИКАТОРА

- Обучение - минимизация доли ошибок классификатора:

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

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*.
- Решение задачи (*) эквивалентно решению задачи

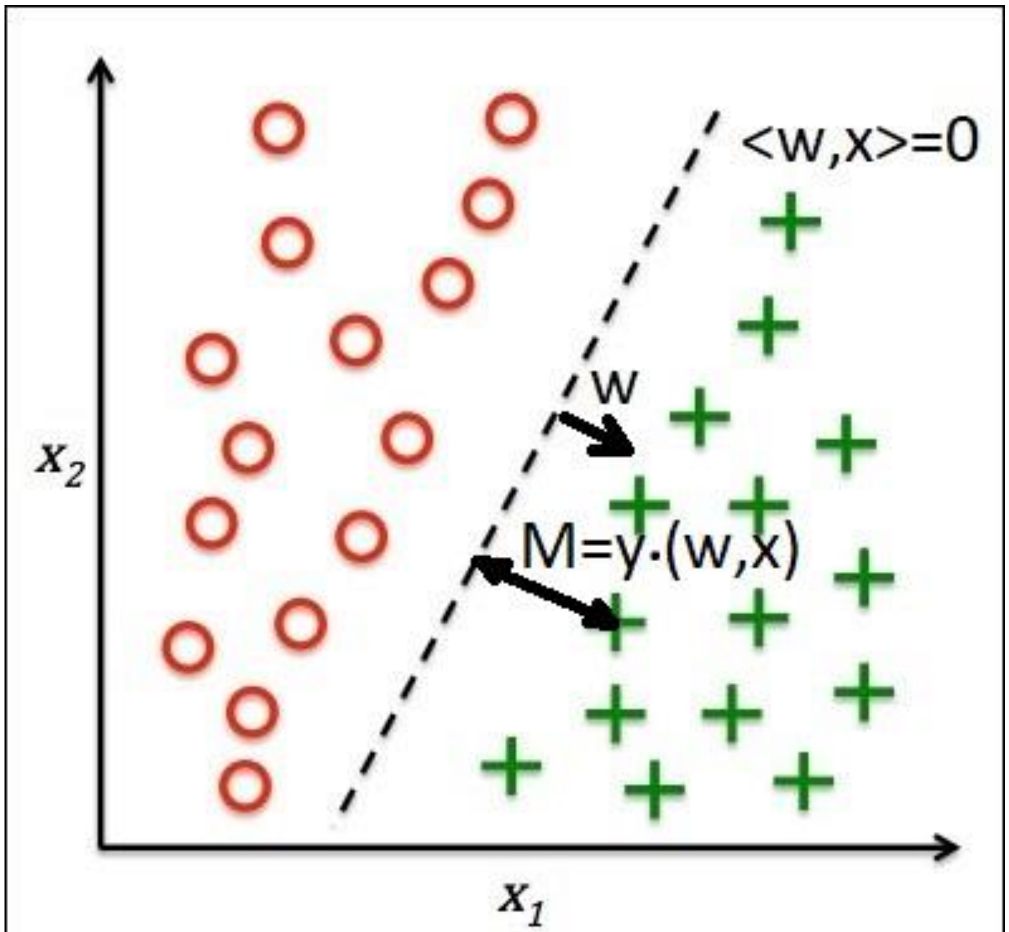
$$Q(a,X) = \frac{1}{n}\sum_{i=1}^{n}[\boldsymbol{M_i} < \boldsymbol{0}] \rightarrow min$$

ОТСТУП (MARGIN)

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

ОТСТУП (MARGIN)

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



ВЕРХНИЕ ОЦЕНКИ ЭМПИРИЧЕСКОГО РИСКА

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

Оценим

$\boldsymbol{L}(\boldsymbol{M}) \leq \ \tilde{\boldsymbol{L}}(\boldsymbol{M})$, 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

ОПТИМИЗАЦИЯ ФУНКЦИОНАЛА ПОТЕРЬ

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

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

ЛОГИСТИЧЕСКАЯ РЕГРЕССИЯ

Хотим предсказывать не классы, а вероятности классов.

- Линейная регрессия: $a(x, w) = (x, w) = w^T x \in \mathbb{R}$
- Логистическая регрессия: $\boldsymbol{a(x, w) = \sigma(w^T x)}$,

где $\sigma(z) = \frac{1}{1+e^{-z}}$ - сигмоида (логистическая функция)

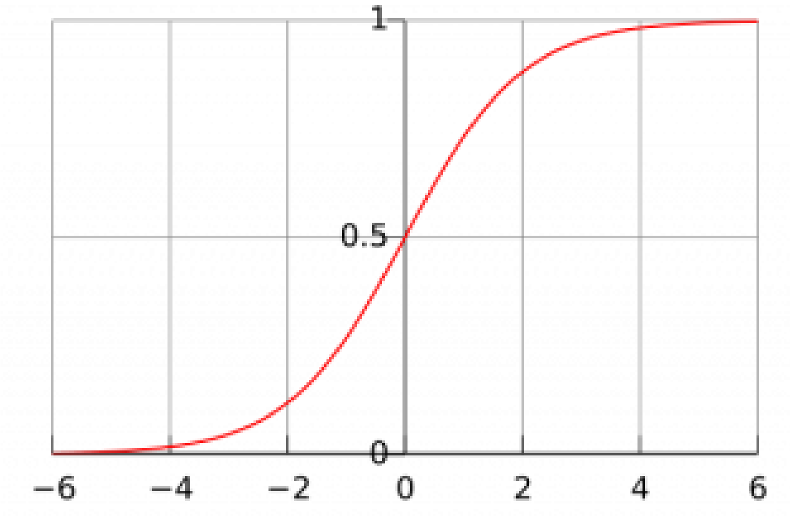
ЛОГИСТИЧЕСКАЯ РЕГРЕССИЯ

Хотим предсказывать не классы, а вероятности классов.

