Наивный Байес

1. Наивный байесовский классификатор

Наивный байесовский классификатор представляет собой еще один *генеративный*[1] алгоритм классификации в добавление к гауссовскому дискриминантному анализу. Напомним, что дискриминативные модели, не вникая в структуру выборки, пытаются разделить объекты разных классов с помощью гиперплоскостей, в то время как генеративные модели находят распределения, которые с наибольшим правдоподобием сгенерировали (породили) объекты разных классов, с чем, собственно, и связаны названиях этих методов.

Наивный байесовский классификатор мы рассмотрим на примере спам-фильтрации. Есть текст, состоящий из произвольного количества слов. Нам необходимо отнести его к одному из двух классов – спам и не спам. Первая задача, которую мы должны решить – это как преобразовать текст произвольной длины в числовой вектор, который уже можно подать на вход классификатору. Для этого мы заведем словарь со всеми словами, которые могут встретиться в наших письмах. Это можно сделать разными способами – либо взять уже готовый словарь, либо проанализировать письма из обучающей выборки. Главная задача словаря – перенумеровать слова, чтобы можно было впоследствии вместо слов использовать индексы (коды). Размер словаря можно ограничить, чтобы спам-фильтр обучался быстрее – например, взять первые N самых часто встречающихся слов. Для всех остальных слов в словаре можно завести специальное слово-метку NOTAWORD, которым будут заменяться все слова, отсутствующие в словаре.

Далее мы представляем каждое письмо в виде вектора признаков длины d (по размеру словаря), где каждый признак отвечает за наличие соответствующего слова в письме. То есть мы устанавливаем $x_j = 1$, если j-тое слово из словаря есть в рассматриваемом письме, и $x_j = 0$ иначе. В результате получаем вектор из нулей и единиц:

$$x = \begin{bmatrix} 0 \\ 1 \\ 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{bmatrix} \begin{matrix} \text{абажур} \\ \text{арбуз} \\ \text{аякс} \\ \vdots \\ \text{купить} \\ \vdots \\ \text{яблоко} \end{matrix}$$

Это одна из форм представления текстовых объектов, называемая мешком слов (bag of words). Отметим, что в этой форме мы отмечаем только факт наличия слова. Есть разные вариации этого формата: в частности, мы могли бы подсчитывать *сколько раз* каждое слово встречается в тексте письма. Для наших целей выбранного нами формата вполне достаточно.

Теперь, научившись преобразовывать текстовые сообщения в числовые вектора, построим нашу генеративную модель классификации. Она должна моделировать условную вероятность $p(y|x)$. Напомним, что x – это конкретное письмо в нашем случае, а y – это конкретный класс, например, спам. Так как напрямую подсчитать эту вероятность не представляется возможным, мы используем теорему Байеса:

$$p(y|x) = \frac{p(x|y)p(y)}{p(x)}.$$

Однако в этой формуле возникает другая проблема – нам нужно подсчитать $p(x|y)$. Если предположить, что в нашем словаре 50000 слов, то $x \in \{0,1\}^{50000}$ (x – 50000-мерный вектор нулей и единиц). Если мы попытаемся построить модель x с помощью категориального распределения,

---

[1] В противоположность *дискриминативным*, к которым относятся, например, логистическая регрессия и машина опорных векторов.

имеющего 2^{50000} значений, то для этого нам потребуется $(2^{50000} - 1)$-мерный вектор параметров[2]. Очевидно, что сделать это невозможно.

Для того чтобы решить эту проблему, мы сделаем одно очень сильное предположение. Мы будем предполагать, что все x_i условно независимы при заданном y. Это предположение называется **наивным предположением о независимости**, а сам алгоритм, его использующий, алгоритмом **наивной байесовской классификации**. То есть, например, если $y = 1$ означает спам, слово «купить» имеет индекс 2087, а слово «цена» индекс 39831, то $p(x_{2087}|y) = p(x_{2087}|y, x_{39831})$. На всякий случай напомним, что условная независимость отличается от просто независимости, которую мы бы записали как $p(x_{2087}) = p(x_{2087}|x_{39831})$.

