Глава 4

Байесовское обучение

§ 4.1. Введение

До сих пор мы старательно избегали даже упоминания слова «вероятность» (разве что в контексте сэмплинга из какой-то выборки с заданной вероятностью — но это ведь не всерьёз). Не то чтобы мы старались быть понятными людям, которые не знают этого слова — сейчас им всё равно придётся вернуться к прочно забытому теорверу и напомнить самим себе хотя бы самые базовые основы этой довольно хитрой науки. На самом деле просто те методы, которые мы до сих пор рассматривали, могли иметь успех и без обращения к теории вероятностей. В этой главе мы переходим к рассмотрению так называемых *байесовских методов* обучения — методов, для которых теория вероятностей играет не вспомогательную, а основополагающую роль. Как понятно из названия, для этих методов важнейшим инструментом станет *теорема Байеса*, которую мы напомним в следующем параграфе.

§ 4.2. Основные понятия теории вероятностей

Мы не хотели бы оскорблять читателя, предполагая, что он может быть не знаком с азами дискретной теории вероятностей. Однако на случай, если познания читателя в этой области были приобретены слишком давно и уже не входят в активный, мы напомним основные определения и свойства теории вероятностей.

Мы не будем подробно останавливаться здесь на понятии вероятности, хотя само оно является предметом долгих и по-своему интересных дискуссий, в том числе в рамках искусственного интеллекта. Дело в том, что классический подход к определению вероятностей — *частотный* (frequentist), в котором участвуют доверительные интервалы, а вероятность (в простейшем случае) мыслится как предел отношения количества удачных экспериментов к общему количеству экспериментов. Но есть и другой, *байесовский* (Bayesian) подход, в котором вероятности могут быть также субъективными, т.е. выражать не столько частотные соотношения, сколько *степень уверенности* (degree of belief) эксперта в том или ином утверждении. Эта фундаментальная разница, впрочем, не окажет ни малейшего влияния на наши дальнейшие рассуждения.

Итак, мы считаем, что понятие вероятности определено, и мы можем говорить о вероятности $p(h)$ утверждения h (h здесь обозначает hypothesis — нам обычно потребуется говорить о вероятностях гипотез). Кроме того, мы не будем подробно останавливаться на понятиях *совместной вероятности* $p(x_1 \dots x_n)$ и *условной вероятности* $p(x|y)$.

Нашим основным инструментом станет *теорема Байеса* — несложное следствие формул для вероятности конъюнкции:[1]

$$p(x \wedge y) = p(x|y)p(y) = p(y|x)p(x).$$

Отсюда следует, что если $p(y) \neq 0$, то

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

Это и есть теорема Байеса.

§ 4.3. Теорема Байеса, данные и гипотезы

В предыдущих главах мы рассматривали большое количество разнообразных алгоритмов машинного обучения. Однако в их структуре было очень много общего. Это общее мы сейчас выделим и поймём, что на самом деле все рассматривавшиеся нами алгоритмы пытались сделать одно и то же.

Во-первых, у каждого алгоритма было некоторое *множество гипотез*, из которых он пытался выбрать наилучшую. Например, для алгоритма обучения концептам Find-S множеством гипотез было множество правил вида «из такого-то набора атрибутов следует, что целевая функция равна 1». У алгоритма ID3 множеством гипотез было множество возможных деревьев принятия решений или, что для нас сейчас эквивалентно, множество всевозможных дизъюнкций гипотез алгоритма Find-S.[2] У нейронных сетей множество возможных гипотез было более богатым — для сетей с

[1] Томас Байес (Thomas Bayes, 1702–1761) — удивительный пример человека, который почти не публиковался и был долгое время не очень известен, но чьё имя осталось в веках в бесчисленных определениях с прилагательным «байесовский», появившихся в последние полвека. Преуспевающий пресвитерианский священник за всю жизнь опубликовал только два труда: «Благость господня, или попытка доказать, что конечной целью божественного провидения и направления является счастье его созданий» и анонимно опубликованное «Введение в теорию флюксий, или в защиту математиков от нападок автора “Комментатора”», где Байес защищал ньютоновский анализ от нападок Беркли. Его работа по теории вероятностей вышла уже после смерти, в 1763 году, и в ней Байес ответил на один из вопросов, оставленных открытым основателем теории вероятностей де Муавром. Впрочем, стоит подчеркнуть, что теорема Байеса — это для нас несложное следствие очевидных свойств вероятности; во времена Байеса и де Муавра эти свойства ещё не были столь чётко сформулированы, и само понятие вероятности ещё не было толком разработано.

[2] На самом деле деревья — более экономичный способ записи, чем набор дизъюнкций, но нам это сейчас не важно.

