

Педро Домингос. Верховный алгоритм

Машинное обучение преобразует науку, технологию, бизнес и позволяет глубже узнать природу и человеческое поведение. Программирующие сами себя компьютеры — одна из самых важных современных технологий, и она же — одна из самых таинственных. Ученый-практик Педро Домингос приоткрывает завесу и впервые доступно рассказывает о машинном обучении и о поиске универсального обучающегося алгоритма, который сможет выуживать любые знания из данных и решать любые задачи. Чтобы заглянуть в будущее и узнать, как машинное обучение изменит наш мир, не нужно специального технического образования — достаточно прочитать эту книгу. См. также. [Джон Форман. Много цифр: Анализ больших данных при помощи Excel](#), [Идеи Байеса для менеджеров](#).

Педро Домингос. Верховный алгоритм. Как машинное обучение изменит наш мир. — М.: Манн, Иванов и Фербер, 2016. — 336 с.



Купить цифровую книгу в [ЛитРес](#), бумажную книгу в [Ozon](#) или [Лабиринте](#)

Величайшая задача науки — объяснить, как можно больше экспериментальных фактов логической дедукцией, исходящей из как можно меньшего количества гипотез и аксиом.
Альберт Эйнштейн

Прогресс цивилизации заключается в увеличении количества важных действий, которые мы выполняем не думая.
Альфред Норт Уайтхед

Пролог. «Как мы учимся?», «Можно ли учиться эффективнее?», «Что мы способны предсказать?», «Можно ли доверять полученному знанию?» Соперничающие школы машинного обучения отвечают на эти вопросы по-разному. Всего существует пять основных научных течений, каждому из которых мы посвятим отдельную главу. У каждого из пяти «племен» машинного обучения есть собственный универсальный обучающийся алгоритм. Хотелось бы, чтобы все их черты слились воедино в окончательном, совершенном Верховном алгоритме. Верховный алгоритм сумеет извлечь из данных вообще все знание — знание прошлого, настоящего и будущего.

ГЛАВА 1. РЕВОЛЮЦИЯ МАШИННОГО ОБУЧЕНИЯ

Клод Шеннон, признанный отец теории информации, первым осознал, что включение и выключение транзисторов в ответ на действия других транзисторов — это, в сущности, логический вывод (подробнее см. [Джеймс Глик. Информация. История. Теория. Поток](#)). Любой алгоритм, как бы сложен он ни был, сводится всего к трем операциям: И, ИЛИ и НЕ (см. [Булева логика: критерии И, ИЛИ](#)).

Алгоритмы предъявляют строгие требования: часто говорят, что по-настоящему понимаешь что-то только тогда, когда можешь выразить это в виде алгоритма (как заметил Ричард Фейнман, «я не понимаю того, чего не могу создать»); см. [Леонард Млодинов. Радуга Фейнмана. Поиск красоты в физике и в жизни](#)).

Многие компьютерщики, особенно старшего поколения, понимают машинное обучение не так хорошо, как им хотелось бы. Дело в том, что компьютерные науки традиционно следовали в русле детерминизма, а в машинном обучении нужно мыслить в категориях статистики. Если какое-то правило, скажем, отмечать определенные письма как спам, срабатывает в 99, а не в 100% случаев, это не значит, что в нем какая-то ошибка: может быть, это лучшее, что можно сделать, и даже такая точность очень полезна. Различия в стиле мышления во многом послужили причиной, по которой Microsoft оказалось намного сложнее нагнать Google, чем в свое время Netscape. В конце концов, браузер всего лишь стандартная программа, а вот поисковая система требует другого склада ума.

Сегодня очевидно, что переход от компьютеров к интернету, а затем к машинному обучению был неизбежен. Компьютеры сделали возможным интернет, тот породил поток данных и проблему безграничного выбора, а машинное обучение использует потоки данных, чтобы решить проблему безграничного выбора. У Netflix может быть хоть сто тысяч разных DVD-дисков, но, если клиент не знает, как найти то, что ему понравится, он будет по умолчанию выбирать хиты. И только когда Netflix обзавелся обучающимся алгоритмом, который угадывает ваши вкусы и советует музыку, длинный хвост менее популярных исполнителей «взлетел» (подробнее см. [Крис Андерсон. Длинный хвост. Эффективная модель бизнеса в Интернете](#)).

ГЛАВА 2. ВЛАСТЕЛИН АЛГОРИТМОВ

За огромным большинством приложений машинного обучения стоят всего несколько алгоритмов. Посмотрите, например, на наивный байесовский классификатор (подробнее см. [Наивный байесовский классификатор документов в Excel](#)). Если взять базу данных из историй болезни — симптомы, результаты анализов, наличие или отсутствие сопутствующих заболеваний, — этот алгоритм может научиться диагностировать болезнь в долю секунды, и часто лучше, чем врачи, которые много лет провели в медицинском институте. При этом тот же самый алгоритм широко используется для фильтрации спама.

Другой простой обучающийся алгоритм, так называемый метод ближайших соседей, используют для массы задач — от распознавания почерка до управления манипуляторами в робототехнике и отбора книг и фильмов, которые могут понравиться клиенту (подробнее см. [Кластерный анализ: сетевые графы и определение сообществ](#)).

Центральная гипотеза этой книги: «Все знание — прошлое, настоящее и будущее — можно извлечь из данных с помощью одного универсального обучающегося алгоритма». Я называю этот алгоритм Верховным. Если его создание оказалось бы возможным, это стало бы одним из величайших научных достижений за всю историю человечества.

Красивый пример того, как очень простая процедура итерации может породить неистощимое разнообразие форм, — множество Мандельброта (см. [Бенуа Мандельброт. \(Не\)послушные рынки: фрактальная революция в финансах](#)). Если горы, реки, облака и деревья — результат аналогичных процессов, а фрактальная геометрия показывает, что так оно и есть, возможно, эти процессы — просто разная параметризация одной-единственной процедуры, которую мы можем вывести на их основе.

Согласно одной из школ статистики, в основе всего обучения лежит одна простая формула, а именно теорема Байеса, которая определяет, как корректировать предположения при появлении новых доказательств (подробнее см. [Идеи Байеса для менеджеров](#)). Байесовский алгоритм начинает с набора гипотез о мире. Когда он видит новые данные, гипотезы, согласующиеся с ними, становятся более вероятными, а те, что с ним не согласуются, — менее вероятными.

Если бы вы отправились в начало XX века и рассказали, что вскоре будет изобретена машина, которая сможет решать проблемы во всех сферах человеческой деятельности — одна и та же машина для всех проблем, — никто бы не поверил. Вам бы объяснили, что машины могут делать что-то одно: сеялки не печатают, а пишущие машинки не сеют. Затем, в 1936 году, Алан Тьюринг придумал любопытное устройство с лентой и головкой, которая читает и пишет символы. Сегодня оно известно, как машина Тьюринга. С ее помощью может быть решена каждая проблема, какую только можно решить с помощью логической дедукции. Более того, так называемая универсальная машина Тьюринга может симулировать любую другую, прочтя с ленты ее спецификацию, — другими словами, ее можно запрограммировать делать что угодно (о Тьюринге см. [Эндрю Ходжес. Игра в имитацию](#), о его алгоритме — [Чарльз Петцольд. Читаем Тьюринга](#)).

К Верховному алгоритму скептически относятся столько же людей, сколько испытывают по поводу его существования энтузиазм. Марвин Минский, профессор Массачусетского технологического института скептически относится к машинному обучению. Минский яростно поддерживал проект «Сайк», самый известный провал в истории искусственного интеллекта. Целью «Сайка» было создание искусственного интеллекта путем ввода в компьютер всего необходимого знания. Проблема такого подхода к искусственному интеллекту — в том, что он не работает.

Другой выдающийся ученый, не верящий в машинное обучение, — лингвист Ноам Хомский (о взглядах Хомского много расскажет его политическое эссе [Как устроен мир](#)). Хомский уверен, что язык обязательно должен быть врожденным, потому что примеров грамматически правильных предложений, которые слышат дети, недостаточно, чтобы научиться грамматике. Однако это только перекладывает бремя обучения языку на эволюцию, и это аргумент не против Верховного алгоритма, а лишь против того, что он похож на головной мозг.

Если говорить более обобщенно, Хомский критически относится к статистическому обучению любого рода. В конце концов, практика — критерий истины. Статистические алгоритмы обучения языку работают, а построенные вручную языковые системы — нет. Обучающиеся алгоритмы теперь используются практически во всех распознавателях речи, включая [Siri](#).