- Линейная регрессия: $a(x, w) = (x, w) = w^T x \in \mathbb{R}$
- Логистическая регрессия: $\boldsymbol{a(x, w) = \sigma(w^T x)}$,

где $\sigma(z) = \frac{1}{1+e^{-z}}$ - сигмоида (логистическая функция),

$\sigma(z) \in (0; 1)$.



Логистическая регрессия: $\boldsymbol{a(x, w) = \frac{1}{1+e^{-w^T x}}}$

ВЕРОЯТНОСТНЫЙ СМЫСЛ

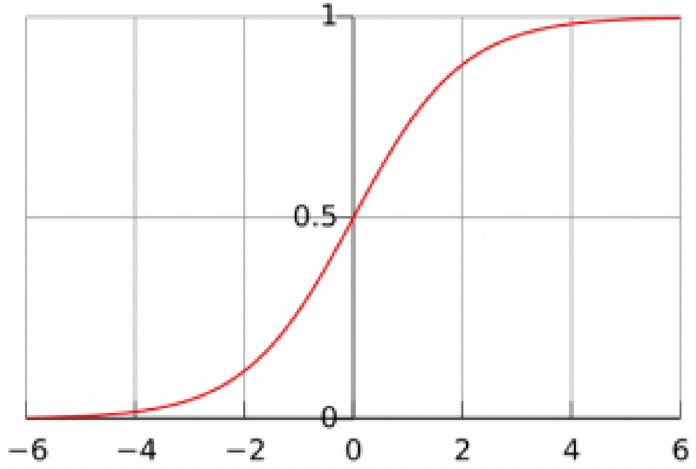
Утверждение. $a(x, w)$ **–** вероятность того, что $y = +1$ на объекте x, т.е.

$$\boldsymbol{a(x, w) = P(y = +1 | x; w)}$$

Доказательство. Дальше в лекции.

РАЗДЕЛЯЮЩАЯ ГРАНИЦА

Предсказываем $y = +1$, если $a(x, w) \geq 0.5$.



$a(x, w) = \sigma(w^T x) \geq 0.5$, если $w^T x \geq 0$.

Получаем, что

- $y = +1$ при $w^T x \geq 0$
- $y = -1$ при $w^T x < 0$,

т.е. $w^T x = 0$ – разделяющая гиперплоскость.

ЛОГИСТИЧЕСКАЯ РЕГРЕССИЯ

Логистическая регрессия - это линейный классификатор!

ФУНКЦИЯ ПОТЕРЬ ЛОГИСТИЧЕСКОЙ РЕГРЕССИИ

Если взять квадратичную функцию потерь

$L(a, y) = (a - y)^2$,

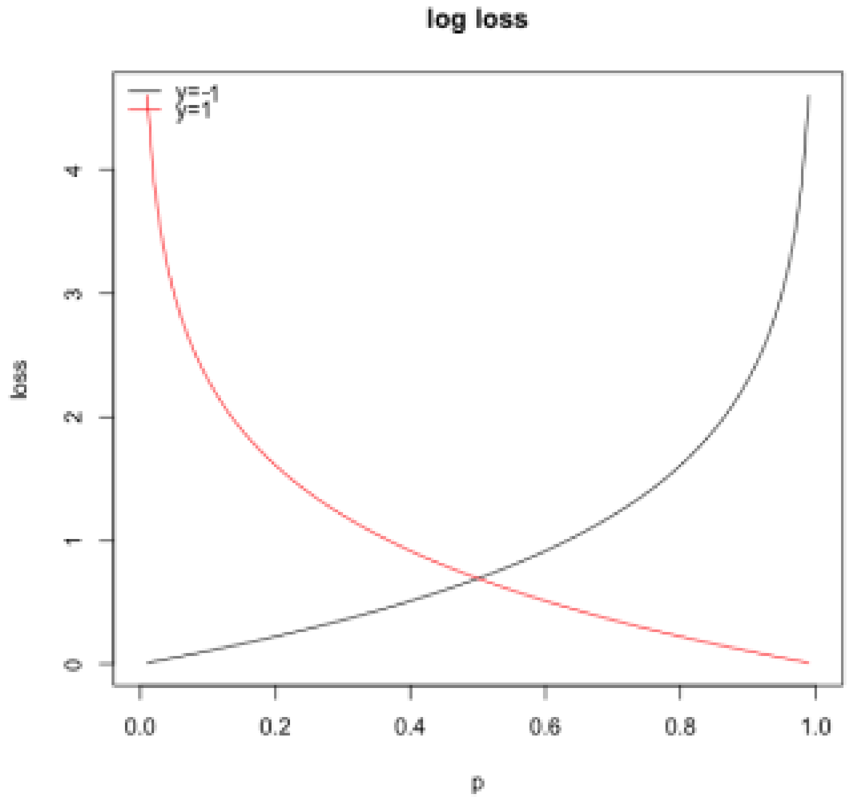
то возникнут проблемы:

- $Q(a, X) = \frac{1}{l}\sum_{i=1}^{l}\left(\frac{1}{1+e^{-w^T x}} - y\right)^2$ - не выпуклая функция (можем не попасть в глобальный минимум при оптимизации)
- На совсем неправильном предсказании маленький штраф (пусть предсказали вероятность 0% на объекте класса $y = +1$, тогда штраф всего $(1 - 0)^2 = 1$)

