KARPOV.COURSES >>>
КОНСПЕКТ

Конспект > 10 урок > Линейная классификация: оценка вероятности

>Оглавление

>Оглавление
>Линейная классификация
- Что если регрессионно решить задачу бинарной классификации?
- Может стоит разделить данные на пространстве признаков?

>Как строить разделяющую плоскость?
>Ликбез №1: метод верхней оценки
>Оценка вероятности
- Метод максимального правдоподобия
- Примерный порядок построения модели для бинарной классификации:

>Линейная классификация

Задача классификации заключается в присвоении объекту некоторого класса. В зависимости от количества классов различают бинарную классификацию и многоклассовую классификацию.

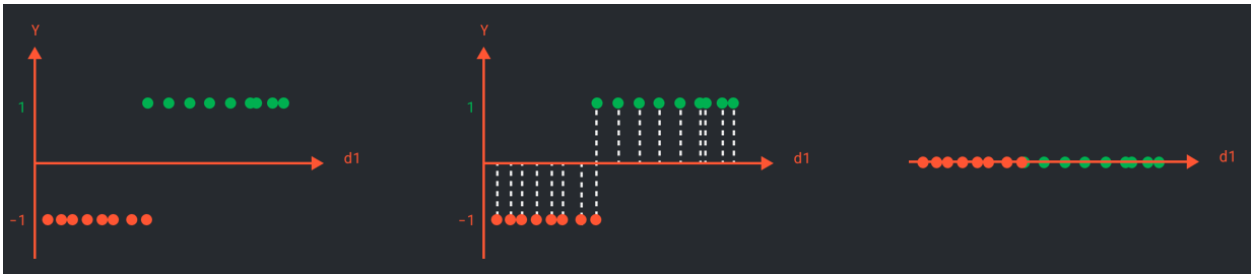
В первом случае, необходимо отнести объект к одному из **двух** классов. Примером таких задач может служить ***оценка состояния здоровья пациента*** (0 - здоров, 1 - болен), **оценка успеваемости** (0 - не поступил, 1 - поступил), **оценка важности письма** (0 - спам, 1 - важное), ***определить содержит ли фотография изображение человека или нет*** и так далее.

$$Y = \{1;\ -1\}$$

Когда решаем задачи регрессии мы пытаемся предсказать вещественный таргет, а когда мы классифицируем объекты мы пытаемся предсказать таргет из ограниченного множенства. Задача классификации является задачей обучения с учителем, что отличает ее от задачи кластеризации.

Для решения задач классификации существуют различные алгоритмы (линейные модели, деревья решений, градиентный бустинг и другие).

