Наивный байесовский классификатор — простой вероятностный классификатор, основанный на применении теоремы Байеса со строгими (наивными) предположениями о независимости.

В зависимости от точной природы вероятностной модели, наивные байесовские классификаторы могут обучаться очень эффективно. Во многих практических приложениях для оценки параметров для наивных байесовых моделей используют метод максимального правдоподобия; другими словами, можно работать с наивной байесовской моделью.

Несмотря на наивный вид и, несомненно, очень упрощенные условия, наивные байесовские классификаторы часто работают намного лучше во многих сложных жизненных ситуациях.

Достоинством наивного байесовского классификатора является малое количество данных необходимых для обучения, оценки параметров и классификации.

Теорема Байеса (или формула Байеса) — одна из основных теорем элементарной теории вероятностей, которая позволяет определить вероятность какого-либо события при условии, что произошло другое статистически взаимозависимое с ним событие. Другими словами, по формуле Байеса можно более точно пересчитать вероятность, взяв в расчёт как ранее известную информацию, так и данные новых наблюдений. Формула Байеса может быть выведена из основных аксиом теории вероятностей, в частности из условной вероятности.

Особенность теоремы Байеса заключается в том, что для её практического применения требуется большое количество расчётов, вычислений, поэтому байесовские оценки стали активно использовать только после революции в компьютерных и сетевых технологиях.

Формула Байеса:

$$\mathrm{P}(\mathrm{A} \mid \mathrm{B}) = \frac{P(B \mid A)}{P(B)},$$

где P(A) — априорная вероятность гипотезы A;
P(A|B) — вероятность гипотезы A при наступлении события B (апостериорная вероятность);
P(B|A) — вероятность наступления события B при истинности гипотезы A;

Априорное распределение вероятностей неопределённой величины p — распределение вероятностей, которое выражает предположения о p до учёта экспериментальных данных.

Например, если p — доля избирателей, готовых голосовать за определённого кандидата, то априорным распределением будет предположение о p до учёта результатов опросов или выборов. Противопоставляется апостериорной вероятности.

Для доказательства формулы Байеса, нам понадобится знание о условной вероятности.

Усло́вная вероя́тность — вероятность наступления события A при условии, что событие B произошло.

Вероятность события A, вычисленную в предположении, что о результате эксперимента уже что-то известно (событие B произошло), мы будем обозначать через P(A|B).

Очевидный частный случай: P(A|A)=1=100% неплохо иллюстрируется шуткой «Интернет-опрос показал, что 100% граждан России пользуются интернетом».

Вероятность совместного появления двух зависимых событий равна произведению вероятности одного из них на условную вероятность второго, вычисленную при условии, что первое событие произошло, т.е.

P(AB)= P(A|B) P(B)= P(B|A)P(A)

И как было уже сказано ранее, Формула Байеса может быть выведена из основных аксиом условной вероятности.

Доказательство

$$P(AB) = P(A \mid B)P(B) = P(B \mid A)P(A)$$

Следовательно

$$P(A \mid B) = \frac{P(AB)}{P(B)} = \frac{P(B \mid A)P(A)}{P(B)}$$

Как работает наивный байесовский алгоритм?

Давайте рассмотрим пример. Перед нами стоит задача: Состоится ли матч при солнечной погоде (sunny)? Что же мы имеем? Ниже, слева, представлен обучающий набор данных, содержащий один признак «Погодные условия» (weather) и целевую переменную «Игра» (play), которая обозначает возможность проведения матча. На основе погодных условий мы должны определить, состоится ли матч. Чтобы сделать это, необходимо выполнить следующие шаги.

Шаг 1. Преобразуем набор данных в частотную таблицу (frequency table). Где, сгруппируем нашу таблицу по погоде.

Шаг 2. Далее создадим таблицу правдоподобия (likelihood table), рассчитав соответствующие вероятности. Например, вероятность облачной погоды (overcast) составляет 0,29(Мы делим количество матчей с облачной погодой, на общее число матчей с различной погодой), а вероятность того, что матч состоится (yes) – 0,64(Делим кол-во состоявшихся матчей, но их общее кол-во).

