Содержание

Предисловие

Идея этой книги появилась летом 2008 года, когда Бристольский университет, где я в то время работал, предоставил мне годичную стипендию для проведения научных исследований. Написать общее введение в машинное обучение я решил по двум причинам. Во-первых, существовала очевидная потребность в такой книге, которая дополняла бы многочисленные специальные издания на эту тему, а во-вторых, это позволило бы мне самому изучить что-то новое – ведь, как известно, лучший способ чему-то научиться – начать это преподавать.

Перед любым автором, желающим написать вводный учебник по машинному обучению, стоит трудная задача – осветить невероятно богатый материал, не упустив из виду объединяющих его принципов. Стоит уделить чрезмерно много внимания разнообразию дисциплины – и вы рискуете получить сборник слабо связанных между собой «рецептов». А увлекшись какой-нибудь из своих излюбленных парадигм, вы оставите за бортом массу других не менее интересных вещей. В конце концов, методом проб и ошибок я остановился на подходе, который позволяет подчеркнуть как единство, так и разнообразие. Единство достигается путем разделения *задач* и *признаков* – и то, и другое присутствует в любом подходе к машинному обучению, но зачастую принимается как само собой разумеющееся. А разнообразие обеспечивается рассмотрением широкого круга логических, геометрических и вероятностных моделей.

Понятно, что сколько-нибудь глубоко охватить весь предмет машинного обучения на 400 страницах нет никакой надежды. В эпилоге перечисляется ряд важных дисциплин, которые я решил не включать. На мой взгляд, машинное обучение – это сочетание статистики и представления знаний, и темы для книги были отобраны соответственно. Поэтому сначала довольно подробно рассматриваются решающие деревья и обучение на основе правил, а затем я перехожу к материалу, основанному на применении математической статистики. В книге постоянно подчеркивается важность интуиции, подкрепленная многочисленными примерами и иллюстрациями, многие из которых взяты из моих работ по применению РХП-анализа[1] в машинном обучении.

Как читать эту книгу

Печатный текст по природе своей линеен, поэтому материал организован так, чтобы книгу можно было читать последовательно. Но это не значит, что выборочное чтение невозможно, – я старался строить изложение по модульному принципу.

Например, если вы хотите поскорее приступить к первому алгоритму обучения, можете начать с раздела 2.1, где описывается бинарная классификация,

[1] Рабочая характеристика приемника (ROC – receiver operating characteristic). – *Прим. перев.*

а затем сразу перейти к главе 5, посвященной решающим деревьям, – без существенного нарушения непрерывности изложения. Прочитав раздел 5.1, можно перескочить к первым двум разделам главы 6, чтобы узнать о классификаторах на основе правил.

С другой стороны, читатель, интересующийся линейными моделями, может после раздела 2.1 перейти к разделу 3.2 о регрессионных задачах, а затем к главе 7, которая начинается с рассмотрения линейной регрессии. В порядке следования глав 4–9, посвященных логическим, геометрическим и вероятностным моделям, есть определенная логика, но по большей части их можно читать независимо; то же самое относится к главам 10–12 о признаках, ансамблях моделей и экспериментах в машинном обучении.

Отмечу также, что пролог и глава 1 носят вступительный характер и в значительной мере независимы от всего остального: в прологе есть кое-какие технические детали, но все они должны быть понятны даже человеку, не обучавшемуся в университете, а глава 1 содержит сжатый общий обзор материала, изложенного в книге. Обе главы можно бесплатно скачать с сайта книги по адресу www.cs.bris.ac.uk/~flach/mlbook; со временем будут добавлены и другие материалы, в частности лекционные слайды. Поскольку в книге такого объема неизбежны мелкие погрешности, на сайте имеется форма, с помощью которой мне можно отправить извещения о замеченных ошибках и опечатках.

Благодарности

Работа над книгой одного автора всегда была уделом одиночек, но мне повезло – я пользовался помощью и поддержкой многочисленных друзей и коллег. Тим Ковач (Tim Kovacs) в Бристоле, Люк де Редт (Luc De Raedt) в Лёвене и Карла Бродли (Carla Brodley) в Бостоне организовали группы читателей, от которых я получал весьма полезные отзывы. Своими замечаниями со мной любезно поделились также Хендрик Блокил (Hendrik Blockeel), Натали Япковиц (Nathalie Japkowicz), Николас Лашиш (Nicolas Lachiche), Мартин ван Отерло (Martijn van Otterlo), Фабрицио Ригуци (Fabrizio Riguzzi) и Мохак Шах (Mohak Shah). Тем или иным способом мне помогали и многие другие – спасибо всем.