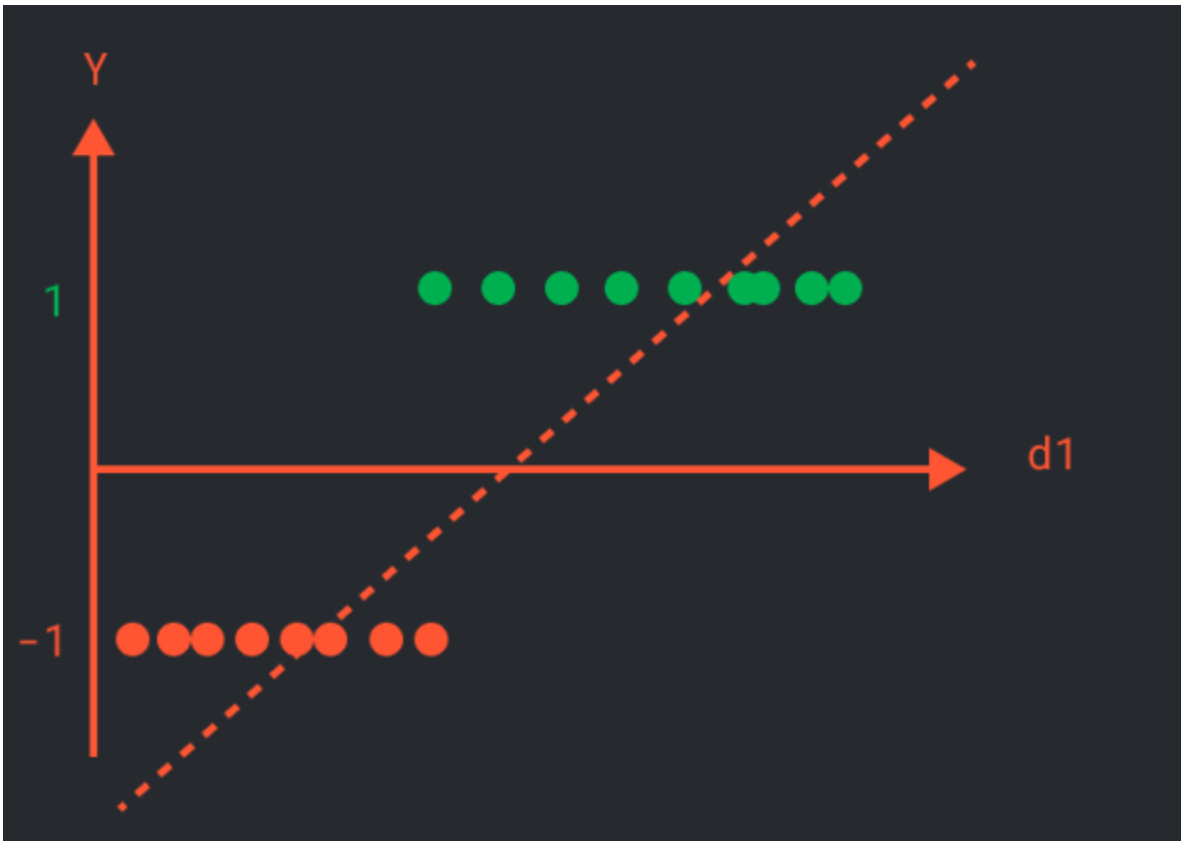
Будьте внимательны, в визуализации которую будем использовать при разборе теоретических выкладок по классификации мы уберём ось ответов OY. Переменная принимает только 2 значения: +1 и -1. Кажется, чтобы хранить эту информацию оси слишком много, можно просто раскрасить значения в красный и зеленый цвета



Слева двумерный график, справа одномерный график, но информация в них совершенно одинаковая.

Что если регрессионно решить задачу бинарной классификации?

Давайте построим модель линейной регрессии, признаки умножим на коэффициенты, просуммируем и попробуем предсказать. Так как мы решаем задачу бинарной классификации, то значения, которые могут принимать объекты равны +1 или -1, тогда графически получатся 2 горизонтальные полоски.



Интуитивно кажется, что модель не очень сильная. Если мы используем такую простую регрессионную модель, то мы:

1. Получаем неограниченные прогнозы (не только +1 или -1).
2. Получим не интерпретируемые результаты

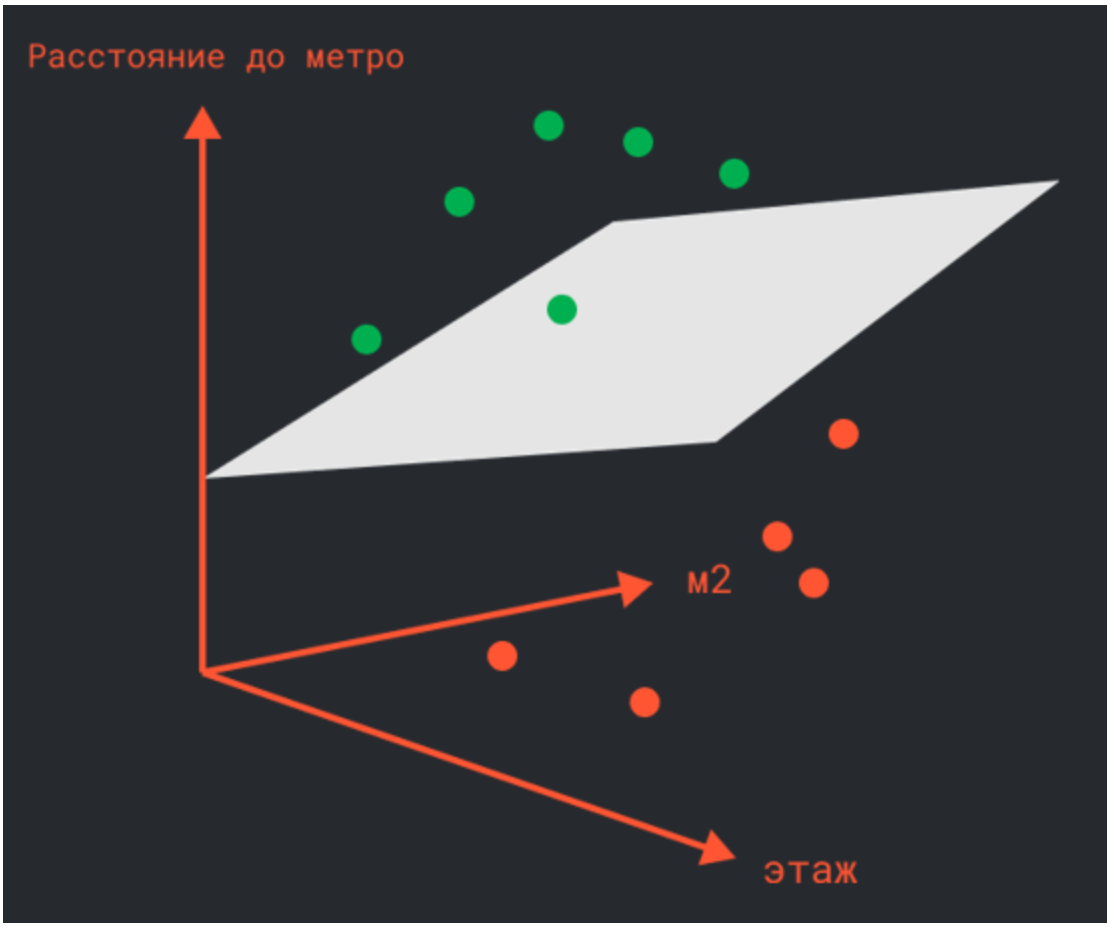
Резюмируя можно сказать, что решать задачу бинарной классификации старыми инструментами нельзя.