линейными нейронами это было множество линейных форм в пространстве, размерность которого зависела от количества входов, а для сетей с нелинейными нейронами — множество функций определённого вида. Общий вид гипотез для генетических алгоритмов мы подробно обсуждали, когда старались закодировать их наиболее эффективным путём в § 3.4. В общем, у каждого алгоритма есть множество гипотез, на котором он ведёт свой поиск.

Второй слон, на котором покоится планета машинного обучения — *данные*, на основании которых алгоритмы принимают свои решения. Здесь уже такого разнообразия не наблюдается: фактически, любой алгоритм готов принять любые данные, нужно только отформатировать их подходящим образом. Важно, что гипотезу выбирают так, чтобы она максимально хорошо подходила под данные. Алгоритмы классификации вроде Find-S и ID3 вообще выбирают гипотезу так, чтобы она идеально подходила под данные. Нейронные сети стараются минимизировать среднеквадратичную ошибку — в случае вещественнозначных функций идеального соответствия данным ожидать сложно[3]. Генетические алгоритмы не могут гарантировать идеального соответствия, но стараются максимизировать функцию Fitness, которая обычно напрямую зависит от того, насколько удачно описываются имеющиеся данные.

Эти два компонента в самом общем их виде — всё, что нам сейчас потребуется. Давайте предположим, что у нас есть некоторое множество гипотез H и множество имеющихся данных D. Тогда цель любого алгоритма машинного обучения — найти гипотезу, которая была бы *наиболее вероятной* при имеющихся данных. Действительно, такое поведение было бы оптимальным — ничего более хорошего ожидать невозможно. Математически это можно записать так:

$$h = \operatorname{argmax}_{h \in H} p(h|D).$$

Такая гипотеза называется *максимальной апостериорной гипотезой* (maximum *a posteriori* hypothesis, MAP).

Давайте перепишем это по теореме Байеса:

$$h = \operatorname{argmax}_{h \in H} p(h|D) = \operatorname{argmax}_{h \in H} \frac{p(D|h)p(h)}{p(D)} = \operatorname{argmax}_{h \in H} p(D|h)p(h),$$

потому что $p(D)$ от h не зависит.

Часто предполагают, что гипотезы изначально равновероятны: $p(h_i) = p(h_j)$ для всех гипотез $h_i, h_j \in H$. Тогда предыдущее выражение мжоно переписать ещё проще:

$$h = \operatorname{argmax}_{h \in H} p(D|h).$$

[3] Более того, в случае вещественнозначных функций довольно проблематично даже определить, что такое «идеальное соответствие данным», особенно если речь идёт о компьютерных, т.е. априори дискретных и конечных, вычислениях.

Эти выражения — суть поиска оптимальной гипотезы. По ним можно сразу построить алгоритм поиска максимальной апостериорной гипотезы: нужно всего лишь для каждой гипотезы $h \in H$ вычислить её апостериорную вероятность $p(h|D)$, а затем выбрать ту из них, для которой эта вероятность максимальна. Разумеется, мы не рекомендуем применять такой алгоритм на практике: в любой хоть немного реалистичной задаче размер множества гипотез сравним с количеством элементарных частиц во Вселенной.[4]

Но, хотя по ним и можно построить алгоритм, основная функция выражений для поиска максимальной апостериорной гипотезы не прямо практическая. MAP-гипотеза нужна для того, чтобы сравнивать с ней другие алгоритмы и выяснять, когда они работают оптимально. Пример такого подхода мы приведём в следующем параграфе.

§ 4.4. MAP и задачи классификации

Для того чтобы найти максимальную апостериорную гипотезу, нужно научиться вычислять $p(h)$ и $p(D|h)$. Пусть выполняются следующие условия:

— в D нет шума (т.е. все тестовые примеры содержат правильные ответы);

— целевая функция c содержится среди гипотез H;

— нет априорных причин верить, что одна из гипотез более вероятна, чем другая.

Эти предположения весьма обычны для задач классификации. Единственное, что может омрачить счастье, происходящее из таких удобных предположений, — это возможный шум в данных, которого мы уже касались, обсуждая проблему оверфиттинга в § 1.8. Но мы пока закроем глаза на эту проблему — всё равно большинство алгоритмов классификации, нами рассмотренных, справиться с ней не могут.