Хосе Хернандес-Оралло (Jose Hernandez-Orallo), выйдя далеко за пределы служебных обязанностей, внимательно прочитал рукопись и высказал много конструктивных критических замечаний, которые я учел в той мере, в какой позволяло время. Хосе, с меня причитается.

Большое спасибо моим коллегам по Бристольскому университету: Тареку Абудавуду (Tarek Abudawood), Рафалу Богачу (Rafal Bogacz), Тило Бургхардту (Tilo Burghardt), Нелло Кристианини (Nello Cristianini), Тийл де Бье (Tijl De Bie), Бруно Голениа (Bruno Golenia), Саймону Прайсу (Simon Price), Оливеру Рэю (Oliver Ray) и Себастьяну Шпиглеру (Sebastian Spiegler) – за совместную работу и поучительные дискуссии. Благодарю также своих зарубежных коллег Иоханнеса Фурнкранца (Johannes Furnkranz), Цезаря Ферри (Cesar Ferri), Томаса

Гартнера (Thomas Gartner), Хосе Хернандес-Оралло, Николаса Лашиша (Nicolas Lachiche), Джона Ллойда (John Lloyd), Эдсона Мацубара (Edson Matsubara) и Роналдо Прати (Ronaldo Prati) за разрешение использовать в книге результаты нашей совместной работы и за иную помощь. В те моменты, когда проекту требовалось резкое ускорение, мне любезно обеспечивали уединение Керри, Пол, Дэвид, Рэни и Тринти.

Дэвид Транах (David Tranah) из издательства Кэмбридж Юниверсити Пресс всемерно содействовал запуску проекта и предложил пуантилистскую метафору для «извлечения знаний из данных», которая нашла отражение в рисунке на обложке книги (по мысли Дэвида, это «обобщенный силуэт», не обладающий портретным сходством ни с кем конкретно). Мэйри Сазерленд (Mairi Sutherland) тщательно отредактировала рукопись перед сдачей в набор.

Я посвящаю свой труд покойному отцу, который, безусловно, откупорил бы бутылку шампанского, узнав, что «эта книга» наконец закончена. Его трактовка проблемы индукции наводила на интересные, хотя и мрачноватые, размышления: та же рука, что каждый день кормит курицу, однажды сворачивает ей шею (приношу извинения читателям-вегетарианцам). Я благодарен своим родителям за то, что они снабдили меня всем необходимым для поиска собственного жизненного пути.

И наконец, не выразить словами все, чем я обязан своей жене Лайзе. Я начал писать эту книгу вскоре после нашей свадьбы, и мы даже представить не могли, что на эту работу уйдет почти четыре года. Ретроспективный взгляд – отличная штука: например, таким образом можно установить, что попытка закончить книгу одновременно с организацией международной конференции и капитальным ремонтом дома – не самая здравая мысль. Но наградой Лайзе за поддержку, ободрение и молчаливые страдания стал тот факт, что все три вещи все-таки приближаются к успешному завершению. Спасибо, любимая!

Петер Флах, Бристоль

Пролог: пример машинного обучения

Очень может быть, что вы, сами того не подозревая, давно уже пользуетесь технологиями машинного обучения. В большинство почтовых клиентов встроены алгоритмы определения и фильтрации спама, или нежелательной почты. Первые фильтры спама основывались на технике сопоставления с образцами, например с регулярными выражениями, причем сами образцы кодировались вручную. Однако скоро стало понятно, что этот подход непригоден для сопровождения и зачастую не обладает достаточной гибкостью – ведь что для одного спам, то для другого желанное послание[1]! Необходимая адаптивность и гибкость достигается с помощью методов машинного обучения.