Может стоит разделить данные на пространстве признаков?

Давайте на оси в пространстве признаков попробуем найти какой-то ограничитель. Слева от которого будут данные одного класса, а справа - другого.

Правее $d*$ – класс +1, левее $d*$ – класс -1

В целом, данная модель куда лучше предыдущей линейной. Представим признаки в виде осей, тогда каждый из объектов будет точкой в трехмерном пространстве. Поделим трехмерное пространство гиперплоскостью.



Линейная модель классификации интерпретируется геометрически: Мы стараемся найти в пространстве признаков такую гиперплоскость, что она хорошо разделит

нам объекты на соответствующие классы! Ее еще называют разделяющей гиперплоскостью.

Как математически в пространствах различного размера строить гиперплоскости? Уравнение любой гиперплоскости:

$$b_1 \times d_1 + ... + b_n \times d_n + b_0 = 0$$

Где, b это точки в пространстве(объекты), а d коэффициенты.

Как сделать прогноз для объекта(положительные объекты или отрицательные, +1 или -1)? Как определить, слева или справа от гиперплоскости лежит нужный объект? Нужно взять объект, посмотреть какие у него признаки, подставить в левую часть уравнения гиперплоскости и посмотреть на знак. Если число отрицательное, то объект будет лежать с одной стороны плоскости, если число положительное, то с другой. Например, представим, у нас есть объект с двумя признаками: 30 и 4. Подставим в левую часть уравнения наши признаки, чтобы понять какой прогноз для данного объекта выдаст наша модель:

$$\alpha\left(30, 4\right) = sgn\left(1 \times 30 + 1.5 \times 4 - 30\right) = sgn\left(6\right) = +1$$

