

ВВЕДЕНИЕ: ИСТОРИЯ И СТРУКТУРА AI И ML

Сергей Николенко

СПбГУ — Санкт-Петербург

02 сентября 2020 г.

Random facts:

- 2 сентября 1666 г. в Лондоне начался Великий пожар; за три дня сгорели 13500 домов и 87 церквей, в том числе Собор святого Павла
- В Британии и её колониях за 2 сентября 1752 г. последовало сразу 14 сентября
- 2 сентября 1967 г. отставной английский офицер Падди Рой Бейтс провозгласил создание независимого Княжества Силенд на морской платформе у берегов Англии
- 2 сентября 1972 г. хоккеисты сборной СССР со счётом 7:3 победили сборную Канады в первом матче Суперсерии
- 2 сентября 1990 г. II Чрезвычайный съезд народных депутатов всех уровней Приднестровья принял Декларацию об образовании Приднестровской Молдавской Советской Социалистической Республики (ПМССР) со столицей в Тирасполе; впрочем, в декабре Горбачёв всё отменил

ЧТО ТАКОЕ МАШИННОЕ ОБУЧЕНИЕ

- Гефест создавал себе роботов–андроидов, например, гигантского человекоподобного робота Талоса.
- Пигмалион оживлял Галатею.
- Иегова и Аллах — куски глины.
- Особо мудрые раввины могли создавать големов.
- Альберт Великий изготовил искусственную говорящую голову (чем очень расстроил Фому Аквинского).
- Начиная с доктора Франкенштейна, дальше AI в литературе появляется постоянно...

- AI как наука начался с *теста Тьюринга* (1950).
- Компьютер должен успешно выдать себя за человека в (письменном) диалоге между судьёй, человеком и компьютером.
- Правда, исходная формулировка была несколько тоньше и интереснее...

- Здесь уже очевидно, сколько всего надо, чтобы сделать AI:
 - обработка естественного языка;
 - представление знаний;
 - выводы из полученных знаний;
 - обучение на опыте (собственно machine learning).

- Термин AI и формулировки основных задач появились в 1956 на семинаре в Дартмуте.
- Его организовали Джон Маккарти (John McCarthy), Марвин Мински (Marvin Minsky), Клод Шеннон (Claude Shannon) и Натаниэль Рочестер (Nathaniel Rochester).
- Это была, наверное, самая амбициозная грантозаявка в истории информатики.

Мы предлагаем исследование искусственного интеллекта сроком в 2 месяца с участием 10 человек летом 1956 года в Дартмутском колледже, ГанOVER, Нью-Гемпшир. Исследование основано на предположении, что всякий аспект обучения или любое другое свойство интеллекта может в принципе быть столь точно описано, что машина сможет его симулировать. Мы попытаемся понять, как обучить машины использовать естественные языки, формировать абстракции и концепции, решать задачи, сейчас подвластные только людям, и улучшать самих себя. Мы считаем, что существенное продвижение в одной или более из этих проблем вполне возможно, если специально подобранная группа учёных будет работать над этим в течение лета.

- Оптимистическое время. Казалось, что ещё немного, ещё чуть-чуть...
- Allen Newell, Herbert Simon: *Logic Theorist*.
 - Программа для логического вывода.
 - Смогла передоказать большую часть *Principia Mathematica*, кое-где даже изящнее, чем сами Рассел с Уайтхедом.

- Оптимистическое время. Казалось, что ещё немного, ещё чуть-чуть...
- General Problem Solver – программа, которая пыталась думать как человек;
- Много программ, которые умели делать некоторые ограниченные вещи (microworlds):
 - Analogy (IQ-тесты на «выберите лишнее»);
 - Student (алгебраические словесные задачи);
 - Blocks World (переставляла 3D-блоки).

- Суть: накопить достаточно большой набор правил и знаний о предметной области, затем делать выводы.
- Первый успех: MYCIN – диагностика инфекций крови:
 - около 450 правил;
 - результаты как у опытного врача и существенно лучше, чем у начинающих врачей.

- Началось внедрение.
- Первый AI-отдел был в компании DEC (Digital Equipment Corporation);
- Утверждают, что к 1986 году он сэкономил DEC \$10 млн. в год;
- Бум закончился к концу 80-х, когда многие компании не смогли оправдать завышенных ожиданий.

- В последние десятилетия основной акцент сместился на машинное обучение и поиск закономерностей в данных.
- Особенно — с развитием интернета.
- Сейчас про AI в смысле трёх законов робототехники уже не очень вспоминают.
- // Но роботика — процветает и пользуется machine learning на каждом шагу.

- Что значит — обучающаяся машина? Как определить «обучаемость»?