SpamAssassin – широко известный фильтр спама с открытым исходным кодом. Он вычисляет оценку входящего почтового сообщения, опираясь на ряд встроенных правил, или «критериев» и, если оценка оказывается не менее 5, включает в заголовки сообщения признак «спам» и сводный отчет. Вот пример отчета для полученного мной сообщения:

```
-0.1 RCVD_IN_MXRATE_WL RBL:         MXRate recommends allowing
                                    [123.45.6.789 listed in sub.mxrate.net]
0.6 HTML_IMAGE_RATIO_02 BODY:       HTML has a low ratio of text to image area
1.2 TVD_FW_GRAPHIC_NAME_MID BODY:   TVD_FW_GRAPHIC_NAME_MID
0.0 HTML_MESSAGE BODY:              HTML included in message
0.6 HTML_FONx_FACE_BAD BODY:        HTML font face is not a word
1.4 SARE_GIF_ATTACH FULL:           Email has a inline gif
0.1 BOUNCE_MESSAGE                  MTA bounce message
0.1 ANY_BOUNCE_MESSAGE              Message is some kind of bounce message
1.4 AWL                             AWL: From: address is in the auto white-list
```

Слева направо идут: оценка, вычисленная для данного критерия, идентификатор критерия и краткое описание, содержащее ссылку на релевантную часть сообщения. Как видим, оценка критерия может быть как отрицательной (свидетельство в пользу того, что сообщение не является спамом), так и положительной. Поскольку итоговая оценка равна 5.3, то сообщение может быть спамом. На самом деле это конкретное сообщение было уведомлением от промежуточного сервера о том, что какое-то другое сообщение, с чудовищной оценкой 14.6, было сочтено спамом и отвергнуто. В сообщение о недоставке было включено и исходное сообщение, а значит, оно унаследовало кое-какие характеристики по-

[1] Слово spam произошло от слияния двух слов: «spiced ham» (пряная ветчина). Так называлось мясное изделие, получившее недобрую славу после высмеивания в сериале «Летающий цирк Монти Пайтона». Поэтому в оригинале употреблена игра слов «one person's spam is another person's ham» – «что одному спам, то другому ветчина». – *Прим. перев.*

следнего, в частности низкое отношение текста к графике; отсюда и оценка, превышающая пороговое значение 5.

А вот пример важного письма, которого я долго ждал и в конце концов обнаружил в папке для спама:

```
2.5 URI_NOVOWEL URI:        URI hostname has long non-vowel sequence
3.1 FROM_DOMAIN_NOVOWEL     From: domain has series of non-vowel letters
```

Это письмо касалось работы, которую мы с коллегой отправили на Европейскую конференцию по машинному обучению (ECML) и на Европейскую конференцию по принципам и практическим методам выявления знаний в базах данных (European Conference on Principles and Practice of Knowledge Discovery in Databases – PKDD), которые, начиная с 2001 года, проводятся совместно. Для освещения этих конференций, состоявшихся в 2008 году, в Интернете был создан домен www.ecmlpkdd2008.org, пользующийся заслуженно высокой репутацией у специалистов по машинному обучению. Однако в имени домена подряд идут одиннадцать согласных – вполне достаточно, чтобы возбудить подозрения у SpamAssassin! Этот пример наглядно демонстрирует, что ценность критериев SpamAssassin для разных пользователей может быть различна. Машинное обучение – великолепный способ создания программ, адаптирующихся к пользователю.

Как SpamAssassin вычисляет оценки, или «веса», для каждого из нескольких десятков критериев? Вот тут-то и приходит на помощь машинное обучение. Допустим, что имеется большой «обучающий набор» почтовых сообщений, которые были вручную помечены как «спам» или «неспам», и для каждого сообщения известны результаты по каждому критерию. Наша цель – вычислить вес для каждого критерия таким образом, чтобы все спамные сообщения получили итоговую оценку выше 5, а все хорошие – оценку ниже 5. В этой книге мы будем обсуждать различные способы решения этой задачи. А пока на простом примере проиллюстрируем основную идею.

Пример 1 (линейная классификация). Предположим, что есть только два критерия и четыре обучающих сообщения, одно из которых – спам (см. табл. 1). Для спамного сообщения оба критерия удовлетворяются, для одного хорошего не удовлетворяется ни один критерий, для второго удовлетворяется только первый, а для третьего – только второй. Легко видеть, что, приписав каждому критерию вес 4, мы сможем правильно «классифицировать» все четыре сообщения. В математической нотации, описываемой в замечании 1, этот классификатор можно было бы записать в виде $4x_1 + 4x_2 > 5$ или $(4,4) \cdot (x_1,x_2) > 5$. На самом деле при любом весе между 2.5 и 5 порог будет превышен, только когда удовлетворяются оба критерия. Можно было бы даже назначить критериям разные веса – при условии, что каждый из них меньше 5, а сумма больше 5; правда, не понятно, зачем нужны такие сложности при имеющихся обучающих данных.