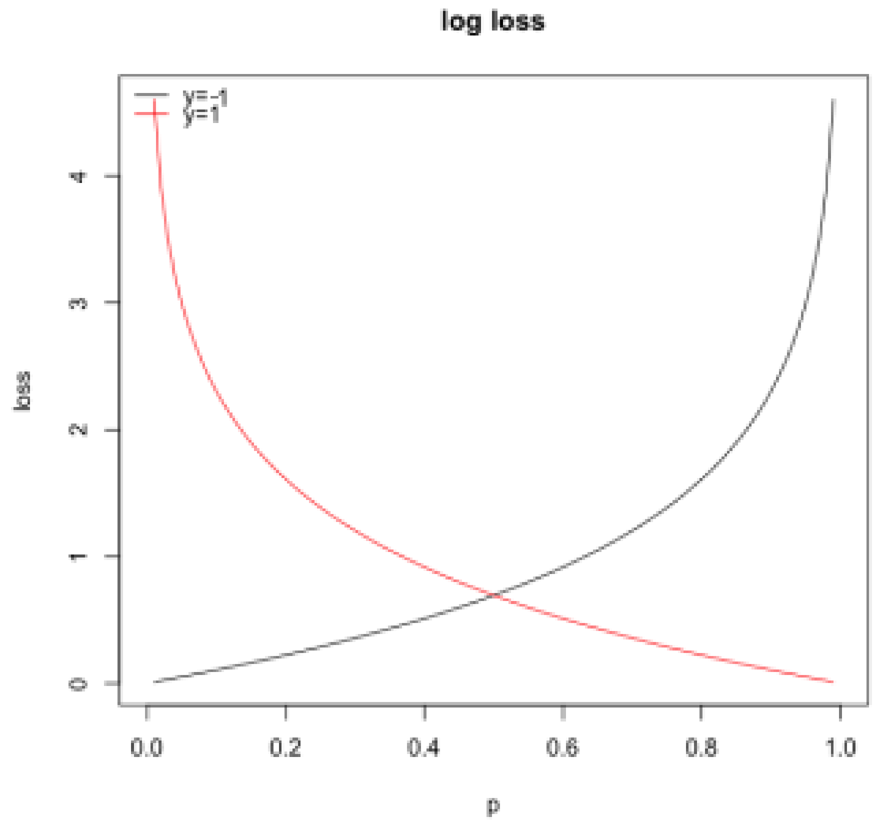
ФУНКЦИЯ ПОТЕРЬ ЛОГИСТИЧЕСКОЙ РЕГРЕССИИ

Возьмем логистическую функцию потерь (***log-loss***):

$$Q(w) = -\sum_{i=1}^{l}\left([y_i = +1] \cdot \log\big(a(x_i, w)\big) + [y_i = -1] \cdot \log\big(1 - a(x_i, w)\big)\right)$$



ЛОГИСТИЧЕСКАЯ ФУНКЦИЯ ПОТЕРЬ



- если $a(x,w) = 1$ и $y = +1$, то штраф $L(a,y) = 0$
- если $a(x,w) \to 0$, а $y = +1$, то штраф $L(a,y) \to +\infty$

ЛОГИСТИЧЕСКАЯ РЕГРЕССИЯ: ВЕРОЯТНОСТНАЯ ПОСТАНОВКА ЗАДАЧИ

ВЕРОЯТНОСТНАЯ ПОСТАНОВКА ЗАДАЧИ

Предположение: В каждой точке x пространства объектов задана вероятность $p(y = +1|x)$

Объекты с одинаковым признаковым описанием могут иметь разные значения целевой переменной.

ВЕРОЯТНОСТНАЯ ПОСТАНОВКА ЗАДАЧИ

Предположение: В каждой точке x пространства объектов задана вероятность $p(y = +1|x)$

Объекты с одинаковым признаковым описанием могут иметь разные значения целевой переменной.

Цель: построить алгоритм $b(x)$, в каждой точке x предсказывающий $p(y = +1|x)$.

ВЕРОЯТНОСТНАЯ ПОСТАНОВКА ЗАДАЧИ

Предположение: В каждой точке x пространства объектов задана вероятность $p(y = +1|x)$

Объекты с одинаковым признаковым описанием могут иметь разные значения целевой переменной.

Цель: построить алгоритм $b(x)$, в каждой точке x предсказывающий $p(y = +1|x)$.

Комментарий: *пока что мы будем решать задачу в общем виде, то есть у нас нет ограничений на вид алгоритма $b(x)$ и на вид функции потерь $L(y, b)$.*

ВЕРОЯТНОСТНАЯ ПОСТАНОВКА ЗАДАЧИ

- Пусть объект x встречается в выборке n раз с ответами $\{y_1, \dots, y_n\}$. Хотим, чтобы алгоритм выдавал вероятность положительного класса:

$$b_*(x) = \underset{b \in \mathbb{R}}{\operatorname{argmin}} \frac{1}{n} \sum_{i=1}^{n} L(y_i, b) \approx p(y = +1 | x)$$

ВЕРОЯТНОСТНАЯ ПОСТАНОВКА ЗАДАЧИ

- Пусть объект x встречается в выборке n раз с ответами $\{y_1, \dots, y_n\}$. Хотим, чтобы алгоритм выдавал вероятность положительного класса:

$$b_*(x) = \operatorname*{argmin}_{b \in \mathbb{R}} \frac{1}{n} \sum_{i=1}^{n} L(y_i, b) \approx p(y = +1|x)$$

