Машинное обучение

Лекция 13: Наивный байесовский классификатор

Докладчик: Артем Лебедевич

Что рассмотрим сегодня?

- Постановка задачи
- Вывод формулы для модели
- Формулы для вычисления вероятностей
- Визуализация, плюсы и минусы модели

Постановка задачи

Постановка задачи

Пусть задана выборка $\mathbb{D} = (X|y)_{i=1}^{n}$, где $X \subseteq \mathbb{X} = \mathbb{R}^{n \times m}$, $y \subseteq \mathbb{Y} = \{C_1, \ldots, C_k\}$, то есть

$$\mathbb{D} = \begin{pmatrix} x_{11} & \ldots & x_{1m} & y_1 \\ \vdots & \ddots & \vdots & \vdots \\ x_{n1} & \ldots & x_{nm} & y_n \end{pmatrix}.$$

Пусть нам дан некоторый объект $(x_1, \ldots, x_m|y) \in (\mathbb{X}|\mathbb{Y})$ с неизвестной меткой класса y.

Перед нами стоит задача найти метку класса для нашего объекта $y = C_k$.

Постановка задачи

Из теории вероятностей известна формула Байеса, которую мы адаптируем под рассматриваемую задачу

$$P(y = C_k|x_1, \ldots, x_m) = \frac{P(y = C_k) \cdot P(x_1, \ldots, x_m|y = C_k)}{P(x_1, \ldots, x_m)}. \quad (1)$$

- Распределение $P(y = C_k|x_1, \ldots, x_m)$ метки класса $y = C_k$, при условии известных данных $x_1, \ldots, x_m$, полученных после эксперимента, называется **апостериорным**.
- Распределение $P(y = C_k)$, которое выражает предположение о вероятности метки класса без учета экспериментальных данных, называется **априорным**.
- Распределение $P(x_1, \ldots, x_m|y = C_k)$ экспериментальных данных при условии некоторой определенной метки класса $y = C_k$ называется **правдоподобием** (то есть насколько правдоподобно при заданной метке класса получить из эксперимента именно такой набор признаков).

Таким образом, теорему Байеса можно трактовать так: нормализованное произведение априорного распределения на функцию правдоподобия является условным распределением неопределённой величины согласно учтённым данным.

Вывод формулы для модели

Условные вероятности

Вспомним формулу условной вероятности

$$P(A|B) = \frac{P(AB)}{P(B)} \text{ или } P(AB) = P(A|B) \cdot P(B).$$

Для упрощения сперва преобразуем числитель в соответствии с этой формулой

$$P(y = C_k) \cdot P(x_1, \ldots, x_m | y = C_k) = P(x_1, \ldots, x_m, y = C_k).$$

А теперь проделаем цепочку разбиений

$$\begin{gathered} P(x_1, \ldots, x_m, y = C_k) = \\ = P(x_1 | x_2, \ldots, x_m, y = C_k) \cdot P(x_2, \ldots, x_m, y = C_k) = \\ = P(x_1 | x_2, \ldots, x_m, y = C_k) \cdot P(x_2 | x_3, \ldots, x_m, y = C_k) \cdot P(x_3, \ldots, x_m, y = C_k) = \\ = \ldots = \\ = P(x_1 | x_2, \ldots, x_m, y = C_k) \cdot \ldots \cdot P(x_{n-1} | x_n, y = C_k) \cdot p(x_n | y = C_k) \cdot P(y = C_k). \end{gathered}$$

Наивное предположение

Наивное предположение: пусть в реальном процессе, из которого пришли данные, все признаки для каждого объекта независимы в совокупности. То есть вероятность $P(x_i|x_{i+1},\ldots,x_m,y=C_k)$ получить в эксперименте признак x_i зависит только от метки класса $y=C_k$:

$$P(x_i|x_{i+1},\ldots,x_m,y=C_k)=P(x_i|y=C_k).$$

Формула точечной оценки

А тогда числитель всей дроби в формуле (1) представим в виде

$$P(x_1, \ldots, x_m, y = C_k) = P(y = C_k) \cdot \prod_{i=1}^{m} P(x_i | y = C_k).$$

При этом здесь уже явно видно, что $\prod_{i=1}^{m} P(x_i | y = C_k)$ – это функция правдоподобия. Таким образом, мы получаем формулу наивного Байеса в виде

$$P(y = C_k | x_1, \ldots, x_m) = \frac{P(y = C_k) \cdot \prod_{i=1}^{m} P(x_i | y = C_k)}{P(x_1, \ldots, x_m)}. \qquad (2)$$