Благодаря этому предположению мы теперь можем записать:

$$\begin{aligned} p(x|y) &= p(x_1, \dots, x_{50000}|y) = p(x_1|y) \cdot p(x_2|y, x_1) \cdot p(x_3|y, x_1, x_2) \cdot \dots \cdot p(x_{50000}|y, x_1, \dots, x_{49999}) \\ &= p(x_1|y) \cdot p(x_2|y) \cdot p(x_3|y) \cdot \dots \cdot p(x_{50000}|y) = \prod_{j=1}^{50000} p(x_j|y). \end{aligned}$$

Первая строка следует из обычных свойств условного совместного распределения, а вторая – из нашего наивного предположения о независимости. Несмотря на то, что это очень сильное упрощение модели, оно все равно позволяет строить достаточно точные классификаторы, применимые на практике.

Полученная модель имеет следующие параметры:

- $\phi_{j|y=1} = p(x_j = 1|y = 1), j = 1, \dots, d,$
- $\phi_{j|y=0} = p\big(x_j = 1\big|y = 0\big), j = 1, \dots, d,$
- $\phi_y = p(y = 1).$

Здесь d – размер словаря. Теперь можем воспользоваться нашим стандартным приемом. У нас есть выборка $\left\{\left(x^{(i)}, y^{(i)}\right)\right\}_{i=1}^{m}$, к которой необходимо «подогнать» параметры модели. Делаем мы это с помощью функции совместного правдоподобия выборки:

$$\mathcal{L}\left(\phi_y, \phi_{j|y=0}, \phi_{j|y=1}\right) = \prod_{i=1}^{m} p\left(x^{(i)}, y^{(i)}\right).$$

Максимизация этой функции по параметрам дает следующие оценки их максимального правдоподобия:

$$\phi_{j|y=1} = \frac{\sum_{i=1}^{m} 1\left\{x_j^{(i)} = 1 \wedge y^{(i)} = 1\right\}}{\sum_{i=1}^{m} 1\{y^{(i)} = 1\}}, \tag{1}$$

$$\phi_{j|y=0} = \frac{\sum_{i=1}^{m} 1\left\{x_j^{(i)} = 1 \wedge y^{(i)} = 0\right\}}{\sum_{i=1}^{m} 1\{y^{(i)} = 0\}}, \tag{2}$$

$$\phi_y = \frac{\sum_{i=1}^{m} 1\{y^{(i)} = 1\}}{m}. \tag{3}$$

[2] Напомним, что категориальное распределение – это обобщение распределения Бернулли. При распределении Бернулли x принимает всего два значения, поэтому достаточно одного параметра: ϕ – вероятность того, что $x = 1$ (или $x = 0$). Если взять категориальное распределение с $k = 4$, то мы получим $x \in \{0,1,2,3\}$ и теперь нам потребуется трехмерный вектор параметров, чтобы задать распределение: $\phi = (\phi_0, \phi_1, \phi_2)$, где $p(x = 0) = \phi_0$, $p(x = 1) = \phi_1$, $p(x = 2) = \phi_2$ и $p(x = 3) = 1 - \phi_0 - \phi_1 - \phi_2$.

В формулах выше символ $\wedge$ обозначает «И». Обратим внимание, что значения параметров имеют очень естественную интерпретацию. В частности, $\phi_{j|y=1}$ – это просто доля спам сообщений ($y = 1$), в которых встречается слово j.

После вычисления всех этих параметров мы можем использовать модель для прогнозирования класса нового сообщения x. Для этого нам нужно всего лишь вычислить значение

$$p(y = 1|x) = \frac{p(x|y = 1)p(y = 1)}{p(x)}.$$

Знаменатель дроби можно переписать с помощью формулы полной вероятности:

$$p(x) = p(x|y = 1)p(y = 1) + p(x|y = 0)p(y = 0).$$

В результате получаем окончательную формулу наивного байесовского классификатора:

$$p(y = 1|x) = \frac{\left(\prod_{j=1}^{d} p\left(x_j\middle|y = 1\right)\right)p(y = 1)}{\left(\prod_{j=1}^{d} p\left(x_j\middle|y = 1\right)\right)p(y = 1) + \left(\prod_{j=1}^{d} p\left(x_j\middle|y = 0\right)\right)p(y = 0)}. \tag{4}$$

В качестве результата классификации выбираем класс с наибольшей вероятностью.

Еще раз обратим внимание на отличительную особенность метода наивного Байеса. Для того чтобы обучить модель нам нужно всего лишь вычислить значения параметров по формулам (1)-(3). Затем эти значения используются в формуле (4) для вычисления значения гипотезы:

$$h(x) = \underset{c}{\mathrm{arg max}}\{p(y = c|x)\}.$$

2. Сглаживание Лапласа

Описанный в предыдущем разделе наивный байесовский классификатор работает хорошо, но обладает одной проблемой. Посмотрим на примере, что это за проблема, и исправим ее.