«Как бы ни был умен алгоритм, всегда есть то, что он не может узнать». Это утверждение — самое частое возражение против машинного обучения. Нассим Талеб из всех сил напирал на него в своей книге [Черный лебедь. Под знаком непредсказуемости](#). Некоторые события просто непредсказуемы: если человек видел только белых лебедей, он будет считать, что вероятность когда-нибудь встретить черного равна нулю. Финансовый крах 2008 года оказался как раз таким «черным лебедем». Действительно, некоторые вещи можно предсказать, а некоторые нельзя, и отличать одно от другого — первейшая задача алгоритма машинного обучения. Обучающиеся алгоритмы вполне способны точно предсказать редкие, никогда до этого не происходившие события.

Еще одно схожее и часто повторяемое возражение: «Данные не могут заменить человеческой интуиции». На самом деле это человеческая интуиция не может заменить данных. Статистический анализ побеждает искателей талантов в бейсболе (это замечательно описано в книге Майкла Льюиса MoneyBall), он превосходит знатоков в дегустации вин, и каждый день мы видим все новые примеры его способностей.

Насколько сложен будет Верховный алгоритм? Тысячи строк кода? Миллионы? Мы пока не знаем. В известном эпизоде книги [Науки об искусственном](#) пионер искусственного интеллекта и нобелевский лауреат Герберт Саймон просит представить себе муравья, который упорно бежит по пляжу к себе домой. Путь муравьишки сложен не потому, что сложен он сам, а потому что вокруг полно маленьких дюн, на которые надо взбираться, и гальки, которую приходится обегать. Попытки смоделировать муравья, запрограммировав все возможные пути, будут обречены на провал. Аналогично самое сложное в машинном обучении — это данные. Все, что должен сделать Верховный алгоритм, — усвоить их, поэтому не надо удивляться, если сам он окажется несложным.

Как заметил Исая Берлин, некоторые мыслители подобны лисам и знают много разного, а некоторые — ежам, которые знают что-то одно, но важное (подробнее см. эссе «Еж и лиса» в книге [Исайя Берлин. История свободы. Россия](#)). То же самое с обучающимися алгоритмами. Я надеюсь, что Верховный алгоритм окажется ежом.

Поиски Верховного алгоритма сложны, но их оживляет соперничество пяти научных школ, действующих в области машинного обучения. Для *символистов* интеллект сводится к манипулированию символами — так математики решают уравнения, заменяя одни выражения другими. Их верховный алгоритм — это обратная дедукция: она определяет недостающее для дедукции знание, а затем как можно в большей степени его обобщает. Для *коннекционистов* обучение — то, чем занимается головной мозг, и поэтому они считают, что этот орган надо воспроизвести путем обратной инженерии. Верховный алгоритм коннекционистов — метод обратного распространения ошибки, который сравнивает выходные данные системы с желаемыми, а потом последовательно, слой за слоем, меняет соединения между нейронами, чтобы сделать результат ближе к тому, что требуется. *Эволюционисты* верят, что мать учения — естественный отбор. Их верховный алгоритм — генетическое программирование, соединяющее и развивающее компьютерные программы точно так же, как природа сводит и развивает живые организмы.

Байесовцы озабочены прежде всего неопределенностью. Как работать с зашумленной, неполной, даже противоречивой информацией? Выходом становится вероятностный вывод, а верховным алгоритмом — теорема Байеса и ее производные. Для *аналогистов* ключ к обучению — находить сходства между разными ситуациями и тем самым логически выводить другие сходства. Верховный алгоритм аналогистов — это метод опорных векторов, который определяет, какой опыт надо запомнить и как соединить опыт, чтобы делать новые прогнозы. Специалисты по машинному обучению, как и все ученые, напоминают слепцов рядом со слонем. Наша цель — соединить все фрагменты в одно решение.

ГЛАВА 3. ПРОБЛЕМА ИНДУКЦИИ ЮМА

Рационалисты считают, что чувства обманчивы и единственный верный путь к знанию — логическое рассуждение. Эмпирики уверены, что рассуждения подвержены ошибкам и знание должно быть получено из наблюдений и экспериментов. Французы — рационалисты. Англосаксы (как их называют французы) — эмпирики. Мыслители, юристы и математики — рационалисты. Журналисты, врачи и ученые — эмпирики. В мире информатики теоретики и инженеры знаний — рационалисты. Хакеры и специалисты по машинному обучению — эмпирики.

«Рационализм или эмпиризм?» — любимый вопрос философов. Платон был ранним рационалистом, а Аристотель — ранним эмпириком. Но по-настоящему дебаты разгорелись в эпоху Просвещения, когда по каждую сторону встали по три великих мыслителя: Декарт, Спиноза и Лейбниц были ведущими рационалистами; Локк, Беркли и Юм — их соперниками- эмпириками. Дэвид Юм родился в Шотландии в 1711 году и большую часть своей жизни провел в Эдинбурге, который в XVIII веке процветал и бурлил интеллектуальной жизнью. Юм задал вопрос, как в принципе можно оправдать экстраполяцию того, что мы видели, на то, чего мы не видели?

Философы спорили о проблеме индукции Юма с тех самых пор, как он ее сформулировал, но так и не пришли к удовлетворительному ответу. Бертран Рассел любил иллюстрировать эту проблему историей об индюке-индуктивисте.¹ В первое утро индюку дали корм в девять утра. Но он был хорошим индуктивистом и не спешил с выводами. Он много дней собирал наблюдения при всевозможных обстоятельствах, однако его раз за разом кормили в девять утра. Наконец он сделал вывод: да, его всегда будут кормить в девять утра. А потом наступил канун Рождества и ему перерезали горло.

Аристотель говорил, что в разуме нет ничего такого, чего не было бы в органах чувств. Лейбниц добавил: «Кроме самого разума». Человеческий мозг — это не *tabula rasa*, потому что это совсем не доска: доска пассивна, на ней пишут, а мозг активно обрабатывает получаемую информацию (критику концепции «чистой доски» см. [Карл Поппер. Предположения и опровержения. Рост научного знания](#)). Принцип Ньютона — первое неписаное правило машинного обучения. Путем индукции мы выводим самые широко применимые законы, какие только возможно, и сужаем их действие, только если данные вынуждают нас это сделать.

Для машинного обучения типично начинать с ограничивающих условий и постепенно ослаблять их, если они не объясняют данных. Сначала алгоритм тестирует все отдельные факторы, затем все сочетания двух факторов, потом все сочетания трех и так далее.

С помощью алгоритма для нахождения конъюнктивных² понятий, с которым мы познакомились выше, можно составлять подобные наборы по одному правилу за правилом. Наборы правил популярны в торговых сетях. Пионер в этой области — Walmart. Еще на заре применения этого метода они открыли, что с подгузниками часто покупают пиво. Звучит странно? Одна из интерпретаций такая: молодые матери посылают мужей в супермаркет за подгузниками, а те в качестве компенсации за моральный ущерб покупают себе ящик пива. Зная это, супермаркеты теперь могут продавать больше пенного напитка, выставляя его на полках по соседству с подгузниками. К такому выводу никогда не придешь без поиска правил: «закон пива и подгузников»

¹ Я не нашел у Бертрانا Рассела истории об индюке-индуктивисте. Рассел был англичанином, и у него есть история о... курице (см. Бертран Рассел. Проблемы философии, раздел 6 «Об индукции»). А вот Нассим Талеб не только приводит индюка в пример, но и богато иллюстрирует эту ситуацию; см., например, [Нассим Николас Талеб. Антихрупкость. Как извлечь выгоду из хаоса](#), глава 5. Ливанский базар и офисное здание.

² [Конъюнкция](#) — логическая операция, по смыслу максимально приближенная к союзу «и».

стал легендой среди специалистов по добыче данных (некоторые, правда, утверждают, что это скорее городская легенда).

Тем не менее на конъюнктивных понятиях далеко не уедешь. Проблема, как выразился Редьярд Киплинг, в том, что «путей в искусстве есть семь и десять раз по шесть, и любой из них для песни — лучше всех». В реальной жизни понятия дизъюнктивны. Так что нам нужно находить концепции, которые заданы целым набором правил, а не одним, например:

- Если вашей кредитной карточкой вчера пользовались в Китае, Канаде и Нигерии, значит, ее украли.
- Если вашей кредитной карточкой пользовались два раза после и вечера в будний день, значит, ее украли.
- Если с вашей кредитной карточки купили бензин на один доллар, значит, ее украли.

