Домашнее задание (ИАД-16)

Выдано: 24 февраля 2016 г.
Срок сдачи: 9 марта 2016 г.

Содержание

1 Ликбез

1.1 Байесовский классификатор

Пусть $\mathbb{X}$ — множество значений, которые могут принимать объекты, а $\mathbb{Y}$ — множество классов объектов.

Предположим, что все объекты обучающей выборке приходят из некоторого вероятностного распределения на $\mathbb{X} \times \mathbb{Y}$, где $\mathsf{P}(x, y)$ — это вероятность получить объект x с ответом y.

Пусть в обучающей выборке l объектов, каждый из которых является вектором из d признаков. Далее значение j-ого признака для k-ого объекта будем обозначать за x_k^j. Совокупность всех объектов обучающей выборки обозначим за X, которую можно понимать как матрицу «объекты-признаки» размера $l \times d$.

Теперь пусть x — объект из тестовой выборки, класс которого нужно предсказать. Разумно выбирать такой класс, к которому x принадлежит с бо́льшей вероятностью. Математически это записывается как

$$a(x) = \arg\max_{y \in \mathbb{Y}} \mathsf{P}(y|x).$$[1]

Такой классификатор a называется *байесовским классификатором.*

Применим формулу Байеса

$$\mathsf{P}(y|x) = \frac{\mathsf{P}(x|y)\,\mathsf{P}(y)}{\mathsf{P}(x)}$$

и заметим, что в её стоит не зависящая от y величина, а значит байесовский классификатор можно переписать в виде

$$\boxed{a(x) = \arg\max_{y \in \mathbb{Y}} \mathsf{P}(x|y)\,\mathsf{P}(y)} \tag{1}$$

Итого для обучения байесовского классификатора нужно определить $\mathsf{P}(y|x)$ и $\mathsf{P}(y)$.

$\mathsf{P}(y)$ выбирают, руководствуясь знаниями о том, как распределены классы. На семинаре Надя приводила пример, что в задаче определения пола человека разумно взять $\mathsf{P}(y = \text{мужчина}) = 0.45$, руководствуясь статистикой. Если никакой информации о классах нет, разумно считать $P(y)$ равномерным распределением.

С $\mathsf{P}(x|y)$ всё гораздо сложнее. Один из подходов, рассказанный вам на лекции — переход к эмпирической плотности распределения

$$\mathsf{P}(x|y) = \frac{1}{l} \sum_{i=1}^{l} [x = x_i][y = y_i],$$

где квадратные скобки обозначают так называемую нотацию Айверсона.

Однако маленьких выборках этот способ работает плохо.

Другой способ — сделать предположение, что $\mathsf{P}(x|y)$ имеет распределение из некоторого параметрического семейства (например, многомерное нормальное распределение), а затем оценить параметры. Но не всегда такие предположения имеют физический смысл, и вообще мы знаем довольно мало многомерных распределений.

[1] $\arg\max_{y \in \mathbb{Y}} \mathsf{P}(y|x)$ обозначает, что результат — это такой $y \in \mathbb{Y}$, при котором достатигается максимум значения $\mathsf{P}(y|x)$.

1.2 Наивный байесовский классификатор

Однако иногда можно сделать предположение, что все признаки независимы в совокупности.

Например, пусть мы решаем всё ту же задачу определения пола человека, и для признаком каждого объекта (человека) является рост и длина волос. На самом деле зависимость между этими признаками есть, потому что рост мужчин больше, а волосы короче. Но можно предположить, что зависимость признаков не очень велика, и ей можно принебречь.

В предположении независимости признаков совместная вероятность распадается в произведение уже одномерных

$$\mathsf{P}(x|y) = \prod_{j=1}^{d} \mathsf{P}(x^j|y). \tag{2}$$

Байесовский классификатор с предположением независимости признаков называется *наивным байесовским классификатором*. Его формулу можно получить подстановкой формулы 2 в формулу 1, выглядит она как

$$\boxed{a(x) = \arg\max_{y \in \mathbb{Y}} \prod_{j=1}^{d} \mathsf{P}(x^j|y)\,\mathsf{P}(y)} \tag{3}$$

При рассчётах этой формулы на компьютере **нужно** переходить к максимизации логарифма произведения, потому что произведение большого числа маленьких чисел вычисляется с большой погрешностью.

1.3 Пуассоновский наивный байесовский классификатор для анализа текстов

Представим, что мы решаем задачу классификации текстов на k классов: $c_1, c_2, \ldots, c_k$.