Weather	Play
Sunny	No
Overcast	Yes
Rainy	Yes
Sunny	Yes
Sunny	Yes
Overcast	Yes
Rainy	No
Rainy	No
Sunny	Yes
Rainy	Yes
Sunny	No
Overcast	Yes
Overcast	Yes
Rainy	No

Frequency Table		
Weather	No	Yes
Overcast		4
Rainy	3	2
Sunny	2	3
Grand Total	5	9

Likelihood table				
Weather	No	Yes		
Overcast		4	=4/14	0.29
Rainy	3	2	=5/14	0.36
Sunny	2	3	=5/14	0.36
All	5	9		
	=5/14	=9/14		
	0.36	0.64		

Sunny – Солнечная погода
Rainy – Дождливая погода
Overcast – Облачная погода

Шаг 3. С помощью теоремы Байеса рассчитаем апостериорную вероятность для каждого класса при данных погодных условиях. (*Апостерио́рная вероя́тность — условная вероятность случайного события при условии того, что известны данные, полученные после опыта или события.)* Класс с наибольшей апостериорной вероятностью будет результатом прогноза.

Задача. Состоится ли матч при солнечной погоде (sunny)?

Мы можем решить эту задачу с помощью описанного выше подхода.

*P(Yes | Sunny) = P(Sunny | Yes) * P(Yes) / P(Sunny)*

Здесь мы имеем следующие значения:

P(Sunny | Yes) = 3 / 9 = 0,33
P(Sunny) = 5 / 14 = 0,36
P(Yes) = 9 / 14 = 0,64

Теперь рассчитаем *P(Yes | Sunny)*:
*P(Yes | Sunny) = 0,33 * 0,64 / 0,36 = 0,60*

Значит, при солнечной погоде более вероятно, что матч состоится.

Модель наивного байесовского классификатора

Вероятностная модель для классификатора — это условная модель

$$p(C \mid F_1, \ldots, F_n)$$

над зависимой переменной класса C с малым количеством результатов или классов, зависимая от нескольких переменных $F_1, \ldots, F_n$. Проблема заключается в том, что когда количество свойств n очень велико или когда свойство может принимать большое количество значений, тогда строить такую модель на вероятностных таблицах становится невозможно. Поэтому мы переформулируем модель, чтобы сделать её легко поддающейся обработке.

Используя теорему Байеса, запишем

$$p(C \mid F_1, \ldots, F_n) = \frac{p(C)\, p(F_1, \ldots, F_n \mid C)}{p(F_1, \ldots, F_n)}.$$

На практике интересен лишь числитель этой дроби, так как знаменатель не зависит от C и значения свойств F_i даны, так что знаменатель — константа.

Числитель эквивалентен совместной вероятности модели

$$p(C,\ F_1,\ \ldots,\ F_n)$$

которая может быть переписана следующим образом, используя повторные приложения определений условной вероятности:

$$p(C,\ F_1,\ \ldots,\ F_n) = p(C)p(F_1,\ \ldots,\ F_n \mid C) = p(C)p(F_1 \mid C)p(F_2,\ \ldots,\ F_n \mid C, F_1) = p(C)p(F_1 \mid C)\, p(F_2 \mid C,\ F_1)p(F_3,\ \ldots,\ F_n \mid C, F_1,\ F_2) = p(C)p(F_1 \mid C)\, p(F_2 \mid C,\ F_1)\ldots p(F_n \mid C, F_1,\ F_2, F_3,\ \ldots, F_{n-1})$$

и т. д.

Теперь можно использовать «наивные» предположения условной независимости: предположим, что каждое свойство F_i условно независимо от любого другого свойства F_j при $j \neq i$. Это означает:

Если вероятность появления события $F_i|C$, не зависит от F_j, значит F_j можно отбросить

$$p\left(F_i \middle| C,\ F_j\right) = p\left(F_i \middle| C\right)$$

таким образом, совместная модель может быть выражена как:

$$p\left(C,\ F_1, \ldots,\ F_n\right) =\ p(C)p\left(F_1 \middle| C\right)\, p\left(F_2 \middle| C\right)p\left(F_3 \middle| C\right) \quad \ldots$$

$$p\left(F_n \middle| C\right) =\ p(C)\prod_{i=1}^{n} p\left(F_i \middle| C\right)$$

Это означает, что из предположения о независимости, условное распределение по классовой переменной C может быть выражено так:

$$p\left(C,\ F_1, \ldots,\ F_n\right) = \frac{1}{Z}\, p(C)\prod_{i=1}^{n} p\left(F_i \middle| C\right)$$

где $Z = p(F_1, \ldots,\ F_n)$ — это масштабный множитель, зависящий только от $F_1, \ldots,\ F_n$, то есть константа, если значения переменных известны.