Сообщение	x_1	x_2	*Спам?*	$4x_1 + 4x_2$
1	1	1	1	8
2	0	0	0	0
3	1	0	0	4
4	0	1	0	4

Таблица 1. Небольшой обучающий набор для фильтра SpamAssassin. В столбцах x_1 и x_2 приведены результаты применения критериев к каждому сообщению. В четвертом столбце указано, какие сообщения являются спамом. И из последнего столбца мы видим, что, сравнивая значение функции $4x_1 + 4x_2$ с 5, мы можем отделить спамные сообщения от хороших

Но, спросите вы, какое отношение все это имеет к обучению? Ведь это обычная математическая задача. Так-то оно так, но разве нельзя сказать, что SpamAssassin обучается распознавать почтовый спам на примерах и контрпримерах? И чем больше доступно обучающих данных, тем лучше SpamAssassin будет справляться с задачей. Идея о том, что качество решения возрастает с накоплением опыта, является главной для большинства, если не для всех форм машинного обучения. Мы будем использовать следующее общее определение: *машинным обучением называется систематическое обучение алгоритмов и систем, в результате которого их знания или качество работы возрастают по мере накопления опыта*. В случае фильтра SpamAssassin под «опытом», на котором он учится, понимается набор правильно размеченных обучающих данных, а под «качеством работы» – способность распознавать почтовый спам. На рис. 2 схематически показано место машинного обучения в задаче классификации почтового спама. В других задачах машинного обучения опыт может принимать иную форму, например исправление ошибок, вознаграждение в случае достижения определенной цели и т. д. Отметим также, что, как и при обучении человека, целью машинного обучения не обязательно является повышение качества решения определенной задачи; желаемым результатом может быть и расширение знаний.

Существует много полезных способов описать классификатор SpamAssassin математически. Если обозначить результат i-го критерия применительно к данному сообщению в виде x_i, где $x_i = 1$, если критерий удовлетворяется, и 0 в противном случае, а вес i-го критерия обозначить w_i, то итоговую оценку сообщения можно записать в виде $\sum_{i=1}^{n} w_i x_i$, отразив тот факт, что w_i дает вклад в сумму, только если $x_i = 1$, то есть если i-й критерий удовлетворяется. Если обозначить t пороговую величину, при превышении которой сообщение классифицируется как спам (в нашем примере 5), то «решающее правило» можно записать в виде $\sum_{i=1}^{n} w_i x_i > t$.

Отметим, что левая часть этого неравенства линейно зависит от переменных x_i, а это означает, что при увеличении какой-либо переменной x_i на некоторую величину δ сумма изменится на величину $(w_i\delta)$, не зависящую от значения x_i. Это уже не так, если зависимость от x_i квадратичная или вообще степенная с показателем степени, отличным от 1.

Описанную нотацию можно упростить, применив линейную алгебру. Обозначим $\mathbf{w}$ вектор весов $(w_1, ..., w_n)$, а $\mathbf{x}$ – вектор результатов вычисления критериев $(x_1, ..., x_n)$. Приведенное выше

неравенство можно записать с помощью скалярного произведения: $\mathbf{w}\cdot\mathbf{x} > t$. Если заменить неравенство равенством, $\mathbf{w}\cdot\mathbf{x} = t$, то получится «решающая граница», отделяющая спам от неспама. В силу линейности левой части решающей границей является плоскость в пространстве, натянутая на переменные x_i. Вектор $\mathbf{w}$ перпендикулярен этой плоскости и направлен в сторону спама. На рис. 1 это наглядно показано в случае двух переменных.

Иногда удобно еще упростить запись, введя дополнительную постоянную «переменную» $x_0 = 1$ с фиксированным весом $w_0 = -t$. Тогда получатся точка в расширенном пространстве данных $\mathbf{x}^\circ = (1, x_1, ..., x_n)$ и расширенный вектор весов $\mathbf{w}^\circ = (-t, w_1, ..., w_n)$. При этом решающее правило примет вид $\mathbf{w}^\circ\cdot\mathbf{x}^\circ > 0$, а уравнение решающей границы – $\mathbf{w}^\circ\cdot\mathbf{x}^\circ = 0$. В этих так называемых однородных координатах решающая граница проходит через начало расширенной системы координат ценой добавления лишнего измерения (отметим, однако, что на данные это не влияет, потому что все точки и «истинная» решающая граница расположены в плоскости $x_0 = 1$).