Будем по-прежнему рассматривать задачу фильтрации спама. Предположим, мы обучили свой классификатор на некоторой обучающей выборке, а затем получили письмо, в котором встречается некоторое слово под номером 35000 из словаря. И предположим также, что в нашей обучающей выборке этого слова не было[3], то есть наш классификатор видит его в первый раз в жизни. Для этого слова будут вычислены следующие значения параметров:

$$\phi_{35000|y=1} = \frac{\sum_{i=1}^{m} 1\left\{x_{35000}^{(i)} = 1 \wedge y^{(i)} = 1\right\}}{\sum_{i=1}^{m} 1\{y^{(i)} = 1\}} = 0,$$

$$\phi_{35000|y=0} = \frac{\sum_{i=1}^{m} 1\left\{x_{35000}^{(i)} = 1 \wedge y^{(i)} = 0\right\}}{\sum_{i=1}^{m} 1\{y^{(i)} = 0\}} = 0.$$

Напомним, что $\phi_{35000|y=1} = p(x_{35000}|y = 1)$, $\phi_{35000|y=0} = p(x_{35000}|y = 0)$, то есть эти параметры участвуют во всех произведениях формулы (4). В результате мы получаем вероятность:

$$p(y = 1|x) = \frac{0}{0}.$$

Если говорить о проблеме чуть более широко, то со статистической точки зрения это плохая идея считать вероятность какого-то события нулевой только потому, что мы его еще не видели. Предположим, мы хотим оценить ожидаемое значение категориальной случайной величины z,

[3] Вполне реальная ситуация, если свой словарь мы не построили по обучающей выборке, а взяли уже готовый.

принимающей значения из множества $\{1, \dots, k\}$. Мы можем параметризовать категориальное распределение с помощью $\phi_j = p(z = j)$. Пусть у нас теперь есть n независимых наблюдений $\{z^{(1)}, \dots, z^{(n)}\}$. Тогда оценка максимального правдоподобия параметров будут даваться формулой:

$$\phi_j = \frac{\sum_{i=1}^{n} 1\{z^{(i)} = j\}}{n}.$$

Очевидно, что если в выборке каких-то значений случайной величины z не встретилось, то соответствующие ϕ_j будут нулевыми. Для решения этой проблемы воспользуемся **сглаживание Лапласа**, который заменяет вышеприведенную формулу следующей:

$$\phi_j = \frac{1 + \sum_{i=1}^{n} 1\{z^{(i)} = j\}}{k + n}.$$

Здесь мы добавили 1 к числителю и k к знаменателю. Вы можете проверить, что $\sum_{j=1}^{k} \phi_j = 1$, а также, что $\phi_j \neq 0$ для всех значений j.

Возвращаясь к нашему наивному байесовскому классификатору, после добавления в него сглаживания Лапласа получаем:

$$\phi_{j|y=1} = \frac{1 + \sum_{i=1}^{m} 1\left\{x_j^{(i)} = 1 \wedge y^{(i)} = 1\right\}}{2 + \sum_{i=1}^{m} 1\{y^{(i)} = 1\}},$$

$$\phi_{j|y=0} = \frac{1 + \sum_{i=1}^{m} 1\left\{x_j^{(i)} = 1 \wedge y^{(i)} = 0\right\}}{2 + \sum_{i=1}^{m} 1\{y^{(i)} = 0\}}.$$

Для нашей конкретной задачи не имеет особого смысла добавлять сглаживание Лапласа в формулу (3), так как в обучающей выборке для нашего спам-классификатора, очевидно, встретятся письма как той, так и другой категории.

3. Модели событий для классификации текста

Та модель наивной байесовской классификации, которая была описана выше, использует так называемую **бернуллиевскую модель событий**. Эта модель предполагает, что письмо «генерируется» следующим образом: сначала с вероятностью $p(y)$ определяется, кто сгенерирует письмо – спамер или нет, а затем отправитель идет по словарю и независимо для каждого слова j решает, включать ли его в письмо или нет в соответствии с вероятностью $p(x_j = 1|y) = \phi_{j|y}$. Таким образом, результирующая вероятность сообщения, которую получает наша модель, есть $p(y) \prod_{j=1}^{d} p(x_j|y)$ [4].