Из третьего условия следует:

$$p(h) = \frac{1}{|H|} \text{ для всех } h \in H.$$

Условная вероятность $p(D|h)$ — это вероятность наблюдать значения целевых функций $\langle t_1, \ldots, t_m \rangle$ для фиксированного набора входных данных $\langle d_1, \ldots, d_m \rangle$ при условии, что выполняется гипотеза h. Поскольку шума нет, $p(t_i|h) = 1$, если $t_i = h(d_i)$, и 0 в противном случае. Итого:

$$p(D|h) = \begin{cases} 1, & \text{если } d_i = h(x_i) \text{ для всех } d_i \in D, \\ 0, & \text{в противном случае.} \end{cases}$$

Для полноты картины, хотя это и не нужно для определения MAP-гипотезы, давайте подсчитаем вероятность $p(D)$. Пусть $\mathrm{Cons}(D)$ — множество гипотез $h \in H$,

[4] Физики оценивают это количество как число порядка 2^{100}; т.е. если гипотезами могут быть строки из 100 бит (не так уж и много, правда?), размер множества гипотез как раз будет порядка количества частиц во Вселенной.

совместимых с D. Тогда

$$p(D) = \sum_{h \in H} p(D|h)p(h) = \sum_{h \in Cons(D)} \frac{1}{|H|} = \frac{|Cons(d)|}{|H|}.$$

Итого получается:

$$p(h|D) = \frac{p(D|h)p(h)}{p(D)} = \begin{cases} \frac{1}{|Cons(d)|}, & \text{если } d_i = h(x_i) \text{ для всех } d_i \in D, \\ 0, & \text{в противном случае.} \end{cases}$$

Иными словами, каждая гипотеза, совместная со всеми данными — это максимальная апостериорная гипотеза. О чём это нам говорит? О том, что все алгоритмы, которые выдают гипотезы, совместные со всеми входящими данными, *в вышеописанных предположениях* выдают максимальные апостериорные гипотезы. Это значит, что они работают идеально — ничего более хорошего сделать принципиально невозможно.

В данном случае мы байесовскими методами пришли к доказательству того, что какие-то другие алгоритмы работают оптимально. Это одна из важнейших функций байесовских методов — это тот редкий в искусственном интеллекте случай, когда что-то действительно можно (и совсем несложно, как мы уже убедились) математически доказать.

Впрочем, результат, который мы получили, даёт ещё больше информации, чем простое подтверждение того, что тот или иной алгоритм решает поставленную задачу. Мы получили конкретный, чётко математически определённый набор предположений, при которых алгоритм классификации работает оптимальным образом. Это уже напрямую даёт программу действий: если найденные предположения выполняются, то, значит, можно пользоваться имеющимся алгоритмом, не желая ничего лучшего. А вот если какие-то условия не выполняются, то можно искать (это, правда, ещё не значит найти) алгоритмы, которые работали бы лучше. Более того, в таком случае мы будем знать, какие именно предположения не выполняются, и это, скорее всего, поможет разработать алгоритм, который будет ближе к оптимальному.

Позднее мы приведём другие примеры аналогичного применения байесовских методов, а пока обратимся к более практическим вопросам — построению байесовских классификаторов.

§ 4.5. Оптимальный байесовский классификатор и классификатор Гиббса

До сих пор мы отвечали на вопрос: «Какова наиболее вероятная гипотеза при имеющихся данных?» Теперь пора ответить на вопрос: «Какова наиболее вероятная классификация нового примера при имеющихся данных?».

Казалось бы, можно просто применить максимальную апостериорную гипотезу. Однако этот метод не всегда приводит к оптимальным результатам.

Пример 4.1. Когда не нужно применять MAP.

Пусть множество гипотез состоит из четырёх элементов, и их апостериорные вероятности равны 0.2, 0.2, 0.2 и 0.4 соответственно. Четвёртая гипотеза — максимальная апостериорная. Но если новый пример классифицируется первыми тремя гипотезами положительно, а четвёртой — отрицательно, то общая вероятность его положительной классификации 0.6, и применять MAP было бы неправильно.

Пусть имеются данные D и множество гипотез h. Для вновь поступившего примера x нужно выбрать такое значение v, чтобы максимизировать $p(v|D)$. Иными словами, наша задача — найти

$$\operatorname{argmax}_{v \in V} \sum_{h \in H} p(v|h)p(h|D).$$

Определение 4.1. *Любой алгоритм, который решает задачу*

$$\operatorname{argmax}_{v \in V} \sum_{h \in H} p(v|h)p(h|D),$$

называется оптимальным байесовским классификатором (*optimal Bayes classifier*).

Пример 4.2. Продолжение примера 4.1.