Замечание 1. SpamAssassin в математических обозначениях. В таких абзацах я буду напоминать о полезных концепциях и обозначениях. Если что-то окажется незнакомым, то для полного понимания изложенного в книге материала придется потратить некоторое время на изучение – обратитесь к другим книгам и сетевым ресурсам, например www.wikipedia.org или mathworld.wolfram.com.

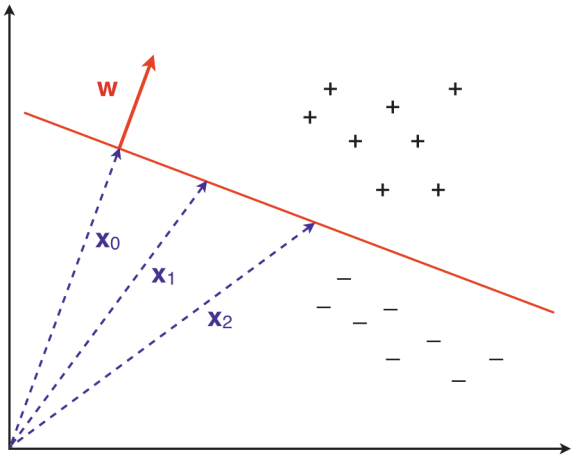

Рис. 1. Пример линейной классификации в двух измерениях. Прямая линия отделяет положительные результаты от отрицательных. Она определена уравнением $\mathbf{w}\cdot\mathbf{x}_i = t$, где $\mathbf{w}$ – вектор, перпендикулярный решающей границе и направленный в сторону положительных результатов, t – порог принятия решения, а $\mathbf{x}_i$ – вектор, оканчивающийся на решающей границе. Вектор $\mathbf{x}_0$ направлен туда же, куда и $\mathbf{w}$, откуда следует, что $\mathbf{w}\cdot\mathbf{x}_0 = \|\mathbf{w}\|\,\|\mathbf{x}_0\| = t$ ($\|\mathbf{x}\|$ обозначает длину вектора $\mathbf{x}$). Следовательно, решающую границу можно также описать уравнением $\mathbf{w}\cdot(\mathbf{x} - \mathbf{x}_0) = 0$, и иногда такая запись оказывается удобнее. В частности, из этого уравнения сразу видно, что положение решающей границы определяется только направлением $\mathbf{w}$, но не его длиной

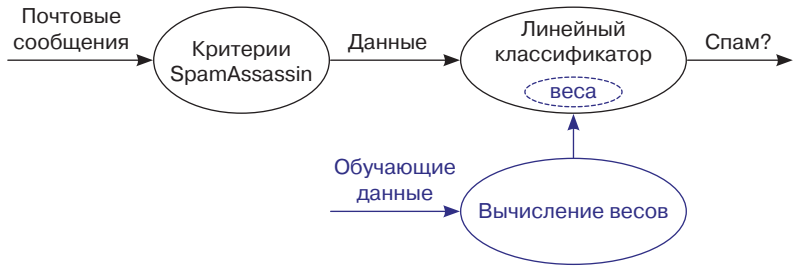

Рис. 2. В верхней части показано, как SpamAssassin подходит к задаче классификации почтового спама: текст каждого сообщения представляется точкой в пространстве данных, определяемой результатами вычисления встроенных критериев, а для решения «спам или неспам» применяется линейный классификатор. В нижней части показано то, что относится к машинному обучению

Мы уже видели, что у задачи машинного обучения, даже у такой простой, как в примере 1, может быть несколько решений. Тогда возникает вопрос: какое решение выбрать? И тут нужно понимать, что качество решения на обучающих данных нас не волнует – мы и так знаем, какие из предъявленных сообщений – спам! А волнует нас, как поведет себя классификатор на *будущих* сообщениях. На первый взгляд, получается порочный круг: чтобы узнать, правильно ли классифицировано сообщение, мне нужно знать его истинный класс, но если истинный класс известен, то классификатор уже не нужен. Однако важно помнить, что хорошее качество работы на обучающих данных – лишь средство к достижению цели, а не сама цель. На самом деле стремление добиться исключительно хорошего качества на обучающих данных легко может привести к удивительному и потенциально опасному явлению – *переобучению*.

