**Пример AdaBoost на бинарной классификации.**

Задача: разделить +1 и -1 простыми порогами по x.

Данные. Возьмём 5 объектов:

|  |  |  |
| --- | --- | --- |
| № | x | y |
| 1 | 1 | +1 |
| 2 | 2 | +1 |
| 3 | 3 | -1 |
| 4 | 4 | -1 |
| 5 | 5 | -1 |

Будем использовать очень простые слабые модели (вида: «если x < t, то класс = +1, иначе −1»).

Шаг 0. Инициализация весов

У всех объектов одинаковый вес:

Итерация 1

Пусть первая модель выбрала порог t=3.5:

Её предсказания:

|  |  |  |  |
| --- | --- | --- | --- |
| x | y |  | Верно? |
| 1 | +1 | +1 | Да |
| 2 | +1 | +1 | Да |
| 3 | -1 | +1 | Нет |
| 4 | -1 | -1 | Да |
| 5 | -1 | -1 | да |

Ошибка на 3 объекте, вес ошибки = 0.2.

Вес модели:

Теперь обновляем веса объектов. Формула:

* Для правильно классифицированных объектов:
* Для ошибочных:

Значит:

* Ошибочный объект (x=3): вес = 0.2⋅2=0.4.
* Остальные: вес = 0.2⋅0.5=0.1.

Сумма весов = 0.4+4⋅0.1=0.8.  
Нормируем:

* Ошибочный объект (x=3): 0.4/0.8=0.5.
* Остальные: 0.1/0.8=0.125.

Итерация 2

Теперь обучаем вторую модель на новых весах.  
Сильнее всего весит объект x=3, y=−1.

Предположим, вторая модель выбирает порог t=1.5:

Предсказания:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| x | y |  | Вес объекта | Верно? |
| 1 | +1 | +1 | 0.125 | Да |
| 2 | +1 | -1 | 0.5 | Нет |
| 3 | -1 | -1 | 0.125 | да |
| 4 | -1 | -1 | 0.125 | Да |
| 5 | -1 | -1 | 0.125 | да |

Ошибочный объект - x=2 с весом 0.125.  
Значит:

Вес модели:

Ансамбль после 2 итераций(если мы возьмём их «взвешенное большинство голосов», то получим )

Предсказание:

Если 0, значит «большинство» (с учётом весов) голосует за класс +1.

Если 0, значит ансамбль считает, что объект принадлежит классу -1.

**Пример градиентного бустинга на простой регрессионной задаче с квадратичной потерей**

Задача

Мы хотим построить модель (ансамбль), которая предсказывает y по x.  
Используем градиентный бустинг с квадратичной ошибкой (MSE).

Условия задачи

Пусть данные (N=4):

x=[1,2,3,4], y=[2,3,2,5].

Берём квадратичную функцию потерь:

Начальное приближение — константа, среднее по y:

Для квадратичной потери отрицательный градиент (антиградиент) равен:

Он пропорционален простым остаткам . Для наглядности мы будем обучать базовые модели на целевых значениях ​.

Выберем скорость обучения ν=0.5.

Итерация 0

Значит, модель сначала предсказывает всегда «3» для любого x.

Остатки:

MSE начальная:

Итерация 1

Шаг 1. Разобьём точки по порогу .

* Группа A (x≤2): остатки [−1,0] → среднее .
* Группа B (x>2): остатки [−1,2] → среднее

Значит, новая модель h₁ будет говорить:

* если x ≤ 2, добавляй -0.5
* если x > 2, добавляй +0.5

Базовая модель выдаёт предсказание остатков:

= [−0.5,−0.5,0.5,0.5].

Шаг 2. Обновляем ансамбль:

Шаг 3. Новые остатки:

MSE после 1 итерации:

Итог: ошибка уменьшилась с 1.50 до 1.3125.

Итерация 2

Шаг 1. Берём простое разбиение x≤3 и x>3.

* Для x≤3 (индексы 1..3) остатки =[−0.75,0.25,−1.25] → среднее
* Для x>3 (индекс 4) остаток 1.75 → среднее .

Базовая модель (аппроксимация антиградиента):

Шаг 2. Обновление:

Вычисляем по компонентам:

Итого:

Шаг 3. Остатки после 2 итераций:

MSE после 2 итераций:

Итог: MSE уменьшилась дальше — с 1.3125 (после 1-й) до ≈0.547 (после 2-й).