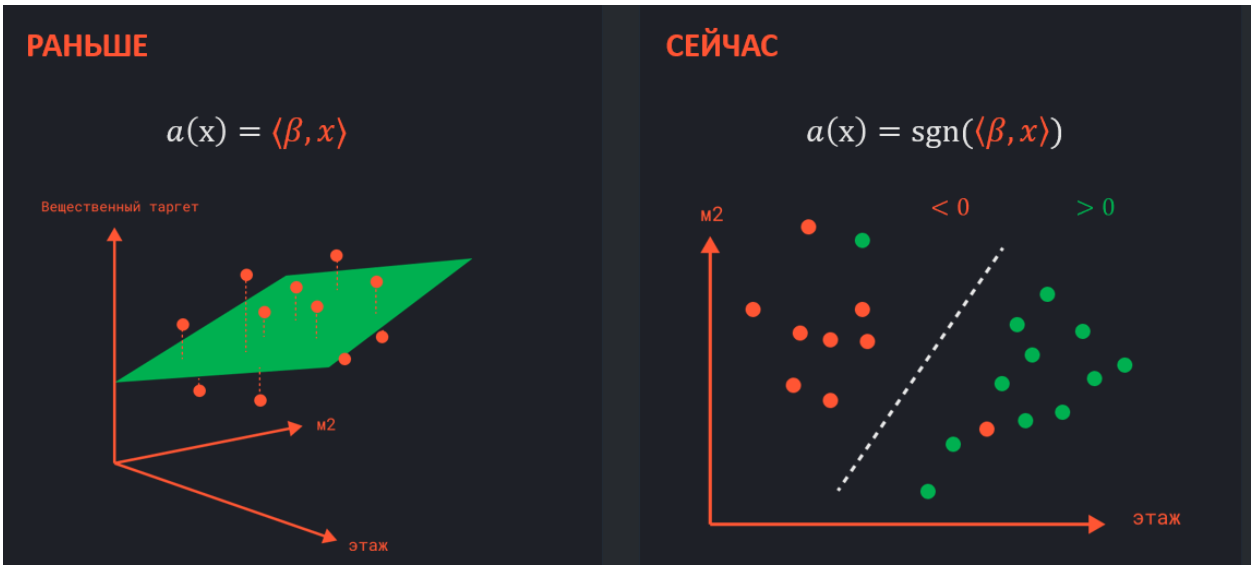
Пусть имеем какой-то набор параметров $\beta = (\beta_1, ..., \beta_n)$ и объект $x = (\delta_1, ..., \delta_n)$

Тогда $\langle \beta, \chi \rangle = \sum_n^i \beta_i * \delta_i = \beta_1 * \delta_i + ... + \beta_n * \delta_n$

Гиперплоскость разбивает пространство на два полупространства, в одном из которых скалярное произведение $\langle \beta, \chi \rangle$ будет принимать только положительные значения, в то время как в другом — только отрицательные. Таким образом, объекты попавшие в первое полупространство будут классифицированы моделью как объекты класса 1. В противном случае, объекту будет присвоена метка класса -1. В случае, если во время обучения классификатору удалось разделить объекты не совершив ни одной ошибки, мы получим линейно разделимую выборку.

Например, представим у нас есть вектор β=(1, 2, 3) и вектор x=(5, 4, 3). Попробуем найти между ними скалярное произведение:

$\langle \beta, \chi \rangle = 1 \cdot 5 + 2 \cdot 4 + 3 \cdot 3 = 22$



>Как строить разделяющую плоскость?

Как находить коэффициенты b для разделяющей плоскости? Нужно придумать метрику, например, долю правильных ответов(accuracy). Надо математически замерить качество на одном объекте через функцию-индикатор:

$$[\alpha(x_i) = y_i] = \begin{cases} 1 & \alpha(x_i) = y_i \\ 0 & \alpha(x_i) \neq y_i \end{cases}$$

Тогда, наша метрика будет выглядеть следующим образом:

Сумма по каждому из объектов функции-индикаторов того, что прогноз совпадает с истинным ответом, деленное на количество объектов в выборке. Для конкретного алгоритма $\alpha(x_i)$ данная формула будет показывать на какой доле объектов модель не ошиблась и правильно разметила на соответствующий им класс.

Удобнее будет считать долю ошибочно размеченных моделью объектов:

$$Q(\beta, \chi) = \frac{1}{n} \sum_{n}^{i} [\alpha(x_i) \neq y_i]$$

Чему равен индикатор $\alpha(x_i) \neq y_i$? По факту, этот индикатор описывает ситуацию, когда модель ошиблась. То есть, когда модель возвращает число, знак которого не совпадает со знаком числа в ответе.

$$Q(\beta, \chi) = \frac{1}{n} \sum_{n}^{i} [\alpha(x_i) \neq y_i] = \sum_{n}^{i} [sgn(\langle \beta, \chi \rangle) \neq y_i]$$

Полученный индикатор можно расписать чуть хитрее. Логика простая: знаки двух выражений не совпадают, когда их произведение отрицательно. Пусть наша модель для кого-то объекта прогнозирует +1(то есть принадлежность к положительному классу), это значит, что при подстановки данного объекта в уравнение гиперплоскости мы получим положительный знак(и если Y объекта на самом деле окажется отрицателен, то модель ошибётся). Простыми словами, если модель предсказывает -1, а ответ +1, тогда произведение $[y_i * \langle \beta, \chi \rangle]$ будет отрицательным.

$$[sgn(\langle \beta, \chi \rangle) \neq y_i] = [y_i * \langle \beta, \chi \rangle < 0] = [M_i < 0]$$

Данное произведение $[y_i * \langle \beta, \chi \rangle]$ называют отступом и обозначают в виде M_i.