По закону больших чисел при $n \to \infty$ получаем

$$b_*(x) = \operatorname*{argmin}_{b \in \mathbb{R}} E[L(y, b)|x]$$

ВЕРОЯТНОСТНАЯ ПОСТАНОВКА ЗАДАЧИ

- Пусть объект x встречается в выборке n раз с ответами $\{y_1, \dots, y_n\}$. Хотим, чтобы алгоритм выдавал вероятность положительного класса:

$$b_*(x) = \underset{b \in \mathbb{R}}{\operatorname{argmin}} \frac{1}{n} \sum_{i=1}^{n} L(y_i, b) \approx p(y = +1|x)$$

По закону больших чисел при $n \to \infty$ получаем

$$b_*(x) = \underset{b \in \mathbb{R}}{\operatorname{argmin}}\, E[L(y, b)|x]$$

Отсюда получаем ***условие на функцию потерь***:

$$\mathbf{argmin}\, \boldsymbol{E[L(y, b)|x] = p(y = +1|x)}$$

ФУНКЦИИ ПОТЕРЬ

Подходят:

- **Квадратичная**

$$L(y,z) = (y-z)^2$$

- **Логистическая (log-loss)**

$$L(y,z) = [y=+1]\cdot\log\big(b(x,w)\big) + [y=-1]\cdot\log(1-b(x,w))$$

Не подходят:

- **Модуль**

$$L(y,z) = |\mathrm{y}-\mathrm{z}|$$

ПРАВДОПОДОБИЕ И LOG-LOSS

- Вероятности, которые выдает алгоритм $b(x)$, должны согласовываться с выборкой
- Вероятность того, что в выборке встретится объект x с классом y:

$$b(x)^{[y=+1]} \cdot \left(1 - b(x)\right)^{[y=-1]}$$

ПРАВДОПОДОБИЕ И LOG-LOSS

- Вероятности, которые выдает алгоритм $b(x)$, должны согласовываться с выборкой
- Вероятность того, что в выборке встретится объект x с классом y:

$$b(x)^{[y=+1]} \cdot \big(1 - b(x)\big)^{[y=-1]}$$

Правдоподобие выборки:

$$(b, X) = \prod_{i=1}^{l} b(x_i)^{[y_i=+1]} \cdot \big(1 - b(x_i)\big)^{[y_i=-1]}$$

ФУНКЦИЯ ПОТЕРЬ ДЛЯ ОБУЧЕНИЯ

- Для нахождения оптимальных параметров алгоритма можно воспользоваться методом максимума правдоподобия (ММП):

$$(b, X) = \prod_{i=1}^{l} b(x_i)^{[y_i=+1]} \cdot \left(1 - b(x_i)\right)^{[y_i=-1]} \to \max_b$$

ФУНКЦИЯ ПОТЕРЬ ДЛЯ ОБУЧЕНИЯ

- Для нахождения оптимальных параметров алгоритма можно воспользоваться методом максимума правдоподобия (ММП):

$$(b, X) = \prod_{i=1}^{l} b(x_i)^{[y_i=+1]} \cdot \left(1 - b(x_i)\right)^{[y_i=-1]} \to \max_b$$

- Прологарифмируем правдоподобие и поставим перед ним минус, получим следующую эквивалентную задачу:

$$-\sum_{i=1}^{l} ([y_i = +1] \log b(x_i) + [y_i = -1] \log(1 - b(x_i))) \to \min_b$$

ФУНКЦИЯ ПОТЕРЬ ДЛЯ ОБУЧЕНИЯ

- Для нахождения оптимальных параметров алгоритма можно воспользоваться методом максимума правдоподобия (ММП):

$$(b, X) = \prod_{i=1}^{l} b(x_i)^{[y_i=+1]} \cdot \left(1 - b(x_i)\right)^{[y_i=-1]} \to \max_b$$

- Прологарифмируем правдоподобие и поставим перед ним минус, получим следующую эквивалентную задачу:

Это log-loss!

$$-\sum_{i=1}^{l} \left([y_i = +1] \log b(x_i) + [y_i = -1] \log(1 - b(x_i))\right) \to \min_b$$

ФУНКЦИЯ ПОТЕРЬ ДЛЯ ОБУЧЕНИЯ

- Для нахождения оптимальных параметров алгоритма можно воспользоваться методом максимума правдоподобия (ММП):

$$(b, X) = \prod_{i=1}^{l} b(x_i)^{[y_i=+1]} \cdot \big(1 - b(x_i)\big)^{[y_i=-1]} \to \max_b$$

- Прологарифмируем правдоподобие и поставим перед ним минус, получим следующую эквивалентную задачу:

Это log-loss!

$$-\sum_{i=1}^{l} \left([y_i = +1] \log b(x_i) + [y_i = -1] \log(1 - b(x_i))\right) \to \min_b$$

Вывод: *логистическая функция потерь корректно предсказывает вероятности.*

ВЫБОР АЛГОРИТМА $b(x)$

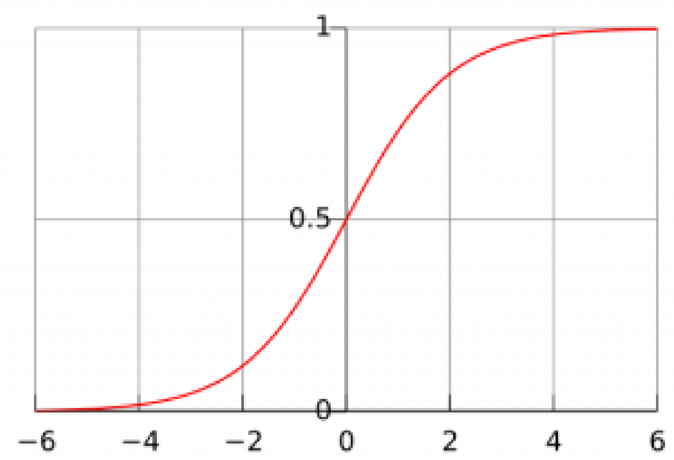
- Хотим, чтобы алгоритм $b(x)$ возвращал числа из отрезка $[0, 1]$.