Есть другая модель, которая называется **мультиномиальной[5] моделью событий**. Для этого определим несколько иначе вектор признаков для представления письма. Пусть теперь x_j равен коду j-го слова в письме, т.е. $x_j \in \{1, \dots, |V|\}$, где $|V|$ – размер словаря. Тогда письмо длины l представляется вектором $(x_1, x_2, \dots, x_l)$; обратите внимание, что l для разных писем может быть разным. Например, если код слова «арбуз» равен 1, код слова «купить» – 35441, тогда сообщение,

[4] Заметим, что это числитель формулы (4), в которой знаменатель играет роль нормирующего множителя, необходимого, чтобы $\sum_y p(y|x) = 1$.

[5] Далее по тексту мы называем использующееся распределение категориальным, каким оно по сути и является. Так как категориальное распределение является частным случаем мультиномиального для значения параметра $n = 1$, то в англоязычной литературе эти два названия часто используются взаимозаменяемо.

начинающееся словами «Купите арбуз! …» будет кодировано вектором, начинающимся с $(1, 35441, \dots)$.

В мультиномиальной модели генерация письма выглядит следующим образом. Сначала с вероятностью $p(y)$ определяется, кто посылает письмо: спамер или нет (как и в бернуллиевской модели), затем отправитель генерирует первое слово x_1 из какого-то категориального распределения на словах $p(x_1|y)$. Далее он независимо определяет следующее слово x_2 из того же категориального распределения и так далее l раз. В результате получаем общую вероятность $p(y)\prod_{j=1}^{l} p(x_j|y)$. Заметьте, что формула выглядит так же, только теперь в произведении l множителей (количество слов в сообщении) вместо d (количество слов в словаре) и каждый из них имеет другой смысл, а именно, $x_j|y$ теперь имеет категориальное распределение вместо Бернулли.

Параметрами новой модели являются $\phi_y = p(y)$ как и ранее, а также $\phi_{k|y=1} = p(x_j = k|y = 1)$, $\phi_{k|y=0} = p(x_j = k|y = 0)$, $j = 1, \dots, l$. Также заметим, что мы предположили, что $p(x_j|y)$ одинаковое для всех j (т.е. распределение вероятности, с которым генерируется слово, не зависит от его позиции в тексте).

Пусть нам дана обучающая выборка $\left\{\left(x^{(i)}, y^{(i)}\right); i = 1, \dots, m\right\}$, где $x^{(i)} = \left(x_1^{(i)}, x_2^{(i)}, \dots, x_{d_i}^{(i)}\right)$. Здесь d_i означает количество слов в i-том обучающем примере. Тогда функция правдоподобия данных выглядит следующим образом:

$$\mathcal{L}\left(\phi_y, \phi_{k|y=0}, \phi_{k|y=1}\right) = \prod_{i=1}^{m} p\left(x^{(i)}, y^{(i)}\right) = \prod_{i=1}^{m} \left(\prod_{j=1}^{d_i} p\left(x_j^{(i)} | y^{(i)}; \phi_{k|y=0}, \phi_{k|y=1}\right) \right) p\left(y^{(i)}; \phi_y\right).$$

Ее максимизация приводит к следующим оценкам максимального правдоподобия:

$$\phi_{k|y=1} = \frac{\sum_{i=1}^{m} \sum_{j=1}^{d_i} 1\left\{x_j^{(i)} = k \wedge y^{(i)} = 1\right\}}{\sum_{i=1}^{m} 1\{y^{(i)} = 1\} d_i},$$
$$\phi_{k|y=0} = \frac{\sum_{i=1}^{m} \sum_{j=1}^{d_i} 1\left\{x_j^{(i)} = k \wedge y^{(i)} = 0\right\}}{\sum_{i=1}^{m} 1\{y^{(i)} = 0\} d_i},$$
$$\phi_y = \frac{\sum_{i=1}^{m} 1\{y^{(i)} = 1\}}{m}.$$

Если мы захотим применить сглаживание Лапласа, то нужно добавить единицу в числитель и $|V|$ в знаменатель:

$$\phi_{k|y=1} = \frac{1 + \sum_{i=1}^{m} \sum_{j=1}^{d_i} 1\left\{x_j^{(i)} = k \wedge y^{(i)} = 1\right\}}{|V| + \sum_{i=1}^{m} 1\{y^{(i)} = 1\} d_i},$$
$$\phi_{k|y=0} = \frac{1 + \sum_{i=1}^{m} \sum_{j=1}^{d_i} 1\left\{x_j^{(i)} = k \wedge y^{(i)} = 0\right\}}{|V| + \sum_{i=1}^{m} 1\{y^{(i)} = 0\} d_i}.$$