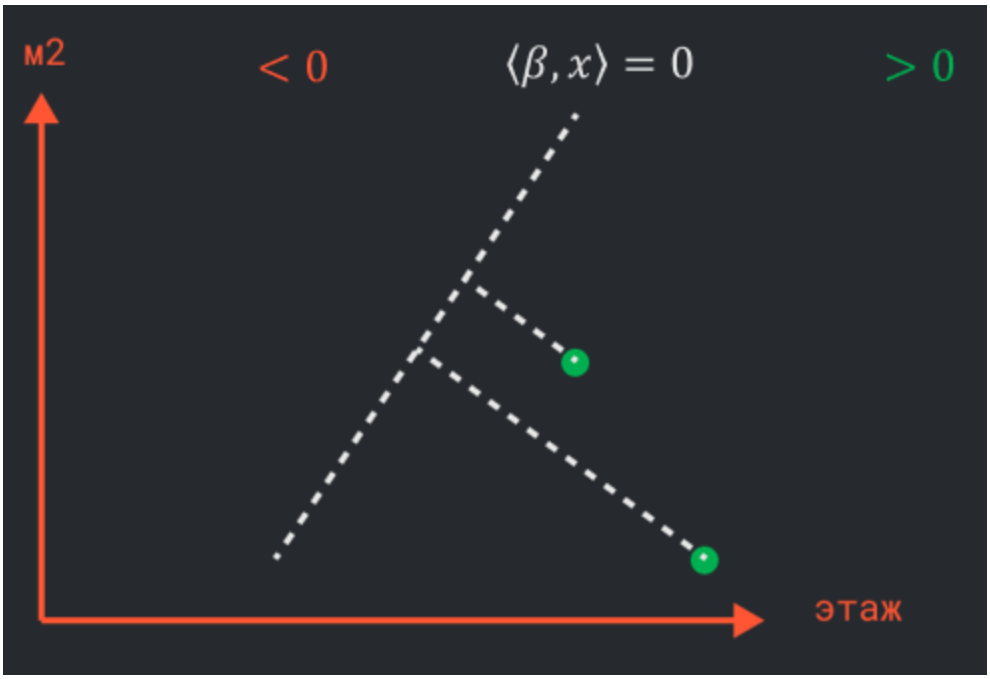
Отступ — это некоторая мера уверенности классификатора в своем ответе. Если $M_i > 0$, то объект отнесен к верному классу, в противном случае — произошла ошибка. Случай, когда классификатор часто ошибается с большим отступом, говорит о том, что была получена слабая модель. Можно минимизировать сумму индикаторов того, что $M_i < 0$ (если $M_i < 0$, значит модель ошиблась).

$$Q(\beta, \chi) = \sum_{n}^{i} [M_i < 0]$$

Ещё, у отступа можно интерпретировать не только знак, но и модуль. То есть, чему абсолютно количественно он равен. Чем больше модуль отступа - тем увереннее модель в прогнозе.