(Если вы не поняли последнее правило, небольшое пояснение: раньше воры обычно покупали бензин на доллар, чтобы убедиться, что украденная карточка работает. Потом специалисты по добыче данных раскусили этот прием.)

Учиться — значит забывать о подробностях в той же степени, как помнить о важных элементах. Компьютеры — высшее проявление синдрома саванта:³ они без малейших проблем запоминают все, но хотим мы от них не этого. Проблема не ограничивается массовым запоминанием частных фактов. Каждый раз, когда обучающийся алгоритм находит в данных закономерность, которая в реальном мире ошибочна, мы говорим, что он «подогнал под ответ». Переобучение — центральная проблема машинного обучения.

Люди тоже не застрахованы от переобучения. Представьте себе ситуацию: маленькая белая девочка видит в торговом центре девочку-мексиканку и кричит: «Мама, смотри, ребенок-служанка!» (это реальный случай). Дело не в врожденном расизме. Скорее, она слишком обобщила представление о тех немногих латиноамериканках, которых успела увидеть за свою короткую пока жизнь. Наши убеждения основаны на опыте, а опыт дает очень неполную картину мира, поэтому перепрыгнуть к ложным выводам несложно.

Джон фон Нейман, один из основоположников информатики, как-то точно заметил: «С четырьмя параметрами я могу подогнать слона, а с пятью заставляю его махать хоботом». Кто-то даже сказал, что «добывать данные — значит пытаться их до тех пор, пока они не признаются».

Итого: обучение — гонка между количеством данных, имеющихся в вашем распоряжении, и количеством рассматриваемых вами гипотез. Увеличение объема данных экспоненциально уменьшает количество прошедших проверку гипотез, но, если гипотез изначально много, в конце все равно может остаться некоторое количество плохих.

Как же решить, верить ли обучающемуся алгоритму? Все просто: не верьте, пока не проверите результаты на данных, которые *обучающийся алгоритм не видел*. Но вам не надо ждать, пока поступят новые данные, чтобы решить, можно ли доверять алгоритму машинного обучения. Лучше взять все данные, которые у вас есть, и произвольно разделить их на обучающее множество, которое вы дадите алгоритму, и тестовое множество, которое надо спрятать от него и использовать для верификации точности. Точность на скрытых данных — золотой стандарт в машинном обучении.

Еще один популярный подход — отдавать предпочтение более простым гипотезам. Предпочтение более простым гипотезам широко известно, как бритва Оккама (в кратком виде он гласит: «Не следует множить сущее без необходимости»). На практике один из лучших обучающихся алгоритмов — бустинг — извлекает на первый взгляд необоснованно сложные модели (подробнее см. [Джон Форман. Много цифр: Анализ больших данных при помощи Excel](#), глава 7 «Комплексные модели»).

Если точность обучающегося алгоритма в тестовой выборке разочаровывает, надо диагностировать проблемы. В машинном обучении них существуют специальные термины: смещение и дисперсия (рис. 1). У большинства обучающихся алгоритмов есть «ручка», с помощью которой можно отрегулировать гибкость: это, например, порог значимости и штрафы за размер модели. Подстройка — первое, что нужно попробовать.

³ Синдром саванта — редкое состояние, при котором лица с отклонением в имеют «остров гениальности».

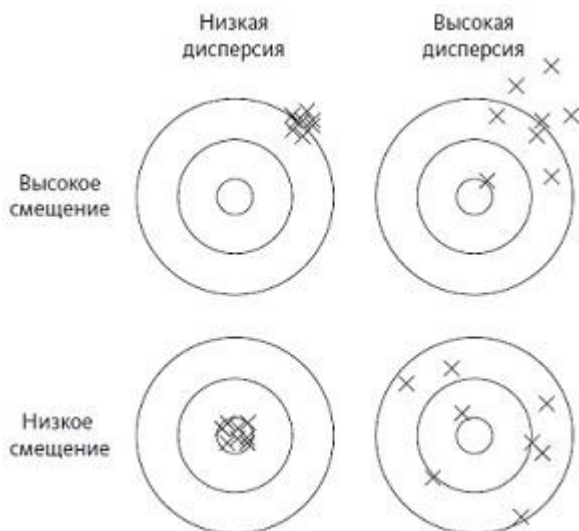


Рис. 1. Смещение и дисперсия

Символисты прибегают к индукции с помощью дерева решений (рис. 2). Каждый путь от корня до листа соответствует правилу. Например, «Если вы за уменьшение налогов, за право на аборт и против свободного оборота оружия, вы демократ».



Рис. 2. Дерево решений

Как выбрать лучший атрибут для тестирования в узле? Поскольку мы стремимся постепенно разделять классы, пока не «очистим» все ветви, символисты опираются на понятие энтропии из теории информации (информационная энтропия — мера неопределенности или непредсказуемости информации). Энтропия набора предметов — мера его неупорядоченности. Если в группе из 150 человек будет 50 республиканцев, 50 демократов и 50 независимых кандидатов, ее политическая энтропия максимальна. С другой стороны, если в группе одни республиканцы, энтропия будет равна нулю, во всяком случае, в отношении партийной принадлежности. Поэтому, чтобы получить хорошее дерево решений, мы выберем в каждом узле атрибут, который в среднем даст самую низкую энтропию классов по всем ее ветвям, с учетом количества примеров в каждой из ветвей (я использовал тот же принцип для поиска оптимального алгоритма в игре «Быки и коровы», см. [Разработка оптимальной стратегии игры «Быки и коровы» на основе теории информации](#)).

Дерево решений, согласно опросам, стало самым широко используемым алгоритмом машинного обучения, что неудивительно: эту методику легко понять и освоить, и обычно она дает довольно точный результат без лишних настроек. Джон Росс Куинлан — самый выдающийся исследователь в школе символистов. Этот невозмутимый прагматичный австралиец год за годом неустанно улучшал деревья решений, сделал их золотым стандартом в области классификации и пишет о них удивительно ясные статьи.

Важнейшее убеждение символистов заключается в том, что интеллект можно свести к манипулированию символами. Согласно этой гипотезе, интеллект не зависит от носителя: можно писать символы мелом на доске, включать и выключать транзисторы, выражать их импульсами между нейронами. Если у вас есть структура, обладающая мощностью универсальной машины Тьюринга, вы сможете сделать все что угодно. Программное обеспечение можно вообще отделить от «железа», и, если вы хотите просто разобраться, как могут учиться машины, вам (к счастью) не надо волноваться о машинах как таковых.

Но несмотря на популярность деревьев решений, более удобный исходный пункт для поисков Верховного алгоритма — обратная дедукция. Обратная дедукция как сверхученый. Он будет систематически рассматривать доказательства, взвешивать возможные выводы, сопоставлять лучшие и использовать их вместе с другими доказательствами для формулировки дальнейших гипотез, и все это с компьютерной скоростью. С другой стороны, у метода есть ряд серьезных недостатков. Обратную дедукцию легко запутать шумом: как разобраться, каких шагов в дедукции не хватает, если предположения или заключения ложны? Еще более серьезно то, что реальные понятия очень часто не определяются сжатым набором правил. Они не черно-белые, а находятся в большой серой зоне между, скажем, спамом и не-спамом, поэтому приходится взвешивать и накапливать слабые доказательства, пока картина не прояснится. Никто еще не преуспел в обучении набору правил, которое будет определять кошку, глядя на пиксели на картинке, и, наверное, это просто невозможно.

Очень критично по отношению к символистскому обучению настроены коннекционисты. Они считают, что понятия, которые можно определить с помощью логических правил, лишь вершина айсберга, а в глубине есть много такого, что формальные рассуждения просто неспособны увидеть, точно так же как значительная часть работы мозга скрыта в подсознании.

ГЛАВА 4. КАК УЧИТСЯ НАШ МОЗГ?

В конце XIX в. великий испанский нейробиолог Сантьяго Рамон-и-Кахаль провел первые подробные исследования мозга, окрашивая нейроны, и каталогизировал свои наблюдения, как ботаники классифицируют новые виды деревьев (подробнее см. [Крис Фрит. Мозг и душа](#)). В вышедшей в 1949 году книге *Организация поведения* канадский психолог Дональд Хебб первым предложил механизм, согласно которому нейроны могут кодировать ассоциации.