В примере 4.1 мы рассматривали четыре гипотезы h_i, $i = 1..4$ со множеством значений $V = \{0, 1\}$. Вероятности распределялись следующим образом:

$$\begin{array}{ll} p(h_1|D) = p(h_2|D) = p(h_3|D) = 0.2, & p(h_4|D) = 0.4, \\ p(x = 1|h_1) = p(x = 1|h_2) = p(x = 1|h_3) = 1, & p(x = 1|h_4) = 0, \\ p(x = 0|h_1) = p(x = 0|h_2) = p(x = 0|h_3) = 0, & p(x = 0|h_4) = 1. \end{array}$$

Тогда

$$\sum_i p(x = 1|h_i)p(h_i|D) = 0.6, \quad \sum_i p(x = 0|h_i)p(h_i|D) = 0.4,$$

и оптимальный классификатор будет классифицировать этот пример как положительный, а не как отрицательный, хотя так рекомендует поступить MAP-гипотеза.

Отметим, что оптимальный классификатор действительно оптимален: никакой другой метод не может в среднем превзойти его. Он может даже классифицировать данные по гипотезам, не содержащимся в H; в частности, он может классифицировать по любому элементу линейной оболочки H.

Но, с другой стороны, оптимальный байесовский классификатор обычно не получается эффективно реализовать — нужно перебирать все гипотезы, а всех гипотез очень много. Поэтому приходится искать возможности ускорить этот процесс.

В оптимальном классификаторе мы должны взять взвешенную сумму условных вероятностей возможных ответов на новый пример при условии гипотез по всем гипотезам. Это выражение до известной степени напоминает выражение для определения математического ожидания. Поэтому напрашивается вполне логичное ускорение алгоритма: вместо суммы по всем гипотезам $\sum_{h\in H} p(v|h)p(h|D)$ можно рассмотреть случайную гипотезу, взятую с распределением $p(h|D)$. Такой метод называется *классификатором Гиббса.*[5]

Итого алгоритм получается довольно простой:

1. Выбрать случайную гипотезу $h \in H$ согласно распределению их апостериорных вероятностей.
2. Классифицировать новый случай x согласно h.

Мы здесь оставляем за кадром подробности того, как сэмплировать с заданными вероятностями — это может зависеть от конкретного множества гипотез. Но обычно это получается реализовать гораздо более эффективно, чем сумму по всем гипотезам. А главное преимущество этого метода в том, что ошибка алгоритма Гиббса при определённых не слишком жёстких условиях лишь вдвое больше ошибки оптимального классификатора. Поэтому он обычно оказывается предпочтительнее.

§ 4.6. Наивный байесовский классификатор

Оптимальный классификатор и гиббсовский классификатор оставляли за кадром проблему того, как искать апостериорные вероятности гипотез. Наивный байесовский классификатор[6] предлагает свой ответ на этот вопрос. Его предположения в высшей степени наивны, настолько, что даже не верится, что он может хорошо работать. Однако практика (и теория тоже, хотя теоретические доказательства его эффективности не входят сейчас в нашу задачу) показывает, что наивный байесовский классификатор оказывается весьма эффективен для задач классификации с большим количеством параметров, таких, как классификация текстов, например.

Постановка задачи ничем не отличается от любой другой задачи классификации: каждый пример x принимает значения из некоторого множества V и описывается

[5] Джосайя Уиллард Гиббс (Josiah Willard Gibbs, 1839–1903) — знаменитый американский физик, который фактически создал теоретические основы химической термодинамики; достаточно неожиданно увидеть его имя в связи с этими алгоритмами, к которым он не имел ни малейшего отношения. На самом деле это имя попало сюда транзитом через *сэмплирование по Гиббсу* (Gibbs sampling), которое так называется потому, что это сэмплирование напоминает статистическую физику. Кстати, сэмплирование по Гиббсу тоже не Гиббс придумал — алгоритм был разработан через восемьдесят лет после его смерти.

[6] Иногда его даже называют Idiot's Bayes, но мы не рискнём употреблять этот термин и переводить его на русский язык.

атрибутами $\langle a_1, a_2, \ldots, a_n \rangle$. Требуется найти наиболее вероятное значение данного атрибута, т.е.

$$v_0 = \operatorname{argmax}_{v \in V} p(x = v | a_1, a_2, \ldots, a_n).$$

По теореме Байеса,

$$v_{MAP} = \operatorname{argmax}_{v \in V} \frac{p(a_1, a_2, \ldots, a_n | x = v) p(x = v)}{p(a_1, a_2, \ldots, a_n)} =$$

$$= \operatorname{argmax}_{v \in V} p(a_1, a_2, \ldots, a_n | x = v) p(x = v).$$

Оценить $p(x = v)$ легко: при наличии даже не слишком большого числа тестовых примеров можно оценить частоту встречаемости каждого из значений x. Но оценить разные $p(a_1, a_2, \ldots, a_n | x = v)$ не представляется возможным — их слишком много. Для того чтобы получить оценки этих вероятностей, нам фактически нужно каждую из возможных комбинаций атрибутов пронаблюдать по несколько раз. Это, разумеется, на практике невозможно.