ВЫБОР АЛГОРИТМА $b(x)$

- Хотим, чтобы алгоритм $b(x)$ возвращал числа из отрезка $[0, 1]$.
- Можно взять $b(x) = \sigma(w^T x)$, где σ – любая монотонно неубывающая функция с областью значений $[0, 1]$.

ВЫБОР АЛГОРИТМА $b(x)$

- Хотим, чтобы алгоритм $b(x)$ возвращал числа из отрезка $[0, 1]$.
- Можно взять $b(x) = \sigma(w^T x)$, где σ – любая монотонно неубывающая функция с областью значений $[0, 1]$.
- Возьмем ***сигмоиду***: $\sigma(z) = \frac{1}{1+e^{-z}}$



СМЫСЛ (w, x) В ЛОГИСТИЧЕСКОЙ РЕГРЕССИИ

- Логистическая регрессия в каждой точке x предсказывает вероятность того, что x принадлежит положительному классу $p(y = +1|x)$.

- То есть $p(y = +1|x) = \frac{1}{1+e^{-w^T x}}$. Отсюда можно выразить $(w, x) = w^T x$:

$$\boxed{(w, x) = w^T x = \log \frac{p(y = +1|x)}{p(y = -1|x)}}$$

СМЫСЛ (w, x) В ЛОГИСТИЧЕСКОЙ РЕГРЕССИИ

- Логистическая регрессия в каждой точке x предсказывает вероятность того, что x принадлежит положительному классу $p(y = +1|x)$.
- То есть $p(y = +1|x) = \frac{1}{1+e^{-w^T x}}$. Отсюда можно выразить $(w, x) = w^T x$:

$$\boxed{(w, x) = w^T x = \log \frac{p(y = +1|x)}{p(y = -1|x)}}$$

- Величина $\log \frac{p(y=+1|x)}{p(y=-1|x)}$ называется ***логарифм отношения шансов (log odds)***. Из формулы видно, что величина может принимать любое значение.

ЛОГАРИФМИЧЕСКАЯ ФУНКЦИЯ ПОТЕРЬ

Утверждение. Логарифмическая функция потерь может быть записана в виде

$$\boldsymbol{L(b, X) = \sum_{i=1}^{l} \log(1 + e^{-y_i(w,x)})}$$

Идея доказательства:

Подставляем явный вид сигмоиды в логарифмическую функцию потерь:

$$-\sum_{i=1}^{l} ([y_i = +1] \log \sigma(w^T x_i) + [y_i = -1] \log(1 - \sigma(w^T x_i))) \to \min_{w}$$

ПЕРСЕПТРОН РОЗЕНБЛАТТА

Перцептрон **–** это простейшая модель классификации, при этом являющаяся предшественником нейронных сетей.

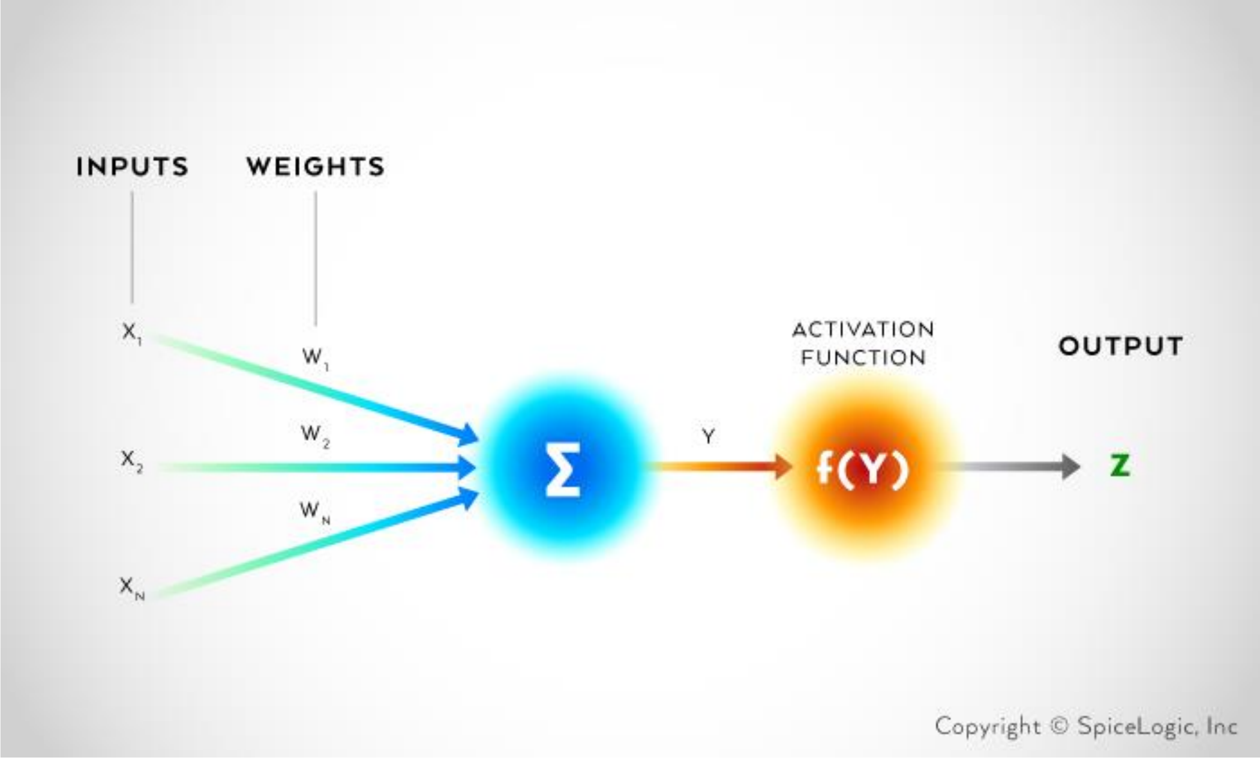
- Задача классификации с двумя классами $y_i \in \{0,1\}$
- Признаки объектов – бинарные: $x_i^j \in \{0,1\}$