В качестве точечной оценки C_k логично выбирать самое вероятное значение $y = C_k | x_1, \ldots, x_m$, а это значит

$$\hat{C}_k = \arg\max_{C_k} P(y = C_k | x_1, \ldots, x_m) = \arg\max_{C_k} \frac{P(y = C_k) \cdot \prod_{i=1}^{m} P(x_i | y = C_k)}{P(x_1, \ldots, x_m)}.$$

Оценка апостериорного максимума

$$\hat{C}_k = \arg\max_{C_k} P(y = C_k | x_1, \ldots, x_m) = \arg\max_{C_k} \frac{P(y = C_k) \cdot \prod_{i=1}^{m} P(x_i | y = C_k)}{P(x_1, \ldots, x_m)}.$$

В силу того, что вероятность $P(x_1, \ldots, x_m)$ не зависит от C_k, а следовательно, является постоянной величиной, она не влияет на расположение точки, в которой функция принимает максимальное значение. Таким образом, можно принять $P(x_1, \ldots, x_m) = 1$ и тогда

$$\hat{C}_k = \arg\max_{C_k} P(y = C_k) \cdot \prod_{i=1}^{m} P(x_i | y = C_k).$$

Полученное число называется **оценкой апостериорного максимума**.

Логарифмирование

Вспомним метод максимального правдоподобия. Переходя от исходной функции к логарифмам, мы можем заменить все произведения на суммирование, при этом аргумент максимума не поменяет своего расположения в пространстве. То есть мы можем упростить эту формулу путем логарифмирования

$$\hat{C}_k = \arg\max_{C_k} \left(\log P(y = C_k) + \sum_{i=1}^{m} \log P(x_i | y = C_k) \right).$$

Таким образом, чтобы найти оценку метки класса $\hat{C}_k$ нужно определить априорную вероятность $P(y = C_k)$ и закон распределения вероятностей признаков $P(x_i | y = C_k)$.

Формулы для вычисления вероятностей

Апостериорная вероятность

С априорной вероятностью все просто. Для ее вычисления можно использовать один из двух вариантов:

- в предположении, что в реальном процессе получение метки определенного класса равновероятно, вероятность того, что метка класса равна C_k

$$P(y = C_k) = \frac{1}{\sum_{j=1}^{k} C_j},$$

  то есть $1/$количество классов;

- апостериорно из наблюдаемых значений

$$P(y = C_k) = \frac{\text{количество объектов класса } k}{\text{количество всех объектов}}.$$

Гауссовский классификатор

Для распределения признаков чуть больше вариантов:

- **Гауссовский наивный байесовский классификатор** – вариант для работы с непрерывными признаками, которые имеют нормальное (гауссовское) распределение. Вероятность признака при заданном классе вычисляется по формуле:

$$P(x_i|y = C_k) = \frac{1}{\sigma_{C_k}\sqrt{2\pi}} \exp\left(-\frac{(x_i - \mu_{C_k})^2}{2\sigma_{C_k}^2}\right),$$

где μ_{C_k} и σ_{C_k} – это среднее и стандартное отклонения признака в классе y. Эти параметры оцениваются с помощью метода максимального правдоподобия по обучающим данным.

Мультиномиальный классификатор

- **Мультиномиальный наивный байесовский классификатор** – вариант для работы с дискретными признаками, которые имеют мультиномиальное распределение. Такие признаки часто встречаются в задачах классификации текстов, где они представляют собой количество вхождений в тексте. Вероятность признака при заданном классе вычисляется по формуле:

$$P(x_i|y = C_k) = \frac{N_{i:C_k} + \alpha}{N_{C_k} + \alpha n},$$

где $N_{i:C_k}$ – это количество раз, когда признак i встречается в классе C_k; N_{C_k} — общее количество всех признаков в классе C_k; n — количество различных признаков; а α — сглаживающий параметр, предотвращающий возникновение нулевых вероятностей.

Бернуллиевский классификатор

- **Бернуллиевский наивный байесовский классификатор** — ещё один вариант для работы с дискретными признаками, но которые имеют бернуллиевское распределение. В данном случае признаки представляют собой бинарные индикаторы наличия или отсутствия определённых свойств в объекте. Например, в задаче классификации текстов это может быть наличие или отсутствие определённых слов в тексте. Вероятность признака при заданном классе вычисляется по формуле:

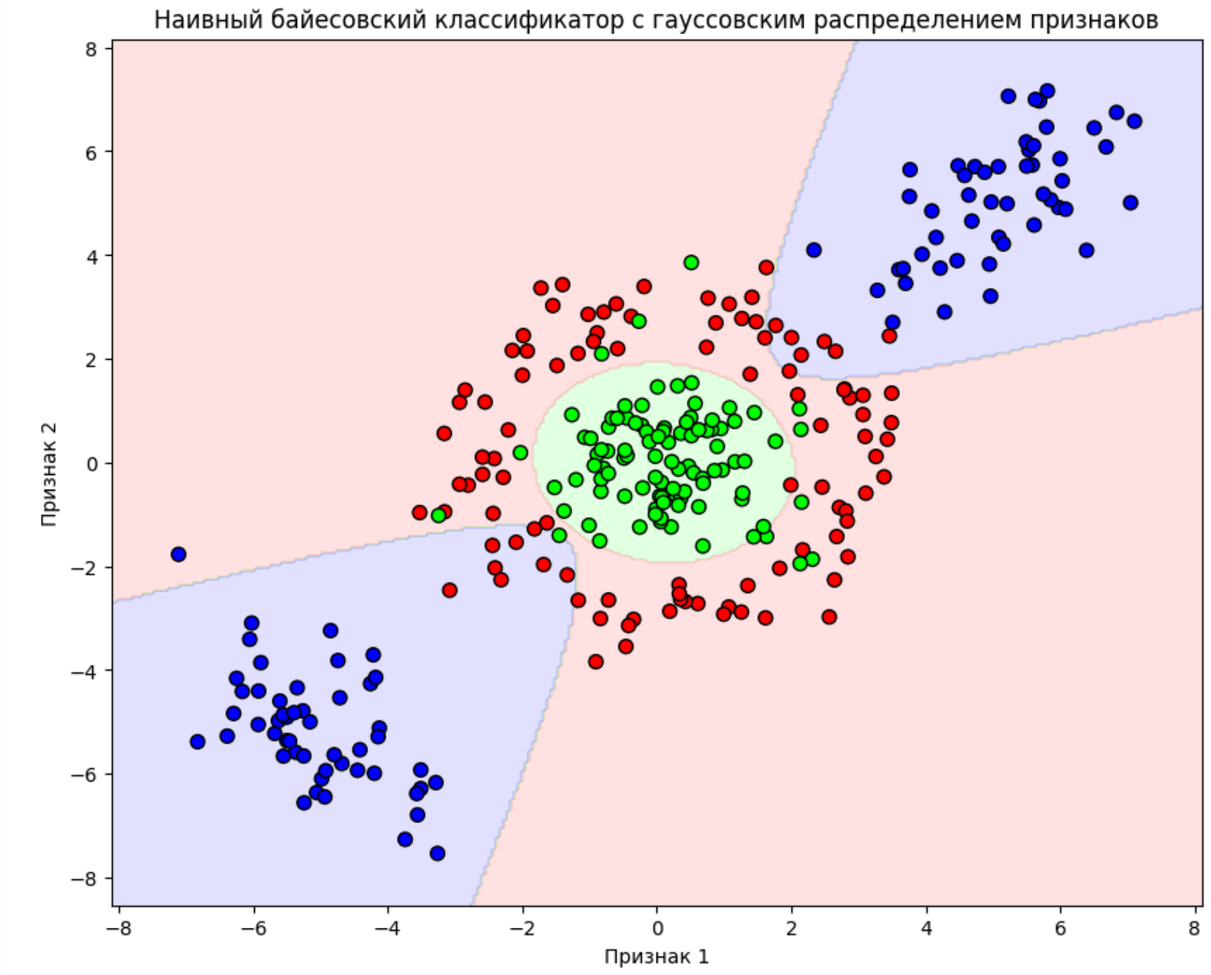
$$P(x_i|y = C_k) = P(x_i = 1|y = C_k) \cdot x_i + (1 - P(x_i = 1|y = C_k))(1 - x_i),$$

где $P(x_i = 1|y = C_k)$ — это вероятность того, что признак i принимает значение 1 (истина) при условии, что объект принадлежит классу $y = C_k$; x_i — значение признака i (0 или 1).

Визуализация
Плюсы и минусы

Визуализация

На графике можно увидеть, несмотря на свою простоту, способна строить довольно сложные области.



Плюсы и минусы NB

Плюсы:

- простота в реализации и интерпретации;
- практически не нуждается в обучении, а требует лишь вычисления вероятностей, что делает модель полезной для больших наборов данных;
- высокая скорость работы и точность прогнозов во многих ситуациях;
- не требует масштабирования признаков;
- имеет относительно хорошую устойчивость к шуму и выбросам, поскольку основан на вероятностных распределениях и наивном предположении о независимости признаков.

Плюсы и минусы NB

Минусы:

- в случае нарушения предположения о независимости признаков, точность прогнозов может значительно снизиться, особенно если между признаками есть корреляции;
- алгоритм не учитывает порядок и вес признаков, что может быть важным в некоторых задачах;
- если набор данных несбалансирован, то вероятность будет больше у классов с большим числом объектов.