Нейроны, которые возбуждают друг друга, образуют, в терминологии Хебба, «ансамбли клеток». С помощью таких собраний в головном мозге представлены понятия и воспоминания. В каждый ансамбль могут входить нейроны из разных областей мозга, ансамбли могут пересекаться. Если вы спросите символистскую систему, где находится понятие «Нью-Йорк», она укажет точное место его хранения в памяти. В коннекционистской системе ответ будет «везде понемногу».

Еще одно отличие между символистским и коннекционистским обучением заключается в том, что первое — последовательное, а второе — параллельное. В случае обратной дедукции мы шаг за шагом разбираемся, какое правило необходимо ввести, чтобы от посылок прийти к желаемым выводам. В коннекционистской модели все нейроны учатся одновременно, согласно правилу Хебба. Компьютеры работают последовательно. При этом шаги могут быть очень быстрыми, потому что транзисторы способны включаться и выключаться миллиарды раз в секунду. Мозг же умеет выполнять большое количество вычислений параллельно благодаря одновременной работе миллиардов нейронов. При этом нейроны могут стимулироваться в лучшем случае тысячу раз в секунду, и каждое из этих вычислений медленное. Количество транзисторов в компьютере приближается к количеству нейронов в головном мозге человека, однако мозг безусловно выигрывает в количестве соединений.

Первая формальная модель нейрона была предложена в 1943 году Уорреном Маккаллоком и Уолтером Питтсом. Она была во многом похожа на логические вентили, из которых состоят

компьютеры. Вентиль ИЛИ включается, когда как минимум один из его входов включен, а вентиль И — когда включены все. Однако нейрон Маккаллока-Питтса не умеет учиться. Для этого соединениям между нейронами надо присвоить переменный вес, и в результате получится так называемый перцептрон. Перцептроны были изобретены в конце 1950-х Фрэнком Розенблаттом, психологом из Корнелльского университета.

В перцептроне положительный вес представляет возбуждающее соединение, а отрицательный — ингибирующее. Если взвешенная сумма входов перцептрона выше порогового значения, он выдает единицу, а если ниже — ноль (рис. 3).

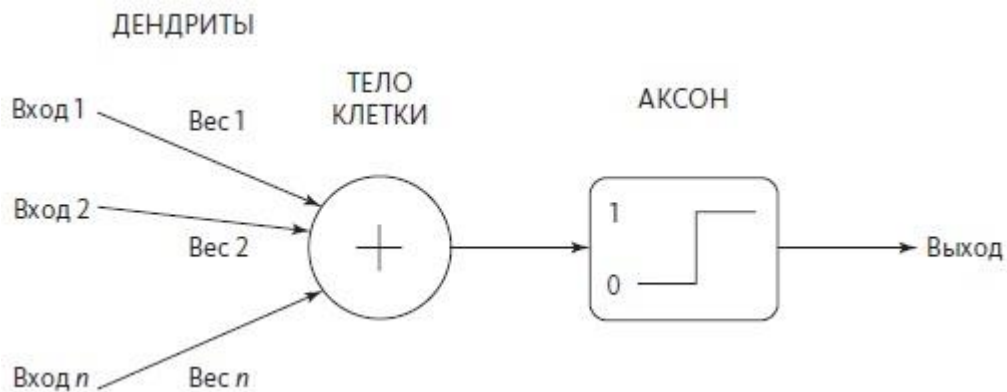


Рис. 3. Модель перцептрона

Перцептрон вызвал восторг в научном сообществе. Он был простым, но при этом умел узнавать печатные буквы и звуки речи: для этого требовалось только обучение на примерах. Но затем перцептрон уперся в стену. В 1969 году Минский и его коллега Сеймур Пейперт опубликовали книгу [Перцептроны](#), где подробно, один за другим описали простые вещи, которым одноименный алгоритм не в состоянии научиться. Самый простой и потому самый убийственный пример — это функция «исключающее ИЛИ» (сокращенно XOR), которая верна, если верен один, но не оба входа.

Поскольку перцептроны могут находить только линейные границы, XOR для них недоступен, а если они неспособны даже на это, значит, перцептрон — не лучшая модель того, как учится мозг, и неподходящий кандидат в Верховные алгоритмы.

Нейрон больше напоминает не логический вентиль, а преобразователь напряжения в частоту (рис. 4). Эту похожую на вытянутую букву S кривую называют по-разному: логистической, S-образной, сигмоидой.



Рис. 4. Кривая зависимости частоты от напряжения, как пример S-образной кривой

Форму сигмоиды имеют всевозможные фазовые переходы: намагничивание железа, таяние льда, испарение воды, прерывистое равновесие в эволюции, смена научных парадигм, распространение новых технологий, бегство белого населения из смешанных районов, слухи, эпидемии, революции, падения империй и многое другое. Книгу Малькольма Гладуэлла [Переломный момент](#) можно было бы (хотя и менее заманчиво) назвать «Сигмоида» (эта кривая встречалась мне многократно, например, она [появляется в модели Раша при пересчете баллов ЕГЭ](#)). В романе Хемингуэя «И восходит солнце» Майка Кэмпбелла спрашивают, как он обанкротился, и тот отвечает: «Двумя

способами. Сначала постепенно, а потом сразу». То же самое могли бы сказать в банке Lehman Brothers. В этом суть сигмоиды. Дифференцируйте сигмоиду, и вы получите гауссову кривую (подробнее см. [Нормальное распределение](#)).

Давайте заменим ступенчатую функцию перцептрона сигмоидой и посмотрим, что произойдет. Допустим, вы обучаете компьютер узнавать бабушку. Всякий раз, когда «сетчатка» обучающегося алгоритма видит новый образ, сигнал распространяется по всей сети, пока не даст выход. Сравнение полученного выхода с желаемым выдает сигнал ошибки, который затем распространяется обратно через все слои и достигает сетчатки. На основе возвращающегося сигнала и вводных, полученных во время прохождения вперед, каждый нейрон корректирует веса. По мере того как сеть видит все новые и новые изображения вашей бабушки и других людей, веса постепенно сходятся со значениями, которые позволяют отличить одно от другого. Метод обратного распространения ошибки, как называется этот алгоритм, несравнимо мощнее перцептрона. Это делает обратное распространение ошибки верховным алгоритмом коннекционистов.

К сожалению, в жизни все не так просто. Представьте, что у вашей сети только один вес (рис. 5). Оптимальный вес, в котором ошибка самая низкая, — это 2,0. Если сеть начнет работу, например, с 0,75, обратное распространение ошибки за несколько шагов придет к оптимуму, как катящийся с горки мячик. Однако если начать с 5,5, мы скатимся к весу 7,0 и застрянем там. Обратное распространение ошибки со своими поэтапными изменениями весов не сможет найти глобальный минимум ошибки.

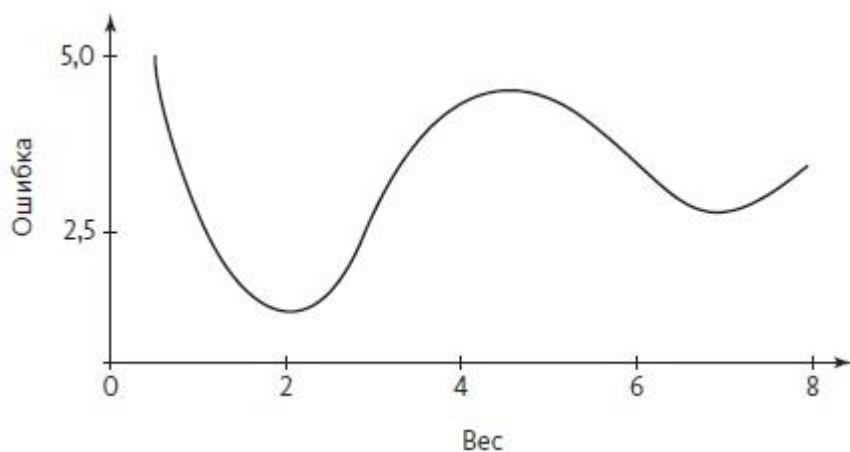


Рис. 5. Зависимости ошибки от веса фактора

Таким образом, обучение многослойного перцептрона хаотично в том смысле, что, начав из слегка отличающихся мест, он может привести к весьма различным решениям. Этот феномен проявляется в случае незначительных отличий как в исходных весах, так и в обучающих данных.