Алгоритм: $\boldsymbol{a(x,w) = [w_1 x_1 + \cdots + w_n\, x_n > 0] = [(w,x) > 0]}$

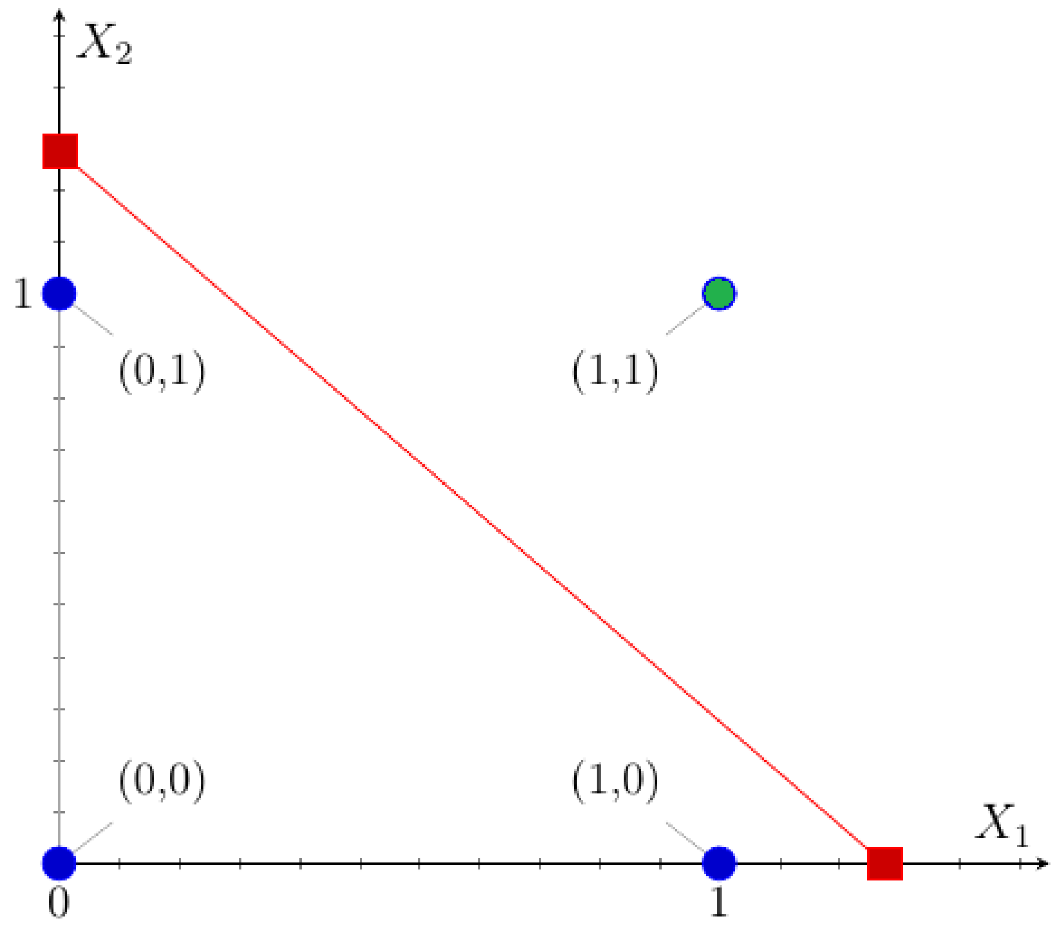
ПЕРСЕПТРОН РОЗЕНБЛАТТА

- Задача классификации с двумя классами $y_i \in \{0,1\}$
- Признаки объектов – бинарные: $x_i^j \in \{0,1\}$