Оценка параметров

Все параметры модели могут быть аппроксимированы относительными частотами из набора данных обучения. Это оценки максимального правдоподобия вероятностей. Непрерывные свойства, как правило, оцениваются через нормальное

распределение. В качестве математического ожидания и дисперсии вычисляются статистики — среднее арифметическое и среднеквадратическое отклонение соответственно.

Если данный класс и значение свойства никогда не встречаются вместе в наборе обучения, тогда оценка, основанная на вероятностях, будет равна нулю. Это проблема, так как при перемножении нулевая оценка приведет к потере информации о других вероятностях. Поэтому предпочтительно проводить небольшие поправки во все оценки вероятностей так, чтобы никакая вероятность не была строго равна нулю.

Построение классификатора по вероятностной модели

Наивный байесовский классификатор объединяет модель с правилом решения. Одно общее правило должно выбрать наиболее вероятную гипотезу; оно известно *как апостериорное правило принятия решения (MAP)*. Соответствующий классификатор — это функция *classify*, определённая следующим образом:

$$classify(f_1, \ldots, f_n) = \operatorname*{argmax}_c p(C = c) \prod_{i=1}^{n} p(F_i = f_i \mid C = c)$$

Байесовская фильтрация спама — метод для фильтрации спама, основанный на применении наивного байесовского классификатора, опирающегося на прямое использование теоремы Байеса.

История

Первой известной программой, фильтрующей почту с использованием байесовского классификатора, была программа iFile Джейсона Ренни, выпущенная в 1996 году. Программа использовала сортировку почты по папкам. Первая академическая публикация по наивной байесовской фильтрации спама появилась в 1998 году. Вскоре после этой публикации была развернута работа по созданию коммерческих фильтров спама. Однако в 2002 г. Пол Грэм смог значительно уменьшить число ложноположительных срабатываний до такой степени, что байесовский фильтр мог использоваться в качестве единственного фильтра спама.

Описание

При обучении фильтра для каждого встреченного в письмах слова высчитывается и сохраняется его «вес» — оценка вероятности того, что письмо с этим словом — спам. В простейшем случае в качестве оценки используется частота: «появлений в спаме / появлений всего». В более сложных случаях возможна предварительная обработка текста: приведение слов в начальную форму, удаление служебных слов, вычисление «веса» для целых фраз, транслитерация и прочее.

При проверке вновь пришедшего письма вероятность «спамовости» вычисляется по формуле ($classify$) для множества гипотез. В данном случае «гипотезы» — это слова, и для каждого слова «достоверность гипотезы» $P(A_i) = N_{wordi}/N_{words\ total}$ — доля этого слова в письме, а «зависимость события от гипотезы» $P(B|A_i)$ — вычисленный ранее «вес» слова. То есть «вес» письма в данном случае — усреднённый «вес» всех его слов.

Отнесение письма к «спаму» или «не-спаму» производится по тому, превышает ли его «вес» некую планку, заданную пользователем (обычно берут 60-80 %). После принятия решения по письму в базе данных обновляются «веса» для вошедших в него слов.

Математические основы

Почтовые байесовские фильтры основываются на теореме Байеса. Теорема Байеса используется несколько раз в контексте спама:

- в первый раз, чтобы вычислить вероятность, что сообщение — спам, зная, что данное слово появляется в этом сообщении;
- во второй раз, чтобы вычислить вероятность, что сообщение — спам, учитывая все его слова (или соответствующие их подмножества);
- иногда в третий раз, когда встречаются сообщения с редкими словами.

Вычисление вероятности того, что сообщение, содержащее данное слово, является спамом

Давайте предположим, что подозреваемое сообщение содержит слово «Replica». Большинство людей, которые привыкли получать электронное письмо, знает, что это сообщение, скорее всего, будет спамом, а точнее — предложением продать поддельные копии часов известных брендов. Программа обнаружения спама, однако, не «знает» такие факты; всё, что она может сделать — вычислить вероятности.