Пример 2 (переобучение). Представьте, что вы готовитесь к экзамену по основам машинного обучения. По счастью, профессор Флах выложил в Сеть вопросы, которые задавались на предыдущем экзамене, и ответы на них. Вы начинаете отвечать на старые вопросы и сравнивать свои ответы с опубликованными. К сожалению, вы слишком увлеклись и тратите все свое время на запоминание ответов на старые вопросы. Если на предстоящем экзамене будут задаваться только старые вопросы, то все у вас сложится прекрасно. Но если материал останется тем же, а вопросы будут другими, то окажется, что ваша методика никуда не годится, и оценка будет гораздо ниже той, что вы заслужили бы при традиционной подготовке. В таком случае можно сказать, что вы переобучились на вопросах прошлых лет и приобретенные знания не обобщаются на вопросы будущего экзамена.

Обобщение – это, пожалуй, самая фундаментальная концепция машинного обучения. Если знания, которые SpamAssassin получил из предъявленных когда-то обучающих данных, переносятся – обобщаются – на ваши почтовые сообщения, вы довольны; если нет, вы начинаете искать более качественный фильтр

спама. Однако переобучение – не единственная возможная причина низкого качества работы на новых данных. Возможно, программисты SpamAssassin использовали обучающие данные, не репрезентативные для тех почтовых сообщений, которые приходят вам. К счастью, у этой проблемы есть решение: взять другие обучающие данные с такими же характеристиками, как у вашей почты. Машинное обучение – замечательная технология, позволяющая адаптировать поведение программы к конкретным обстоятельствам, и есть немало фильтров почтового спама, которые допускают обучение на пользовательских данных.

Таким образом, если существует несколько решений, то нужно выбрать то, которое не переобучено. Ниже мы обсудим, как это сделать, и даже разными способами. Ну а что сказать о противоположной ситуации, когда не существует ни одного решения, идеально классифицирующего обучающие данные? Представим, например, что сообщение 2 из примера 1, на котором оба критерия не удовлетворяются, – это спам, тогда не найдется ни одной прямой линии, разделяющей спам и неспам (можете убедиться в этом, изобразив все четыре сообщения в виде точек на плоскости, так что x_1 лежит на одной оси, а x_2 – на другой). В этом случае есть несколько вариантов действий. Первый – просто игнорировать это сообщение; возможно, оно нетипично или неправильно помечено (так называемый *шум*). Второй – взять более выразительный классификатор. Например, можно ввести второе решающее правило для фильтрации спама: в дополнение к условию $4x_1 + 4x_2 > 5$ добавить альтернативное условие $4x_1 + 4x_2 < 1$. Отметим, что при этом придется вычислить также другой порог и, возможно, другой вектор весов. Этот вариант можно рассматривать, только если обучающих данных достаточно для надежного вывода дополнительных параметров.

Линейная классификация в духе SpamAssassin может послужить полезным введением, но если бы она была единственным способом машинного обучения, то книга получилась бы куда тоньше. А что, если результатом обучения должны быть не только веса, но и сами критерии? Как решить, является ли отношение текстового материала к графическому хорошим критерием? Да и вообще, откуда этот критерий взялся? В этой области машинному обучению есть что предложить. Вероятно, вы уже заметили, что рассмотренные до сих пор критерии SpamAssassin не принимали во внимание *содержание* почтового сообщения. А ведь такие слова и словосочетания, как «виагра», «бесплатный iPod» или «подтвердите данные вашего счета», – верные индикаторы спама, тогда как другие – например, использование прозвища, известного только вашим друзьям, – свидетельствуют, что сообщение хорошее. Поэтому многие фильтры спама применяют методы классификации текста. Грубо говоря, хранят словарь слов и словосочетаний, являющихся признаками спама и неспама. Для каждого включенного в словарь элемента собирается статистика по обучающему набору. Пусть, например, слово «виагра» встретилось в четырех спамных сообщениях и одном хорошем. Если затем в новом сообщении нам попадется слово «виагра», то можно будет

заключить, что оно с вероятностью 0.80 (4:1) является спамом, а с вероятностью 0.20 – неспамом (см. основные понятия теории вероятностей в замечании 2).