Алгоритм: $\boldsymbol{a(x,w) = [w_1 x_1 + \cdots + w_n x_n > 0] = [(w,x) > 0]}$

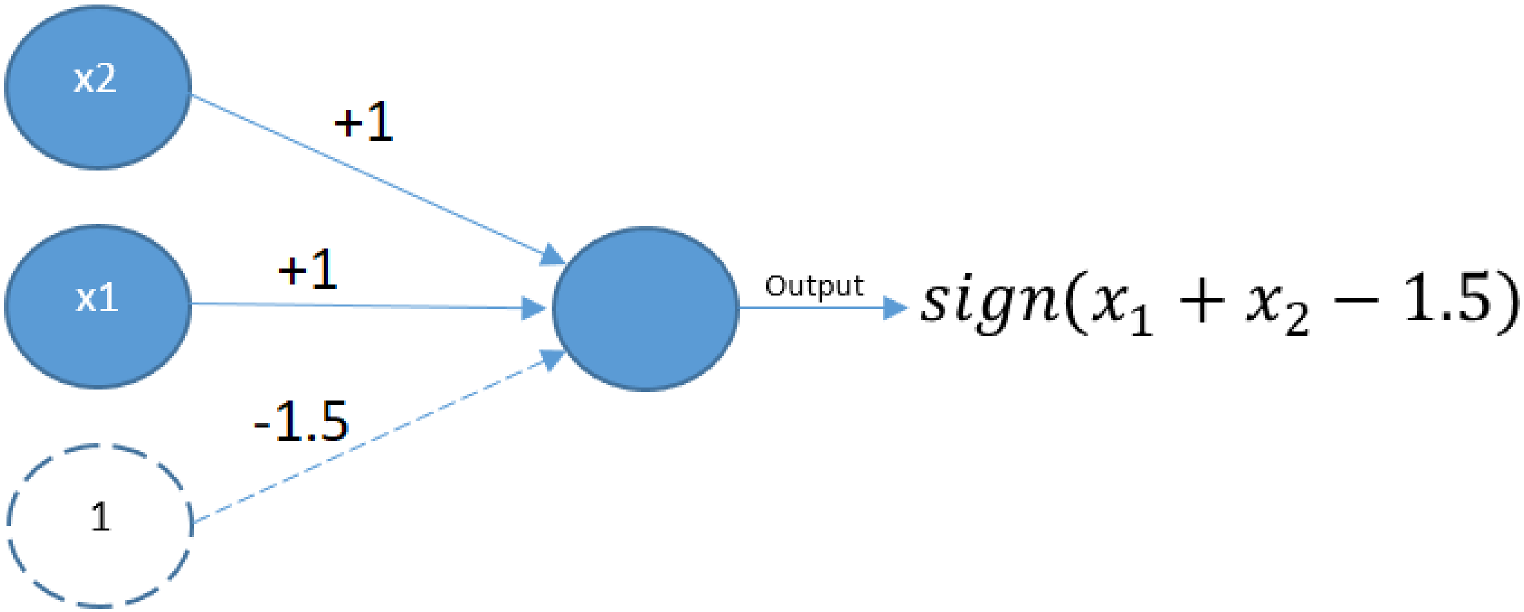


ПРИМЕР: РЕАЛИЗАЦИЯ ЛОГИЧЕСКОГО AND С ПОМОЩЬЮ ПЕРСЕПТРОНА

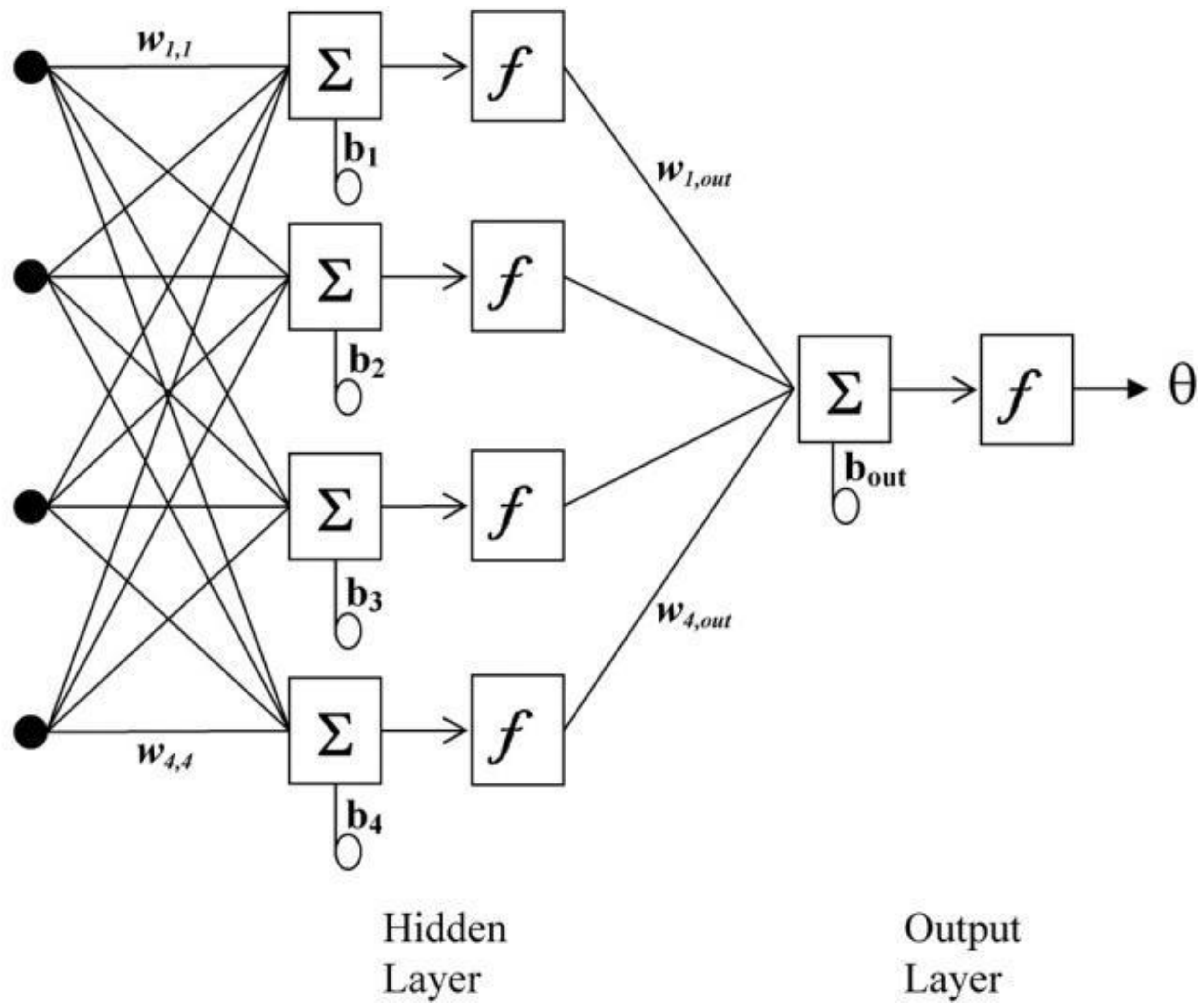


ПРИМЕР: РЕАЛИЗАЦИЯ ЛОГИЧЕСКОГО AND С ПОМОЩЬЮ ПЕРСЕПТРОНА

$$a(x, w) = sign(x_1 + x_2 - 1.5)$$



ПРИМЕР ДВУХСЛОЙНОГО ПЕРСЕПТРОНА



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

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 $p(y = +1|x)$.