Пусть в текстах обучающей выборки d различных слов. Для каждого текста посчитаем, сколько раз каждое слово встречается в нём и объявим одним из признаков. Здесь мы неявно предполагаем, что справедлива гипотеза «мешка слов», то есть класс текста не зависит от порядка слов в нём. Итого каждому объекту (тексту) будет соответствовать вектор признаков длины d.

Распределение каждого признака можно приблизить распределением Пуассона, которое моделирует число событий, произошедших за фиксированное время, при условии, что данные события происходят с некоторой фиксированной средней интенсивностью и независимо друг от друга. Эта фиксированная средняя интенсивность задаётся параметром λ.

Мы считаем, что **разные классы отличаются только параметром**. Обозначим параметр распределения Пуассона j-ого признака объектов класса c_k за λ_k^j. [2] В таком случае *обучение* классификатора — это нахождение всех λ_k^j.

Если обозначить за X_k множество объектов класса c_k в обучающей выборке, а за $|X_k|$ — их количество, то с помощью принципа максимума правдоподобия можно показать, что

$$\boxed{\lambda_k^j = \frac{1}{|X_k|} \sum_{x \in X_k} x^j} \tag{4}$$

то есть берётся просто среднее по всем объектам соответствующего класса.

Зная параметры, мы сможем вычислить $a(x)$ по формулам выше.

2 Задание

Вам нужно будет реализовать пуассоновский наивный байесовский классификатор и применить его определения того, является ли рецензия на фильм положительной или отрицательной.

Сдать вам нужно jupyter notebook с выполненным заданием.

2.1 Данные

Данные состоят из 1000 положительных отзывов и 1000 отрицательных отзывов на фильмы с сайта IMDb. Их следует скачать архивом по ссылке `https://yadi.sk/d/GROOCtRVpNbXJ`, а затем распаковать у себя на компьютере.

Каждая новая рецензия представлена в виде отдельного .txt файла. Функции для скачивания и преобразования текстов мы подготовили в jupyter-ноутбуке.

[2] То есть $p(x^j|y)$ — плотность распределения Пуассона с параметром λ_y.

2.2 Задание

1. Загрузите и преобразуйте данные с помощью функции read_txts() из выданного ноутбука. В итоге должно получиться два списка: с положительными и с отрицательными рецензиями.

2. Из всех рецензий сформируйте два списка: тексты для обучающей выборки (по 700 случайных каждого класса) и для контрольной выборки (по 300 оставшихся), а также вектор правильных ответов для обучающей и контрольной выборки. Например, положительные рецензии можно относить к классу «1», а отрицательные — к классу «0».

3. Прочитайте, как работает класс sklearn.feature_extraction.text.CountVectorizer, и с его помощью создайте две матрицы «объекты × признаки»: для обучающей и контрольной выборки. Учтите, что CountVectorizer.transform возвращает разреженную матрицу — чтобы преобразовать её к знакомому нам np.array, воспользуйтесь функцией .toarray().

4. Сами реализуйте класс PoissonNB, реализующий пуассоновский наивный байесовский классификатор. Методы, которые должны быть реализованы в этом классе, описаны в jupyter ноутбуке, выданном вместе с заданием.

5. Протестируйте ваш классификатор на данных и посчитайте accuracy — долю правильных ответов.

6. Протестируйте мультиномиальный и гауссовский наивный байесовский классификатор, реализованный в библиотеке scikit-learn (в sklearn.naive_bayes). Можно использовать параметры по умолчанию.

7. Напишите функцию, которая принимает на вход строку с текстом рецензии, обученный классификатор, обученный объект класса CountVectorizer и печатает, положительна ли данная рецензия.

8. Сделайте выводы, почему наивный байесовский классификатор плохо или хорошо работает для данной задачи.

Бонус 1: Выведите формулу 4 из принципа максимума правдоподобия. Решение можно либо оформить в LaTeX/doc, либо написать от руки, а после — отправить нам его фотографию по почте вместе с заданием.

Бонус 2: Самостоятельно найдите другую выборку текстов и примените к ней три наивных байесовских классификатора.

2.3 Рекомендации по программированию

- Максимизируйте не произведения вероятностей, а логарифм произведения вероятностей
- От вычисления логарифма параметра λ перейдите к вычислению логарифма $\lambda+\varepsilon$, где ε — некоторое маленькое положительное число (например, 1e-9). В таком случае не возникнет ошибок со взятием логарифма нуля.
- Без одного цикла нельзя обойтись только в методе fit. Остальных циклов для эффективной работы программы нужно избегать.
- PoissonNB должен работать не только для двух, но вообще для любого количества классов в задаче.