Формула, используемая программным обеспечением, чтобы определить это, получена из теоремы Байеса и формулы полной вероятности:

$$\Pr(S|W) = \frac{\Pr(W|S)*\Pr(S)}{\Pr(W)} = \frac{\Pr(W|S)*\Pr(S)}{\Pr(W|S)*\Pr(S) + \Pr(W|H)*\Pr(H)}$$

где:

$\Pr(S|W)$ — условная вероятность того, что сообщение—спам(S), при условии, что слово «Replica» находится в нём(W);

Pr (W) – полная вероятность того, что слово «Replica» содержится в сообщении

$\Pr(S)$ — полная вероятность того, что произвольное сообщение—спам;

$\Pr(W \mid S)$ — условная вероятность того, что слово «replica» появляется в сообщениях, если они являются спамом;

$\Pr(H)$ — полная вероятность того, что произвольное сообщение не спам (то есть «ham»);

$\Pr(W \mid H)$ — условная вероятность того, что слово «replica» появляется в сообщениях, если они являются «ham».

Спамовость слова

Недавние статистические исследования показали, что на сегодняшний день вероятность любого сообщения быть спамом составляет по меньшей мере 80 %: Pr(*S*)=0.8; Pr(*H*)=0.2.

Однако большинство байесовских программ обнаружения спама делают предположение об отсутствии априорных предпочтений у сообщения быть «spam», а не «ham», и полагает, что у обоих случаев есть равные вероятности 50 %: Pr(*S*)=0.5, Pr(*H*)=0.5.

О фильтрах, которые используют эту гипотезу, говорят как о фильтрах «без предубеждений». Это означает, что у них нет никакого предубеждения относительно входящей электронной почты. Данное предположение позволяет упрощать общую формулу до:

$$\Pr(S \mid W) = \frac{\Pr(W \mid S)}{\Pr(W \mid S) + \Pr(W \mid H)}$$

Значение $\Pr(S \mid W)$ называют «спамовостью» слова *W*; при этом число $\Pr(W \mid S)$, используемое в приведённой выше формуле, приближённо равно относительной частоте сообщений, которые содержат слово *W* в сообщениях, идентифицированных как спам во время фазы обучения, то есть:

$$\Pr(W_i \mid S) = \frac{count(M: W_i \epsilon M,\ M \epsilon S)}{\sum_j count(M: W_j \epsilon M,\ M \epsilon S)}$$

Точно так же $\Pr(W \mid H)$ приближённо равно относительной частоте сообщений, содержащих слово *W* в сообщениях, идентифицированных как «ham» во время фазы обучения.

$$\Pr(W_i \mid H) = \frac{count(M: W_i \epsilon M,\ M \epsilon H)}{\sum_j count(M: W_j \epsilon M,\ M \epsilon H)}$$

Для того, чтобы эти приближения имели смысл, набор обучающих сообщений должен быть большим и достаточно представительным. Также желательно, чтобы набор обучающих сообщений соответствовал 50 % гипотезе о перераспределении между спамом и «ham», то есть что наборы сообщений «spam» и «ham» имели один и тот же размер.

Конечно, определение, является ли сообщение «spam» или «ham», базируемой только на присутствии лишь одного определённого слова, подвержено ошибкам. Именно поэтому байесовские фильтры спама пытаются рассмотреть несколько слов и комбинировать их спамовость, чтобы определить полную вероятность того, что сообщение является спамом.

Объединение индивидуальных вероятностей

Программные спам-фильтры, построенные на принципах наивного байесовского классификатора, делают «наивное» предположение о том, что события, соответствующие наличию того или иного слова в электронном письме или сообщении, являются независимыми по отношению друг к другу. Это упрощение в общем случае является неверным для естественных языков — таких, как английский, где вероятность обнаружения прилагательного повышается при наличии, к примеру, существительного. Исходя из такого «наивного» предположения, для решения задачи классификации сообщений лишь на 2 класса: S (спам) и $H = \neg S$ («хэм», то есть не спам) из теоремы Байеса можно вывести следующую формулу оценки вероятности «спамовости» всего сообщения, содержащего слова $W_1,\ W_2,\ \ldots\ W_N$

$p(S \mid W_1,\ W_2,\ \ldots\ W_N) =$ [по теореме Байеса]

$$= \frac{p(W_1,\ W_2,\ \ldots\ W_N \mid S) * p(S)}{p(W_1,\ W_2,\ \ldots\ W_N)} =$$

= [так как W_i предполагаются независимыми]=

$$= \frac{\prod_i p(W_i \mid S) * p(S)}{p(W_1,\ W_2,\ \ldots\ W_N)} =$$

= [по теореме Байеса]

$$= \frac{\prod_i \frac{p(S|W_i)*p(W_i)}{p(S)} * p(S)}{p(W_1,\ W_2,\ \dots\ W_N)} =$$