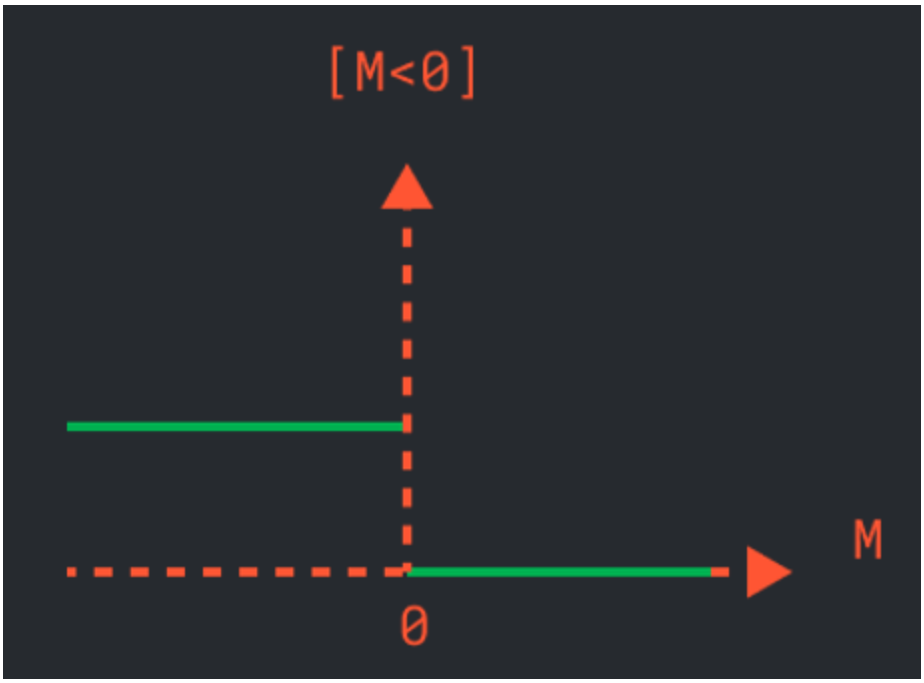
Представим, у нас есть какой-то объект, у него положительный класс, скалярное произведение равно +3 и модуль отступа равен +3. Есть другой объект, у него модуль отсупа равен 10. Модуль вырос в 3 раза, а значит данный объект в 3 раза дальше от разделяющей гиперплоскости чем первый объект. Ну а раз данный

объект находится далеко от гиперплоскости, значит модель уверена, что объект относится к тому классу, к которому она отнесла.



Достаточно логично: чем ближе наши объекты к гиперплоскости, тем меньше моедль в них уверена, ведь, кажется, ещё чуть-чуть их сдвинуть и они окажутся в другом классе. Чем больше модуль отступа — тем сильнее модель уверена в своем прогнозе.

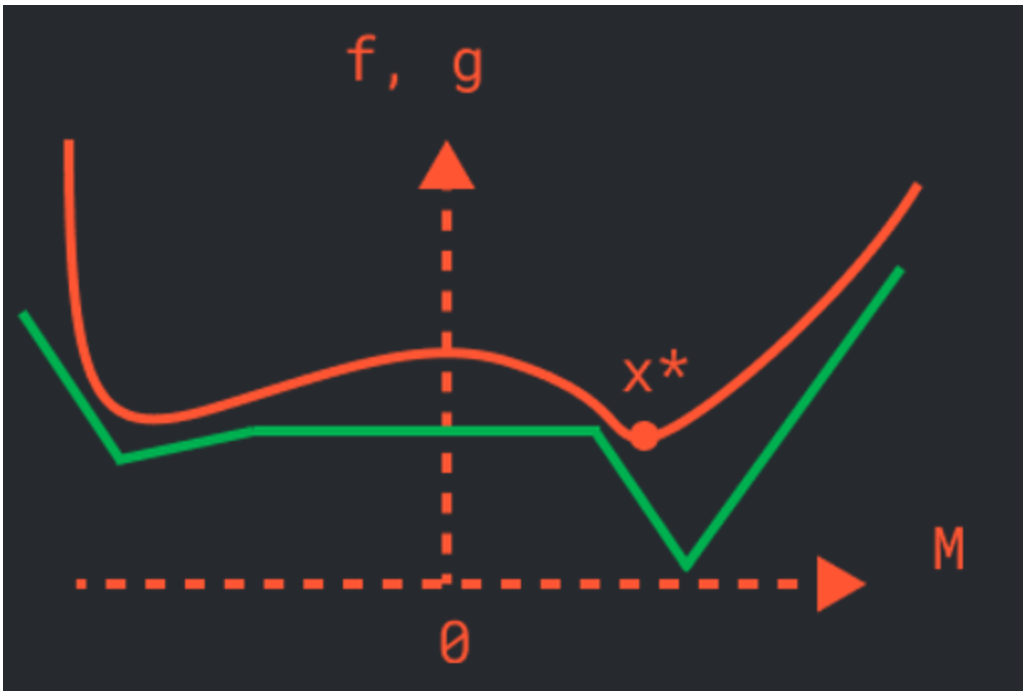
Давайте нарисуем функцию индикатора нарисуем в осях функции и отступа M. Данная функция возвращает следующие результаты: когда M>0, индикатор равен 0, а когда M<0, индикатор равен 1. Данная функция называется ступенчатой или функцией Хевисайда.