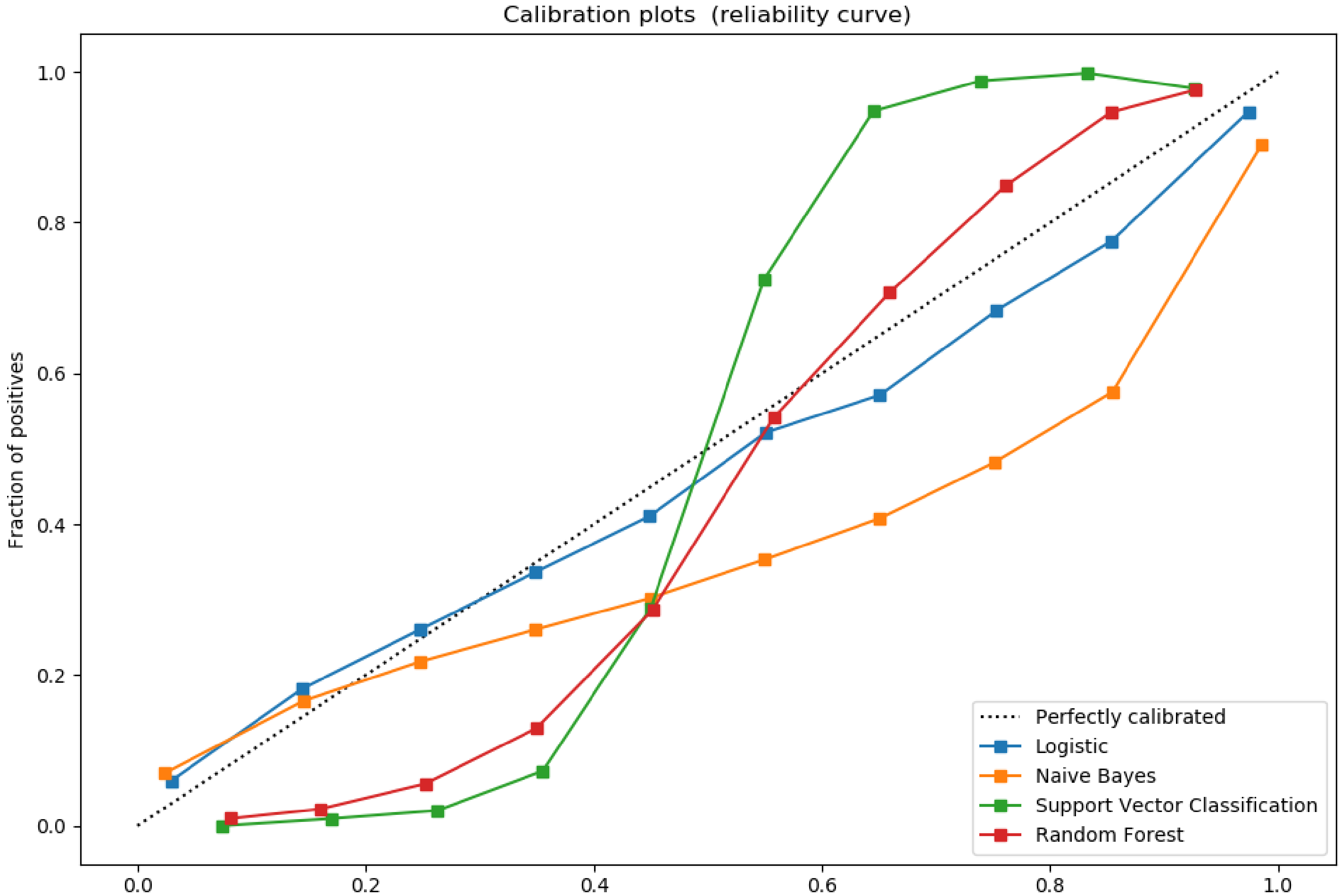
КАЛИБРОВКА ПЛАТТА

- 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 $a(x)$.

ПРИМЕР ИЗ SKLEARN



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*** $\boldsymbol{a(x)}$.

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 $\boldsymbol{a(x)}$***.

- $\pi(x; \alpha; \beta) = \sigma(\alpha \cdot \boldsymbol{a(x)} + \beta) = \frac{1}{1+e^{-(\alpha \cdot a(x) + \beta)}}$