ГЛАВА 5. ЭВОЛЮЦИЯ: ОБУЧАЮЩИЙСЯ АЛГОРИТМ ПРИРОДЫ

Рональд Фишер в своей книге [Генетическая теория естественного отбора](#) сформулировал первую математическую теорию эволюции. Генетические алгоритмы во многом похожи на искусственную селекцию. Дарвин открывает «Происхождение видов» дискуссией на эту тему, чтобы, оттолкнувшись от нее, перейти к более сложной концепции естественного отбора. Все одомашненные растения и животные, которые мы сегодня воспринимаем как должное, появились в результате многих поколений отбора и спаривания организмов, лучше всего подходящих для наших целей: кукурузы с самыми крупными початками, деревьев с самыми сладкими фруктами, самых длинношерстных овец, самых выносливых лошадей. Генетические алгоритмы делают то же самое, только выращивают они не живых существ, а программы, и поколение длится несколько секунд компьютерного времени, а не целую жизнь.

Джон Холланд видел настоящую мощь генетических алгоритмов в половом размножении. Обмен генетического материала между хромосомами отца и матери называется «кроссинговер», и его результат — появление двух новых хромосом (рис. 6). Генетический алгоритм основан на подражании этому процессу. В каждом поколении он сводит друг с другом самые приспособленные особи, перекрещивает их битовые строки в произвольной точке и получает двух потомков от каждой пары родителей. После этого алгоритм делает в новых строках точечные мутации и отпускает в

виртуальный мир. Когда строки возвращаются с присвоенным значением приспособленности, процесс повторяется заново. Каждое новое поколение более приспособлено, чем предыдущее, и процесс прерывается либо после достижения желаемой приспособленности, либо, когда заканчивается время.

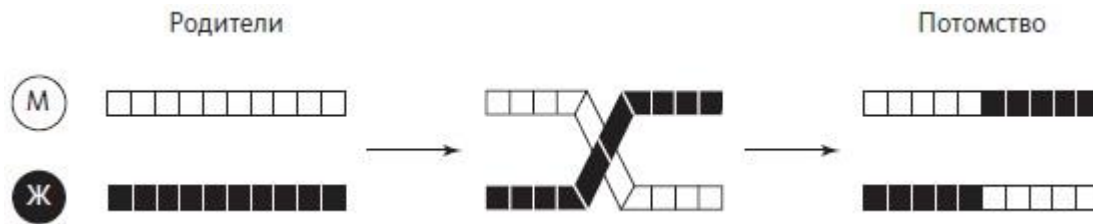


Рис. 6. Обмен генетическим материалом в процессе полового размножения – кроссинговер

Поймать естественный отбор в серии уравнений крайне сложно, однако выразить его в виде алгоритма — совсем другое дело, и это могло бы пролить свет на многие мучающие человечество вопросы. Почему виды появляются в палеонтологической летописи внезапно? Где доказательства, что они постепенно эволюционировали из более ранних видов? В 1972 году Нильс Эддридж и Стивен Джей Гулд предположили, что эволюция состоит из ряда «прерывистых равновесий»: перемежающихся длинных периодов застоя и коротких всплесков быстрых изменений, одним из которых стал кембрийский взрыв.

Эксперименты с генетическими алгоритмами говорят в пользу скачков. Если запустить такой алгоритм на 100 тысяч поколений и понаблюдать за популяцией в тысячепоколенных отрезках, график зависимости приспособленности от времени будет, вероятно, похож на неровную лестницу с внезапными скачками улучшений, за которыми идут плоские периоды затишья, со временем длящиеся все дольше.

Обратите внимание, насколько генетические алгоритмы отличаются от многослойных перцептронов. Метод обратного распространения в любой момент времени рассматривает одну гипотезу, и эта гипотеза постепенно меняется, пока не найдет локальный оптимум. Генетические алгоритмы на каждом этапе рассматривают всю популяцию гипотез и благодаря кроссинговеру способны делать большие скачки от одного поколения к другому.

Благодаря этому генетические алгоритмы с намного меньшей вероятностью, чем обратное распространение ошибки, застревают в локальном оптимуме и в принципе более способны прийти к чему-то по-настоящему новому. В то же время их намного сложнее анализировать. Откуда нам знать, что генетический алгоритм получит что-то осмысленное, а не будет, как пьяный, слоняться вокруг да около?

В реальности природа не предшествует воспитанию: они скорее перемежаются, и каждый раунд обучения «воспитанием» готовит сцену для следующего раунда обучения «природой», и наоборот. Гусята постоянно ходят за своей мамой (поведение, сформировавшееся в ходе эволюции), но для этого они должны ее узнавать (выученная способность). Если вместо гусыни вылупившиеся птенцы увидят человека, они будут следовать за ним: это замечательно показал Конрад Лоренц (подробнее см. [Конрад Лоренц. Так называемое зло](#)). В мозге новорожденного свойства среды уже закодированы, но косвенно: эволюция оптимизирует мозг для извлечения этих свойств из ожидаемых вводных.

Эволюция ищет удачные структуры, а нейронное обучение их заполняет: такое сочетание — самый легкий шаг к Верховному алгоритму. Этот подход может удивить любого, кто знаком с бесконечными перипетиями спора о роли природы и воспитания, который не утихает две с половиной тысячи лет. Однако если посмотреть на жизнь глазами компьютера, многое проясняется. «Природа» для компьютера — это программа, которую он выполняет, а «воспитание» — получаемые им данные. Вопрос, что важнее, очевидно абсурден.

Однако, у порождений эволюции много очевидных изъянов. Например, зрительный нерв млекопитающих связан с передней, а не с задней частью сетчатки, из-за чего рядом с центральной ямкой, областью самого четкого зрения, появляется просто вопиюще ненужное слепое пятно. В архитектуре головного мозга тоже могут быть недостатки: у мозга много ограничений, которых

лишены компьютеры — например, очень ограниченная краткосрочная память, — и нет причин их сохранять. Более того, известно много ситуаций, в которых люди постоянно поступают неправильно, и Даниэль Канеман пространно иллюстрирует это в своей книге [Думай медленно... решай быстро](#).

ГЛАВА 6. В СВЯТИЛИЩЕ ПРЕПОДОБНОГО БАЙЕСА

Теорема Байеса — это просто несложное правило обновления уровня доверия к гипотезе при получении новых доказательств: если свидетельство совпадает с гипотезой, ее вероятность идет вверх, если нет — вниз. Томас Байес — английский священник, живший в XVIII веке. Но байесианство в привычном для нас виде было изобретено Пьер-Симоном де Лапласом — французом, родившимся на пять десятилетий позже Байеса. Байес был проповедником и первым описал новый подход к вероятностям, но именно Лаплас выразил его идеи в виде теоремы.

Откуда, например, мы знаем, что завтра взойдет солнце? Однажды — скажем, в начале времен, которое для Лапласа было приблизительно 5 тысяч лет назад, — мы просыпаемся, прекрасно проводим день, а вечером видим, что солнце заходит. Вернется ли оно? Мы никогда не видели восхода, и у нас нет причин полагать, что оно взойдет или не взойдет. Таким образом мы должны рассмотреть два одинаково вероятных сценария и сказать, что солнце снова взойдет с вероятностью $\frac{1}{2}$. Но, продолжал Лаплас, если прошлое хоть как-то указывает на будущее, каждый день, когда солнце восходит, должен укреплять нашу уверенность, что так будет происходить и дальше. Спустя пять тысячелетий вероятность, что солнце завтра снова взойдет, должна быть очень близка единице, но не равняться ей, потому что полной уверенности никогда не будет. Из этого мысленного эксперимента Лаплас вывел свое так называемое правило следования, согласно которому вероятность, что солнце снова взойдет после n восходов, равна $(n + 1) / (n + 2)$. Если $n = 0$, то это просто $\frac{1}{2}$, а когда n увеличивается, растет и вероятность, стремясь к единице, когда n стремится к бесконечности.

Возникает важнейший вопрос: как именно должна меняться апостериорная вероятность при появлении все большего объема доказательств? Ответ дает теорема Байеса:

$$P(\text{причина} | \text{следствие}) = P(\text{причина}) \times P(\text{следствие} | \text{причина}) / P(\text{следствие})$$

Люди, оказывается, не очень хорошо владеют байесовским выводом, по крайней мере в устных рассуждениях. Проблема в том, что мы склонны пренебрегать априорной вероятностью причины. Теорема Байеса полезна, потому что обычно известна вероятность следствий при данных причинах, а узнать хотим вероятность причин при данных следствиях. Например, мы знаем процент пациентов с гриппом, у которых повышена температура, но на самом деле нам нужно определить вероятность, что пациент с температурой болен гриппом. Теорема Байеса позволяет нам перейти от одного к другому.