У ступенчатой функции производная всегда либо равна 0, либо не существует.

>Ликбез №1: метод верхней оценки

Пусть мы хотим найти минимум зеленой функции $f(x)$, но по какой-то причине, не можем ее дифференцировать.



Но зато известна красная функция $g(x)$, такая, что:

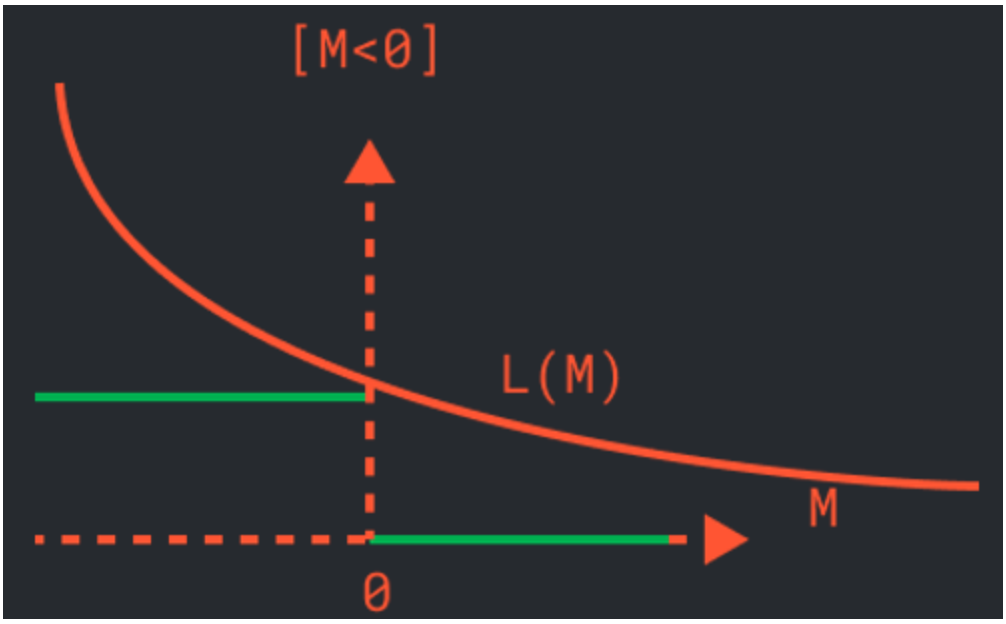
1. $g(x) \geq f(x)$

2. $g(x)$ дифференцируема

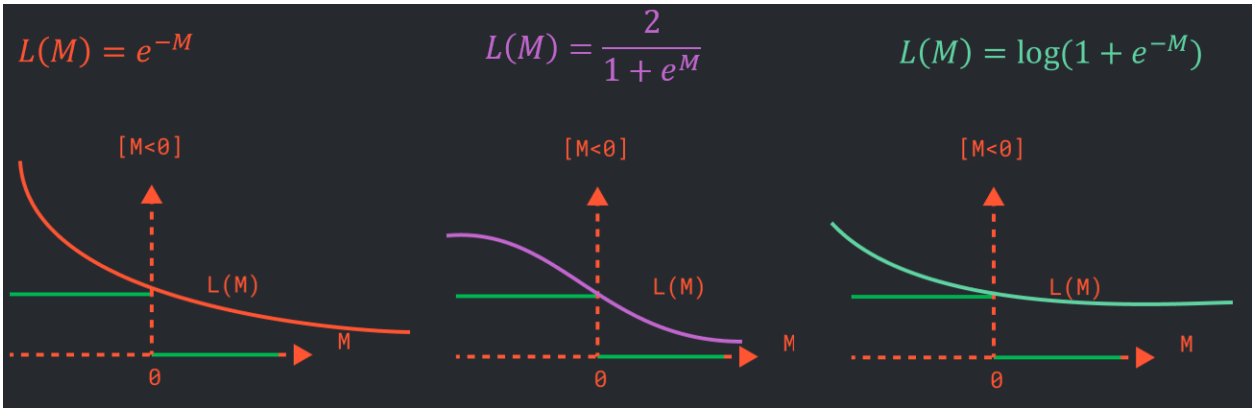
Тогда, найдя минимум $x*$ для $g(x)$, есть большая надежда на то, что $x*$ будет близок к $argmin\ f(x)$. Когда у вас есть функция, для которой вы хотите найти минимум, но по какой-то причине не можете её дифференциировать, можно придумать дополнительный слой, который всюду будет сверху. И у этой функции найти минимум.