= [по формуле полной вероятности] =

$$= \frac{\prod_i \frac{p(S|W_i)*p(W_i)}{p(S|W_i)*p(W_i)} * p(S)}{\prod_i (p(W_i|S))*p(S) + \prod_i (p(W_i|\neg S))*p(\neg S)} =$$

$$= \frac{\prod_i (p(S|W_i)*p(W_i))*p(S)^{1-N}}{\prod_i (p(S|W_i)*p(W_i))*p(S)^{1-N} + \prod_i (p(\neg S|W_i)*p(W_i))*p(\neg S)^{1-N}} =$$

$$\frac{\prod_i p(S|W_i)}{\prod_i (p(S|W_i)) + \left(\frac{p(\neg S)}{p(S)}\right)^{1-N} * \prod_i p(\neg S|W_i)}$$

Таким образом, предполагая $p(S) = p(\neg S)$=0.5, имеем:

$$p = \frac{p_1 p_2 \dots p_N}{p_1 p_2 \dots p_N + (1-p_1)(1-p_2)\dots(1-p_N)}$$

где:

$p = \Pr(S \mid W_1,\ W_2,\ \dots\ W_N)$ — вероятность, что сообщение, содержащее слова $W_1,\ W_2,\ \dots\ W_N$— спам;

p_1 — условная вероятность $p(S|W_1)$ того, что сообщение — спам, при условии, что оно содержит первое слово (к примеру, «replica»);

p_2 — условная вероятность $p(S|W_2)$ того, что сообщение — спам, при условии, что оно содержит второе слово (к примеру, «watches»);

p_N — условная вероятность $p(S|W_N)$ того, что сообщение — спам, при условии, что оно содержит N-е слово (к примеру, «home»).

Результат *p* обычно сравнивают с некоторым порогом (например, *0.5*, чтобы решить, является ли сообщение спамом или нет. Если p ниже, чем порог, сообщение рассматривают как вероятный «ham», иначе его рассматривают как вероятный спам.

Проблема редких слов

Она возникает в случае, если слово никогда не встречалось во время фазы обучения: и числитель, и знаменатель равны нулю, и в общей формуле, и в формуле спамовости.

В целом, слова, с которыми программа столкнулась только несколько раз во время фазы обучения, не являются репрезентативными (набор данных в выборке мал для того, чтобы сделать надёжный вывод о свойстве такого слова). Простое решение состоит в том, чтобы игнорировать такие ненадёжные слова.

Другие эвристические усовершенствования

«Нейтральные» слова — такие, как, «the», «a», «some», или «is» (в английском языке), или их эквиваленты на других языках — могут быть проигнорированы. Вообще говоря, некоторые байесовские фильтры просто игнорируют все слова, у которых спамовость около 0.5, так как в этом случае получается качественно лучшее решение. Учитываются только те слова, спамовость которых около 0.0 (отличительный признак законных сообщений — «ham»), или рядом с 1.0 (отличительный признаки спама). Метод отсева может быть настроен, например, так, чтобы держать только те десять слов в исследованном сообщении, у которых есть самое большое absolute value $|0.5 - Pr|$.

Характеристика

Данный метод прост (алгоритмы элементарны), удобен (позволяет обходиться без «чёрных списков» и подобных искусственных приёмов), эффективен (после обучения на достаточно большой выборке отсекает до 95—97 % спама), причём в случае любых ошибок его можно дообучать. В общем, есть все показания для его повсеместного использования, что и имеет место на практике — на его основе построены практически все современные спам-фильтры.

Впрочем, у метода есть и принципиальный недостаток: он базируется на предположении, что одни слова чаще встречаются в спаме, а другие — в обычных письмах, и неэффективен, если данное предположение неверно. Ещё один не принципиальный недостаток, связанный с реализацией — метод работает только с текстом. Зная об этом ограничении, спамеры стали вкладывать рекламную информацию в картинку. Текст же в письме либо отсутствует, либо не несёт смысла. Против этого приходится пользоваться либо средствами распознавания текста («дорогая» процедура, применяется только при крайней необходимости), либо старыми методами фильтрации — «чёрные списки» и регулярные выражения (так как такие письма часто имеют стереотипную форму).