Теорема Байеса как основа статистики и машинного обучения страдает, как от вычислительной сложности, так и от крайней противоречивости. Для большинства статистиков единственный допустимый способ оценки вероятностей — вычисление частоты соответствующего события. Например, вероятность гриппа равна 0,2, потому что им болело 20 из 100 обследованных пациентов. Это «частотная» интерпретация вероятности, и она дала название господствующему учению в статистике. Но обратите внимание, что в примере с восходом солнца мы просто высасываем вероятность из пальца. Чем оправдано априорное предположение, что вероятность восхода солнца равна одной второй, двум третьим или еще какой-то величине? На это байесовцы отвечают, что вероятность — это не частота, а субъективная степень убежденности, поэтому вам решать, какая она будет, а байесовский вывод просто позволяет обновлять априорные убеждения после появления новых доказательств, чтобы получать апостериорные убеждения (это называется «провернуть ручку Байеса»).

Алгоритм машинного обучения, который применяет теорему Байеса и исходит из того, что следствия данной причины независимы, называется наивный байесовский классификатор (подробнее см. [Наивный байесовский классификатор документов в Excel](#)). Дело в том, что такое допущение, прямо скажем, довольно наивное: в реальности наличие одного следствия повышает вероятность обнаружить и иные следствия.

Наивный байесовский алгоритм сейчас используется очень широко: на нем основаны, например, многие спам-фильтры. Это применение придумал Дэвид Хекерман, выдающийся ученый-байесовец,

врач. Ему пришла в голову мысль, что к спаму надо относиться как к заболеванию, симптомы которого — слова в электронном письме.

В простейших поисковых системах алгоритмы, довольно похожие на наивный байесовский, используются, чтобы определить, какие сайты выдавать в ответ на запрос. Основное различие в том, что вместо классификации на спам и не спам нужно определить, подходит сайт к запросу или не подходит.

В начале 1980-х годов Джуда Перл, профессор информатики в Калифорнийском университете в Лос-Анджелесе, изобрел новое представление — байесовские сети. В любой байесовской сети, чтобы получить вероятность полного состояния, перемножают вероятности меньшего числа состояний (некоторые из них зависимы), чем, если считать, что все состояния независимы. Первый закон экологии, согласно биологу Барри Коммонеру, заключается в том, что все взаимосвязано. Может быть, это действительно так, но в таком случае мир был бы непостижим, если бы не спасительная условная независимость: все связано, но лишь косвенно. Как заметил один шутник, благодаря пространству с нами происходит не все сразу. Иначе говоря, структура пространства — это частный случай условной независимости.

При условной независимости информация не теряется из-за перехода на более компактное представление, и можно легко вычислить вероятности крайне необычных состояний, включая те, что до этого никогда не наблюдались. Байесовские сети показывают ошибочность расхожего мнения, будто машинное обучение неспособно предсказывать очень редкие события — «черных лебедей», как их называет Нассим Талеб.

Для байесовца не существует такого понятия, как истина: есть априорное распределение гипотез, и после появления данных оно становится апостериорным распределением по теореме Байеса:

$$P(\text{гипотеза} \mid \text{данные}) = P(\text{гипотеза}) \times P(\text{данные} \mid \text{гипотеза}) / P(\text{данные}).$$

Соедините сложность и вычислительные затраты с неоднозначностью байесовской идеи о том, что объективной реальности вообще не существует, и вы поймете, почему в науке последние 100 лет доминирует частотный подход к вероятности. Однако, байесовцы могут заявить сторонникам частотного подхода: «Смотрите, то, что вы делаете, — частный случай того, что делаем мы, но наши допущения хотя бы явные». Байесовцы могут сделать нечто гораздо более интересное: использовать априорное распределение, чтобы закодировать экспертное знание о проблеме. Это их ответ на вопрос Юма.

Несмотря на то что частотный и байесовский типы мышления по некоторым вопросам сходятся, между ними остается философское различие в отношении значения вероятности. Многим ученым неприятно рассматривать его как нечто субъективное, хотя благодаря этому становятся возможными многие применения, которые в противном случае запрещены. Если вы сторонник частотного подхода, можно оценивать вероятности только тех событий, которые происходят более одного раза, и вопросы вроде «Какова вероятность, что Хиллари Клинтон победит Джеба Буша⁴ на следующих президентских выборах?» не имеют ответа, потому что еще не было выборов, в которых сошлись бы эти кандидаты. Для байесовца же вероятность — субъективная степень веры, поэтому он волен выдвигать обоснованные предположения, и анализ суждений делает все его предположения состоятельными.

ГЛАВА 7. ТЫ – ТО, НА ЧТО ТЫ ПОХОЖ

У рассуждений по аналогии выдающаяся интеллектуальная родословная. Еще Аристотель выразил их в своем законе подобия: если две вещи схожи, мысль об одной из них будет склонна вызывать мысль о другой. Если вы молодой президент и столкнулись с мировым кризисом, как в свое время Кеннеди, когда самолет-разведчик обнаружил на Кубе советские ядерные ракеты, вполне вероятно, что готового сценария у вас не окажется. Вместо этого можно поискать похожие примеры в истории и попытаться сделать из них выводы. Объединенный комитет начальников штабов подталкивал президента напасть на Кубу, но Кеннеди только что прочитал [Августовские пушки](#) — бестселлер о начале Первой мировой войны — и хорошо осознавал, что такой шаг легко может вылиться в

⁴ Книга вышла в 2015 г. Автор, видимо, считал, что наиболее вероятным кандидатом от республиканцев на выборах 2016 г. станет Джем Буш. Сегодня мы знаем, что кандидатом стал Дональд Трамп.

тотальную войну. Кеннеди предпочел морскую блокаду — и, может быть, спас мир от ядерной катастрофы.

Первое воплощение аналогии в алгоритме появилось в малоизвестном отчете, написанном в 1951 году Эвелин Фикс и Джо Ходжесом — статистиками из Университета Беркли. Назывался он — алгоритм ближайшего соседа.

Представьте, что вы, как Netflix, собрали базу данных, где каждый пользователь присваивает просмотренным фильмам рейтинг от одной до пяти звезд. Вы хотите определить, понравится ли вашему пользователю по имени Кен фильм «Гравитация», поэтому ищите пользователей, оценки которых лучше всего коррелируют с оценками Кена. Если все они присвоили «Гравитации» высокий рейтинг, вероятно, так поступит и Кен, и этот фильм можно ему посоветовать.

Сегодня для того, чтобы давать рекомендации посетителям сайта, используются все виды алгоритмов, но взвешенный k -ближайший сосед был первым, нашедшим широкое применение в этой области, и его до сих пор сложно победить (о реализации алгоритма в Excel см. [Кластерный анализ: сетевые графы и определение сообществ](#)).

Конечно, в райском саду есть и Змей. Его зовут Проклятие Размерности, и, хотя он в большей или меньшей степени поражает все алгоритмы машинного обучения, для ближайшего соседа он особенно опасен. Все даже хуже. Ближайший сосед основан на нахождении схожих объектов, а в высоких измерениях распадается сама идея сходства. Гиперпространство — как сумеречная зона. Наша интуиция, основанная на опыте жизни в трех измерениях, там не действует, и начинают происходить все более и более странные вещи. Представьте себе апельсин: шарик вкусной мякоти, окруженный тонкой кожицей. Мякоть в апельсине занимает, скажем, 90% радиуса, а оставшиеся десять приходятся на кожуру. Это означает, что 73% объема апельсина — это мякоть ($0,9^3$). Теперь рассмотрим гиперапельсин: если мякоть занимает все те же 90% радиуса, но, скажем, в сотне измерений, то она сократится примерно до всего лишь 3/1000 процента объема ($0,9^{100}$). Гиперапельсин будет состоять из одной кожуры, и его никогда нельзя будет очистить!

Самый главный вопрос во многих аналогических обучающихся алгоритмах — как измерять сходство. Например, сходство между двумя молекулами можно определить по числу идентичных субструктур, которые они содержат. Метан и метанол схожи, потому что в них есть три связи углерода с водородом, а отличаются они только тем, что в метаноле один атом водорода замещен гидроксильной группой (рис. 7). Однако это не означает, что схожи химические свойства веществ, ведь метан — это газ, а метанол — спирт.