Придумаем какую-нибудь гладкую дифференцируемую функцию $L(M)$, чтобы она для любых M находилась выше [M<0]. Тогда справедливо, $\sum_i^n [M_i < 0] \leq \sum_i^n L(M_i)$. Таким образом мы напрямую не минимизируем зеленую функцию, но мы сглаживаем близкой к ней(оранжевой) функцией и у нас появляется надежда, что у нужной нам функции похожий минимум.

$$Q^{new} = \sum_i^n L(M_i) = \sum_i^n (y_i * \langle \beta, \chi_i \rangle)$$



Несколько примеров:



Пример справа, логарифмическая функция потерь самый популярный способ сгладить отступы верхней оценкой.

>Оценка вероятности

Можем ли, учитывая уверенность нашей модели, давать оценки вероятностей? При решении задачи классификации хотелось бы строить модели не просто возвращающие метки класса(+1 или -1), а строить модели, которые еще и говорят с какой вероятностью тот или иной объект принадлежит к классу +1 или -1.

Метод максимального правдоподобия

Представим, мы хотим найти модель которая хорошо предсказывает вероятность принадлежности объекта к какому-то классу. У нас есть некоторая выборка.

$a(x) = P(y_i = +1 \mid x_i)$

Можно выписать следующую функцию, которая называется функцией правдоподобия:

$$Q = \prod_{n}^{i} a(x_i)^{y_i=+1} * (1 - a(x_i))^{y_i \neq +1}$$

Эта функция оценивает вероятность наблюдаемой выборки. Подставляя различные модели мы будем получать оценку вероятности получения нашей выборки (при данном законе распределения). Хотелось бы найти такой $a(x)$, при котором значения функции будут максимальными. То есть, мы хотим найти такой алгоритм, который моделирует вероятности при котором реализация той выборки которую мы наблюдаем наиболее вероятна по сравнению со всеми остальными реализациями.

Данную функцию необходимо максимизировать, т.к максимизируя функцию мы найдем такой алгоритм a(x), который оценивает вероятности наиболее реалистично и правдоподобно относительно данных которые наблюдали:

$$lnQ = \sum_{i}^{n}([y_i = +1] * ln(a(x_i)) + [y_i \neq +1] * ln(1 - (a(x_i))))$$

Один из примеров как задать функциональную форму алгоритма a(x), это сигмоидная функция:

$$a(x_i) = \frac{1}{1 + e^{-\langle \beta, \chi_i \rangle}} \in [0; 1]$$

Данная функция позволит нам для каждого b (т.е для каждой гиперплоскости и объекта) получать какое-то значение от 0 до 1.

Итого, чтобы получить оценку вероятности, нужно положить $\langle \beta, x\_i \rangle$ в какую-либо монотонную неубывающую функцию, принимающую значения от 0 до 1.

Например, в сигмоиду! При этом, оценки вероятности получатся корректными с точки зрения вероятностной парадигмы, если использовать логистическую функцию потерь.

Примерный порядок построения модели для бинарной классификации:

Мы хотим чтобы модель ошибалась как можно меньше, т.е минимизировать сумму неправильных ответов. Но такая задача негладкая, её нельзя дифференциировать и нельзя применять методы градиентного спуска. Поэтому, метрику $\sum_n^i [M_i < 0] \leq \sum_n^i L(M_i)$ мы обычно сглаживаем верхней оценкой и уже её минимизируем. Например, в качестве верхней оценки индикатора от отступа можно взять логарифм, т.е $\sum_n^i L(M_i) = \sum_n^i log(1 + e^{-M})$. Такую верхнюю оценку называют логистической функцией потерь, а модель, которая получается, логистической регрессией.

Далее используем градиентный спуск, получаем разделяющую гиперплоскость. Теперь, чтобы понять к отрицательному или положительному классу относится тот или иной объект, необходимо подставить объекты в левую часть выражения $a(x) = sgn(b_1 \times d_1 + ... + b_n \times d_n + b_0)$ и замерить знак полученного числа. Если хотим получить ещё и вероятность принадлежности к классу можно воспользоваться сигмоидной функцией $P(y_i = +1 \mid x_i) = \frac{1}{1+e^{-\langle \beta, x_i \rangle}}$