Поэтому наивный байесовский классификатор предполагает условную независимость атрибутов при условии данного значения целевой функции. Иначе говоря:

$$p(a_1, a_2, \ldots, a_n | x = v) = p(a_1 | x = v) p(a_2 | x = v) \ldots p(a_n | x = v).$$

Теперь уже даже относительно небольшое количество тестовых примеров достаточно для того, чтобы достаточно надёжно оценить каждую из этих вероятностей.

Итак, наивный байесовский классификатор выбирает v как

$$v_{NB}(a_1, a_2, \ldots, a_n) = \operatorname{argmax}_{v \in V} p(x = v) \prod_{i=1}^{n} p(a_i | x = v).$$

В практической задаче распознавания текстов (например, при создании спам-фильтра) атрибутами будет появление того или иного слова в тексте. Соответственно, количество атрибутов получается совершенно запредельным для прямого перебора: их по меньшей мере несколько десятков тысяч. Более того, если строить полную модель текста, то нужно учитывать местоположение слов друг относительно друга, что умножает количество атрибутов на длину документа. Поэтому какой-то метод, устанавливающий условные независимости атрибутов друг относительно друга, совершенно необходим. Но наивный байесовский классификатор предполагает какие-то совершенно фантастические вещи: получается, что появление слова не зависит от других слов, его окружающих, и даже, более того, не зависит от длины документа. И, тем не менее, наивный байесовский классификатор оказывается на редкость эффективен.

§ 4.7. Байесовское обучение и нейронные сети

В § 4.4 мы уже касались вопроса о том, что байесовское обучение часто оказывается стандартом, идеалом, с которым следует сравнивать другие методы обучения и выяснять, насколько тот или иной алгоритм приближается к поиску максимальной апостериорной гипотезы. В § 4.4 было доказано, что алгоритмы классификации, рассмотренные нами в Главе 1, работают оптимальным образом. Теперь настала пора отдать старый долг: в Главе 2 мы строили алгоритмы с целью минимизировать среднеквадратичное отклонение от целевой функции и упоминали, что позже расскажем, почему нужно минимизировать именно такую меру ошибки. Сейчас настала пора сделать это.

Как читателям уже, наверное, понятно, без чётких предположений доказать обычно ничего не получается, тем более в вероятностном контексте. Не обойдётся без них и здесь. Мы пытаемся приблизить нейронной сетью целевую функцию $f : \mathcal{X} \longrightarrow \mathbb{R}$, имея набор тестовых примеров

$$D = \{\langle x_1, t_1\rangle, \ldots, \langle x_m, t_m\rangle\}.$$

Совершенно очевидно, что в условиях приближения вещественнозначных функций не приходится ожидать, что данные точно лягут на график целевой функции. Наше главное предположение касается распределения шума, т.е. отклонения тестовых примеров от идеального графика f.

Мы предполагаем, что $t_i = f(x_i) + e_i$, где e_i — *равномерно распределённый шум с нулевым средним*. Кроме того, мы предполагаем независимость тестовых примеров.

Теперь несложные математические преобразования приведут нас к цели. Мы ищем

$$h_{MAP} = \operatorname{argmax}_H p(D|h) = \operatorname{argmax}_H \prod_{i=1}^{m} p(t_i|h).$$

$p(t_i|h)$ — нормальное распределение с вариацией σ и с центром в $\mu = f(x_i)$. Поскольку нас интересует условная вероятность при условии того, что гипотеза h верна, то в данной ситуации $f(x_i) = h(x_i)$. Итого получаем (произведение функций плотности получается потому, что тестовые примеры предполагаются независимыми):

$$h_{MAP} = \operatorname{argmax}_H \prod_{i=1}^{m} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2\sigma^2}(d_i - h(x_i))^2} =$$

$$= \operatorname{argmax}_H \sum_{i=1}^{m} \left(\ln \frac{1}{\sqrt{2\pi\sigma^2}} - \frac{1}{2\sigma^2}(d_i - h(x_i))^2 \right) =$$

$$= \operatorname{argmin}_{H} \sum_{i=1}^{m} (d_i - h(x_i))^2.$$

Таким образом, мы доказали, что минимизация среднеквадратичной ошибки ведёт к получению MAP-гипотезы. Это фактически оправдывает те алгоритмы обучения нейронных сетей, которые мы разработали в Главе 2.