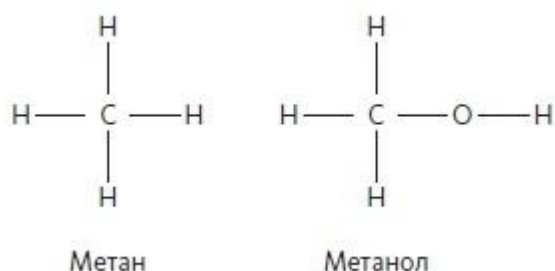


Рис. 7. Критерии сходства: метан и метанол

Дэвид Коуп, почетный профессор музыки в Калифорнийском университете в Санта-Круз, разработал алгоритм, который пишет новые музыкальные произведения в стиле известных композиторов путем отбора и рекомбинации коротких отрывков из их сочинений. На конференции Коуп продемонстрировал три пьесы: одну на самом деле написанную Моцартом, другую — композитором, имитировавшим его, и третью — сгенерированную системой. Затем Коуп попросил аудиторию проголосовать. Вольфганг Амадей победил, но имитатор-человек уступил компьютеру. Если Коуп прав, то творчество — высшее из непостижимого — сводится к аналогии и рекомбинации.

Есть ли что-то, на что неспособна аналогия? Нет, считает Даглас Хофштадтер, когнитивный психолог и автор книги [Гедель, Эшер, Бах. Эта бесконечная гирлянда](#). В книге «Поверхности и сущности: аналогия в роли топлива и огня мышления» Хофштадтер и Эммануэль Сандер страстно доказывают, что все разумное поведение сводится к аналогии. Гедель, Эшер, Бах — расширенная аналогия между теоремой Гёделя, искусством Эшера и музыкой Баха. Если Верховный алгоритм — это не аналогия, он

несомненно должен быть в чем-то схож с ней (см. также [Эрнест Нагель, Джеймс Рой Ньюмен. Теорема Гёделя](#)).

ГЛАВА 8. ОБУЧЕНИЕ БЕЗ УЧИТЕЛЯ

Нам нужен алгоритм, который будет спонтанно группировать схожие объекты и разные изображения одного и того же объекта. Это проблема кластеризации, одна из наиболее интенсивно изучаемых тем в науке о машинном обучении. Кластер — это набор схожих сущностей или как минимум набор сущностей, которые похожи друг на друга больше, чем на элементы других кластеров. Делить все на кластеры — в природе человека, и часто это первый шаг на пути к знанию. Наблюдение, что определенные группы веществ имеют очень схожие химические свойства, стало первым шагом к открытию периодической системы элементов: каждая группа в ней заняла свой столбец. Рынок в целом — слишком общий, отдельные клиенты — слишком мелкие, поэтому маркетологи делят его на сегменты, как они называют кластеры.

Метод k -средних появился в 50-е годы XX века. Он простой, красивый, но имеет ряд недостатков. Во-первых, количество кластеров надо зафиксировать заранее, а в реальном мире это невозможно. Один вариант решения — позволить открывать новый кластер, если объект слишком сильно отличается от имеющихся. Другой — разрешить кластерам делиться и сливаться в процессе работы. Более серьезная проблема заключается в том, что метод k -средних работает, только когда кластеры легко различимы.

Психологи обнаружили, что личность можно выразить [пятью факторами](#) — это экстраверсия, доброжелательность, добросовестность, нейротизм и открытость опыту, — которые оценивают по твитам и постам в блогах.

Есть целый подраздел машинного обучения, посвященный алгоритмам, которые исследуют мир сами по себе: трудятся, сталкиваются с наградами, определяют, как получить их снова. Во многом они похожи на детей, которые ползают по комнате и тащат все в рот. Это *обучение с подкреплением*. Мысль, что не все состояния ведут к награде (положительной или отрицательной), но у каждого состояния имеется ценность, — центральный пункт обучения с подкреплением. В настольных играх награды есть только у конечных позиций (например, 1, 0 и -1 для победы, ничьей и поражения). Другие позиции не дают немедленной награды, но их ценность в том, что они могут обеспечить награду в будущем.

Детская игра намного серьезнее, чем может показаться: если эволюция создала существо, которое в первые несколько лет своей жизни беспомощно и только обременяет родителей, такая расточительность должна давать большие преимущества. По сути, обучение с подкреплением — своего рода ускоренная эволюция, которая позволяет попробовать, отбросить и отточить действия в течение одной жизни, а не многих поколений, и по этим меркам оно крайне эффективно. В 1989 году Крис Уоткинс из Кембриджа пришел к современной формулировке обучения с подкреплением как оптимального контроля в неизвестной среде.

Элементы обучения с подкреплением, также называемые привычками, составляют большую часть наших действий: проголодался — идешь к холодильнику и берешь что-нибудь перекусить. Как показал Чарльз Дахигг в книге [Сила привычки](#), понимание и управление этим циклом намеков, рутинных действий и наград — ключ к успеху не только для отдельных людей, но и для бизнеса, и даже для общества в целом.

Учиться — значит становиться лучше с практикой. С практикой вы становитесь искуснее, но скорость не постоянна: сначала улучшения приходят быстро, потом все медленнее, а затем совсем замедляются. Неважно, осваиваете вы игры или учитесь играть на гитаре: кривая зависимости улучшения результатов от времени — насколько хорошо вы что-то делаете и сколько времени это занимает — имеет очень характерную форму (рис. 8). Этот тип кривой называют степенным законом, потому что изменение эффективности зависит от возведения времени в какую-то отрицательную степень (см. также [Закон Ципфа и фрактальная природа социальных и экономических явлений](#) и [Кривая обучаемости, кривая обучения, кривая опыта](#)).

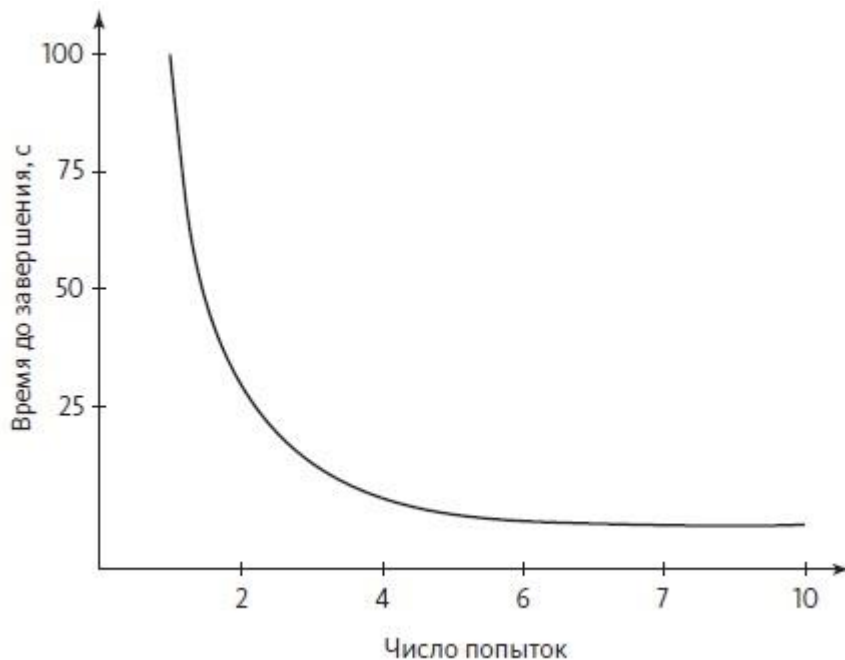


Рис. 8. Кривая обучения

До появления Google поисковые движки решали, соответствует ли веб-страница вашему запросу, заглядывая в ее содержимое, — что еще можно сделать? Идея Брина и Пейджа заключалась в том, что самый сильный признак, указывающий на то, что страница подходит, — это ссылки на нее с других подходящих страниц. Хищника и жертву объединяют сильно взаимосвязанные свойства, каждое из которых эволюционировало, чтобы победить соперника. Во всех этих случаях лучший способ понять сущность — будь то человек, животное, веб-страница или молекула, — понять, как она связана с другими сущностями.

ГЛАВА 9. КУСОЧКИ МОЗАИКИ ВСТАЮТ НА МЕСТО

Наш первый шаг к Верховному алгоритму будет на удивление простым. Как оказывается, несложно соединить много разных обучающихся алгоритмов в один, используя так называемое метаобучение. Представьте, что каждый из обучающихся алгоритмов — эксперт в комитете. Алгоритмы, которые часто предсказывают правильный класс, получают высокий вес, а неточные будут чаще игнорироваться. Нейт Сильвер соединяет результаты опросов аналогичным образом, чтобы спрогнозировать результаты выборов.