Правда, ситуация несколько сложнее, чем кажется на первый взгляд, потому что мы должны принимать в расчет частоту спама. Предположим для определенности, что в среднем я получаю одно спамное сообщение на каждые шесть хороших (ах, если бы!). Это означает, что у следующего сообщения шанс оказаться спамным составляет 1:6, то есть не очень высокий, хотя и не пренебрежимо малый. Если затем выясняется, что это сообщение содержит слово «виагра», которое в спаме встречается в четыре раза чаще, чем в хороших сообщениях, то нужно каким-то образом учесть обе вероятности. Ниже мы узнаем о правиле Байеса, которое говорит, что их нужно просто перемножить: если умножить 1:6 на 4:1, то получится 4:6, то есть вероятность, что сообщение является спамом, равна 0.4. Иными словами, несмотря на наличие слова «виагра», сообщение, скорее всего, является хорошим. Но это же бред какой-то! Или не бред?

С вероятностями связаны «случайные величины», которые описывают исходы «событий». События часто гипотетические, и потому вероятности приходится оценивать. Взять, к примеру, утверждение «42% населения Великобритании одобряет премьер-министра». Единственный способ узнать, верно оно или неверно, состоит в том, чтобы задать вопрос каждому жителю Великобритании. Очевидно, это нереально. Вместо этого опрашивается некоторая выборка (хочется надеяться, репрезентативная), поэтому правильнее было бы сформулировать утверждение так: «42% опрошенной выборки из населения Великобритании одобряет премьер-министра». Отметим, что утверждения сформулированы в терминах процентной доли, или «относительной частоты»; в терминах вероятностей то же утверждение звучало бы так: «вероятность того, что случайно выбранный житель Великобритании одобряет премьер-министра, оценивается как 0.42». В данном случае событием является «данный случайно выбранный человек одобряет премьер-министра».

Условной вероятностью $P(A|B)$ называется вероятность того, что событие A произойдет, если известно, что событие B произошло. Например, возможно, что мужчины и женщины одобряют премьер-министра с разной частотой. Если обозначить $P(\text{РМ})$ вероятность того, что случайно выбранный человек одобряет премьер-министра, а $P(\text{РМ}|\text{женщина})$ – вероятность того, что случайно выбранная женщина одобряет премьер-министра, то $P(\text{РМ}|\text{женщина}) = P(\text{РМ,женщина})/P(\text{женщина})$, где $P(\text{РМ,женщина})$ – вероятность «совместного события», заключающегося в том, что случайно выбранный человек одобряет премьер-министра и одновременно является женщиной, а $P(\text{женщина})$ – вероятность того, что случайно выбранный человек является женщиной (то есть доля женщин в населении Великобритании).

Приведем еще два полезных тождества: $P(A,B) = P(A|B)P(B) = P(B|A)P(A)$ и $P(A|B) = P(B|A)P(A)/P(B)$. Последнее, называемое «правилом Байеса», будет играть важную роль в этой книге. Отметим, что многие из этих тождеств обобщаются на случай, когда случайных величин больше двух, например «цепное правило исчисления вероятностей»: $P(A,B,C,D) = P(A|B,C,D)P(B|C,D)P(C|D)P(D)$.

Два события A и B называются независимыми, если $P(A|B) = P(A)$, то есть от того, что мы знаем, что B произошло, вероятность A не изменяется. Эквивалентная формулировка: $P(A,B) = P(A)P(B)$. В общем случае перемножение вероятностей основывается на предположении о независимости соответствующих событий.

«Шанс» события – это отношение вероятности того, что событие произойдет, к вероятности того, что оно не произойдет. Иначе говоря, если вероятность некоторого события равна p, то его шанс равен $o = p/(1-p)$. И наоборот, $p = o/(o+1)$. Таким образом, вероятность 0.8 соответствует шансу 4:1, а противоположному шансу 1:4 соответствует вероятность 0.2. Если же событие мо-

жет с равным успехом как произойти, так и не произойти, то его вероятность равна 0.5, а шанс – 1:1. И хотя по большей части мы будем использовать вероятностную нотацию, шансы иногда удобнее, потому что выражаются в виде отношения.

Замечание 2. Основные понятия теории вероятностей

В применении к рассматриваемой задаче нужно ясно понимать, что имеются два независимых свидетельства: частота спама и вхождение слова «виагра». Они действуют в противоположных направлениях, и потому важно оценить их относительную силу. Числа говорят нам, что для преодоления того факта, что спам – относительно редкое явление, необходимо, чтобы шанс был не ниже 6:1. Шанс появления слова «виагра» оценивается как 4:1, этого недостаточно для перевеса в сторону спама, то есть мы не можем заключить, что сообщение действительно является спамом. Наличие слова «виагра» позволяет лишь сказать, что утверждение «это сообщение хорошее» стало гораздо менее вероятным, поскольку его вероятность снизилась с 6/7 = 0.86 до 6/10 = 0.60.