Одной из разновидностей метаалгоритма является бэггинг, изобретенный статистиком Лео Брейманом (реализацию бэггинга в Excel см. [Комплексное моделирование или бэггинг](#)). Бэггинг генерирует случайные вариации обучающего набора путем перевыборки, применяет к каждой вариации один и тот же алгоритм машинного обучения и соединяет результаты путем голосования. Это нужно для того, чтобы уменьшить дисперсию: объединенная модель гораздо менее чувствительна к капризам данных, чем любая единичная, поэтому это замечательно легкий способ улучшить точность.

Один из самых сообразительных метаалгоритмов — бустинг, созданный двумя теоретиками обучения, Йоавом Фройндом и Робом Шапире. Бустинг не соединяет разные обучающиеся алгоритмы, а раз за разом применяет к данным один и тот же классификатор, используя новую модель, чтобы исправить ошибки предыдущей путем присвоения весов обучающим примерам (о бустинге см. [Джон Форман. Много цифр: Анализ больших данных при помощи Excel!](#)).

ГЛАВА 10. МИР МАШИННОГО ОБУЧЕНИЯ

В ближайшие десятилетия машинное обучение повлияет на множество аспектов человеческой жизни. Цифровое будущее начинается с осознания факта: взаимодействуя с компьютером — будь то ваш собственный смартфон или удаленный за тысячи километров сервер, — вы каждый раз делаете это на двух уровнях. Первый — желание немедленно получить то, что вам нужно: ответ на вопрос, желаемый товар, новую кредитную карточку. На втором уровне, стратегическом и самом важном, вы рассказываете компьютеру о себе. Чем больше вы его учите, тем лучше он будет вам служить или

манипулировать вами. Жизнь — это игра между вами и окружающими вас обучающимися алгоритмами.

Если вы не хотите, чтобы из-за рождественских покупок Amazon запутался в ваших предпочтениях, заказывайте подарки на других сайтах. (Прости, Amazon.) Если дома и на работе вы смотрите разные типы видео, заведите два аккаунта на YouTube. А если вы собираетесь посмотреть то, что вас обычно не интересует, сначала разлогиньтесь. Безопасный режим Chrome используйте не для просмотра сомнительных сайтов, а когда хотите, чтобы текущая сессия не повлияла на персонализацию в будущем. Если в аккаунт на Netflix добавить профили для разных людей, сайт не станет рекомендовать вам взрослые фильмы для вечернего просмотра в кругу семьи.

Сегодня данные можно разделить на четыре категории: те, которыми вы делитесь со всеми, те, которыми вы делитесь только с друзьями и коллегами, те, которыми вы делитесь с различными компаниями (сознательно или нет), и те, которые вы вообще не распространяете.

На заре появления искусственного интеллекта считалось, что «синих воротничков» компьютеры заменят раньше, чем «белых», потому что последним приходится больше думать. Но дело обернулось совсем не так. Как оказалось, оценивать заявления о кредите машинам проще, чем не спотыкаясь ходить по стройплощадке, хотя у людей все наоборот. Узко определенные задания легко научиться решать, имея данные, а вот задачи, требующие широкого сочетания навыков и знания, научиться решать не так просто. Большая часть мозга человека выделена для обеспечения зрения и движений, то есть ходьба — намного более сложное дело, чем может показаться, но мы принимаем необходимость ходить как должное, поскольку этот процесс доведен до совершенства эволюцией и в основном выполняется подсознательно.

Чем больше контекста требует профессия, тем менее вероятно, что компьютеры быстро ее освоят. Здравый смысл важен не только потому, что так говорила мама, но и потому, что у компьютеров его нет. Лучший способ не потерять работу — самому ее автоматизировать и сосредоточиться на тех ее аспектах, на которые вам не хватало времени и которые компьютер пока не может освоить.

Сейчас есть опасения, что гуманитарные науки вошли в штопор и вымирают, однако, когда другие области будут автоматизированы, они восстанут из пепла. Чем обширнее и дешевле автоматизированное производство, тем ценнее вклад гуманитариев.

ЛИТЕРАТУРА НА РУССКОМ ЯЗЫКЕ

Абигнейл Ф., Реддинг С. *Поймай меня, если сможешь*. М.: Эт Сетера Пабблишинг, 2003.

Айзексон У. *Иноваторы*. М.: АСТ: Corpus, 2015.

Андерсон К. [Длинный хвост](#). Новая модель ведения бизнеса. М.: Вершина, 2008.

Берлин И. [История свободы. Россия](#). — М.: Новое литературное обозрение, 2014.

Востром Н. *Искусственный интеллект. Этапы. Угрозы. Стратегии*. М.: Манн, Иванов и Фербер, 2015.

Гладуэлл М. [Переломный момент](#): Как незначительные изменения приводят к глобальным переменам». М.: Альпина Паблишер, 2015.

Глейк Дж. [Хаос: создание новой науки](#). СПб.: Амфора, 2001.

Дасгупта С., Пападимитриу Х., Вазирани У. *Алгоритмы*. М.: Издательство МЦНМО, 2014.

Дахигг Ч. [Сила привычки](#). Почему мы живем и работаем так, а не иначе. М.: Карьера Пресс, 2015.

Докинз Р. *Расширенный фенотип. Длинная рука гена*. М.: Астрель: Corpus, 2010.

Докинз Р. [Слепой часовщик](#). Как эволюция доказывает отсутствие замысла во Вселенной. М.: АСТ: Corpus, 2015.

Дэвенпорт Т., Харрис Дж. [Аналитика как конкурентное преимущество](#). Новая наука побеждать. М.: BestBusinessBooks, 2010.

Зигель Э. *Просчитать будущее. Кто кликнет, купит, совет или умрет*. М.: Альпина Паблишер, 2014.

Канеман Д. [Думай медленно... решай быстро](#). М.: АСТ, 2013.

Кларк Р., Нейк Р. *Третья мировая война. Какой она будет?* СПб.: Питер, 2011.

Курцвейл Р. Эволюция разума. М.: Эксмо, 2015.

Льюис М. MoneyBall. Как математика изменила самую популярную спортивную лигу в мире. М.: Манн, Иванов и Фербер, 2013.

Майер-Шенбергер В., Кукьер К. Большие данные. Революция, которая изменит то, как мы живем, работаем и мыслим. М.: Манн, Иванов и Фербер, 2013.

Маккормик Дж. Девять алгоритмов, которые изменили будущее. М.: ДМК Пресс, 2014.

Мандельброт Б. Фрактальная геометрия природы. Ижевск: Институт компьютерных исследований, 2002.

Марр Д. Зрение. Информационный подход к изучению представления и обработки зрительных образов. М.: Радио и связь, 1987.

Минский М., Пейперт С. Перцептроны. М.: Мир, 1971.

Петцольд Ч. [Читаем Тьюринга](#). Путешествие по исторической статье Тьюринга о вычислимости и машинах Тьюринга. М.: ДМК-Пресс, 2014.

Разумный мозг / под ред. Д. Эделмена и В. Маунткасла. М.: Мир, 1981.

Рассел С., Норвег П. Искусственный интеллект: современный подход. М.: Вильямс, 2000.

Саймон Г. [Науки об искусственном](#). М.: Едиториал УРСС, 2004.

Саттон Р., Барто Э. Обучение с подкреплением. М.: Бином, Лаборатория знаний, 2012.

Сеунг С. Коннектом. Как мозг делает нас тем, что мы есть. М.: Бином. Лаборатория знаний, 2014.

Сильвер Н. Сигнал и шум. Почему одни прогнозы сбываются, а другие — нет. М.: КоЛибри : Азбука-Аттикус, 2015.

Такман Б. Августовские пушки. М.: Астрель, 2012.

Талеб Н. [Черный лебедь](#). М.: КоЛибри: Азбука-Аттикус, 2015.

Фишер Р. С. Генетическая теория естественного отбора. Ижевск: Регулярная и хаотическая динамика. Институт компьютерных исследований, 2011.

Френкель Э. Любовь и математика. Сердце скрытой реальности. СПб.: Питер, 2016.

Хокинс Дж., Блейкли С. Об интеллекте. М.: Вильямс, 2016.

Хофштадтер Д. [Гёдель, Эшер, Бах](#): эта бесконечная гирлянда. Самара: Бахрах-М, 2001.