Схема «байесовской» классификации хороша тем, что ее можно повторить при наличии дополнительных свидетельств. Пусть, например, шансы в пользу спама при наличии словосочетания «голубая таблетка» оцениваются как 3:1 (то есть среди сообщений, содержащих это словосочетание, спама в три раза больше, чем неспама), и еще предположим, что наше сообщение содержит как «виагра», так и «голубая таблетка». Тогда произведение шансов 4:1 и 3:1 дает 12:1, и этого вполне достаточно, чтобы перевесить шанс 1:6, ассоциированный с низкой частотой спама (результирующий шанс равен 2:1, то есть вероятность спама повысилась до 0.67 против 0.40 без «голубой таблетки»).

Из того, что нам нет необходимости оценивать совместные вероятности событий, следует, что можно ввести в рассмотрение большее число переменных. На самом деле словарь типичного байесовского фильтра спама или классификатора текстов может содержать порядка 10 000 терминов[1]. Поэтому, вместо того чтобы вручную составлять небольшой набор «признаков», которые эксперты считают релевантными, или обладающими предсказательной силой, мы включаем куда больший набор и поручаем классификатору определить, какие признаки важны и в каких сочетаниях.

Следует отметить, что, перемножая шансы «виагры» и «голубой таблетки», мы неявно предполагаем, что соответствующие события независимы. Очевидно, что это не так: зная, что сообщение содержит словосочетание «голубая таблетка», мы не удивимся, встретив также и «виагру». В терминах вероятностей это формулируется следующим образом:

---

[1] В действительности словосочетания, содержащие несколько слов, обычно разлагаются на отдельные слова, то есть P(голубая таблетка) оценивается как P(голубая) P(таблетка).

- вероятность P(виагра|голубая таблетка) близка к 1;
- следовательно, совместная вероятность P(виагра, голубая таблетка) близка к P(голубая таблетка);
- следовательно, шанс сообщения оказаться спамом вследствие вхождения слов «виагра» и «голубая таблетка» мало отличается от шанса оказаться спамом вследствие вхождения одной лишь «голубой таблетки».

Иначе говоря, перемножая эти шансы, мы дважды учитываем одну и ту же информацию. Результирующее произведение 12:1 почти наверняка является завышенной оценкой, а истинный шанс вряд ли будет больше, чем 5:1.

Похоже, мы загнали себя в угол. Чтобы избежать завышения оценки, мы должны принимать во внимание совместное вхождение словосочетаний, но с вычислительной точки зрения задача практически разрешима, только если считать их независимыми.

Кажется, в действительности нам нужно нечто, близкое к такой основанной на правилах модели:

1. Если почтовое сообщение содержит слово «виагра», то его шанс оказаться спамом оценивается как 4:1.
2. Иначе, если оно содержит словосочетание «голубая таблетка», то его шанс оказаться спамом оценивается как 3:1.
3. Иначе его шанс оказаться спамом оценивается как 1:6.

Под первое правило подпадают все сообщения, содержащие слово «виагра» вне зависимости от того, встречается в них словосочетание «голубая таблетка» или нет, поэтому завышения оценки не происходит. Под второе правило подпадают *только* сообщения, содержащие словосочетание «голубая таблетка» без слова «виагра», – это гарантируется союзом «иначе». И под третье правило подпадают все остальные сообщения: не содержащие ни «виагры», ни «голубой таблетки».

Существо таких классификаторов, основанных на правилах, состоит в том, что сообщения рассматриваются не единообразно, а индивидуально. В каждом случае выделяются только наиболее релевантные признаки. Случаи можно определить с помощью нескольких вложенных признаков.

1. Сообщение содержит слово «виагра»?
   (a) если да: сообщение содержит словосочетание «голубая таблетка»?
      i. если да: оценить шанс спама как 5:1.
      ii. если нет: оценить шанс спама как 4:1.
   (b) если нет: сообщение содержит слово «лотерея»?
      i. если да: оценить шанс спама как 3:1.
      ii. если нет: оценить шанс спама как 1:6.

Эти четыре случая характеризуются логическими условиями вида «сообщение содержит слово “виагра”, но не содержит словосочетание “голубая таблетка”». Существуют эффективные алгоритмы выявления комбинаций признаков, обладающих наибольшей предсказательной силой, и организации их в виде правил или деревьев. Мы познакомимся с ними ниже.