- Что значит — обучающаяся машина? Как определить «обучаемость»?

Definition

Компьютерная программа обучается по мере накопления опыта относительно некоторого класса задач T и целевой функции P , если качество решения этих задач (относительно P) улучшается с получением нового опыта.

- Определение очень (слишком?) общее.
- Какие конкретные примеры можно привести?

- Мы будем рассматривать разные алгоритмы, которые решают ту ли иную задачу, причём решают тем лучше, чем больше начальных (тестовых) данных ему дадут.
- Сегодня мы поговорим об общей теории байесовского вывода, в которую обычно можно погрузить любой алгоритм машинного обучения.
- Но сначала – краткий обзор основных задач машинного обучения в целом.

- *Обучение с учителем* (supervised learning) – обучение, в котором есть некоторое число примеров с правильными ответами:
 - *обучающая выборка* (training set) – набор примеров, каждый из которых состоит из *признаков* (features, attributes);
 - у примеров есть правильные ответы – переменная (response), которую мы предсказываем; она может быть категориальная (categorical), непрерывная или ординальная (ordinal);

- *Обучение с учителем* (supervised learning) – обучение, в котором есть некоторое число примеров с правильными ответами:
 - модель *обучается* на этой выборке (training phase, learning phase), затем может быть применена к новым примерам (test set);
 - главное – обучить модель, которая не только точки из обучающей выборки объясняет, но и на новые примеры хорошо *обобщается* (generalizes);
 - иначе – оверфиттинг (overfitting);

- *Обучение с учителем* (supervised learning) – обучение, в котором есть некоторое число примеров с правильными ответами:
 - обычно нам дают просто обучающую выборку – как тогда проверить, обобщаются ли модели?
 - кросс-валидация – разбиваем выборку на тренировочный и валидационный набор (validation set);
 - перед тем как подавать что-то на вход, обычно делают предобработку, стараясь выделить из входных данных самые содержательные аспекты (feature extraction).

- *Обучение с учителем* (supervised learning) – обучение, в котором есть некоторое число примеров с правильными ответами:
 - *классификация*: есть некоторый дискретный набор категорий (классов), и надо новые примеры определить в какой-нибудь класс;
 - классификация текстов по темам, спам-фильтр;
 - распознавание лиц/объектов/текста;

- *Обучение с учителем* (supervised learning) – обучение, в котором есть некоторое число примеров с правильными ответами:
 - *регрессия*: есть некоторая неизвестная функция, и надо предсказать её значения на новых примерах:
 - инженерные приложения (предсказать температуру, положение робота, whatever);
 - финансы – предсказать цену акций;
 - то же плюс изменения во времени – например, распознавание речи.

- *Обучение без учителя* (unsupervised learning) – обучение, в котором нет правильных ответов, только данные:
 - *кластеризация* (clustering): надо разбить данные на заранее неизвестные классы по некоторой мере похожести:
 - выделить семейства генов из последовательностей нуклеотидов;
 - кластеризовать пользователей и персонализировать под них приложение;
 - кластеризовать масс-спектрометрическое изображение на части с разным составом;

- *Обучение без учителя* (unsupervised learning) – обучение, в котором нет правильных ответов, только данные:
 - *снижение размерности* (dimensionality reduction): данные имеют огромную размерность (очень много признаков), нужно уменьшить её, выделить самые информативные признаки, чтобы все вышеописанные алгоритмы смогли работать;
 - *дополнение матриц* (matrix completion): есть разреженная матрица, надо предсказать, что на недостающих позициях.
 - Часто даны правильные ответы для небольшой части данных – semi-supervised learning.

- *Обучение с подкреплением* (reinforcement learning) – обучение, в котором агент учится из собственных проб и ошибок:
 - *многорукие бандиты*: есть некоторый набор действий, каждое из которых ведёт к случайным результатам; нужно получить как можно больший доход;
 - *exploration vs. exploitation*: как и когда от исследования нового переходить к использованию того, что уже изучил;
 - *credit assignment*: конфетку дают в самом конце (выиграл партию), и надо как-то распределить эту конфетку по всем ходам, которые привели к победе.

- *активное обучение* (active learning) – как выбрать следующий (относительно дорогой) тест;
- *обучение ранжированию* (learning to rank) – ординальная регрессия, как породить упорядоченный список (интернет-поиск);
- *бустинг* (boosting) – как скомбинировать несколько слабых классификаторов так, чтобы получился хороший;
- *выбор модели* (model selection) – где провести черту между моделями с многими параметрами и с немногими.

- Во всех методах и подходах очень пригодится метод, который мог бы не просто выдавать ответ, а ещё оценивать, насколько модель уверена в этом ответе, насколько модель хорошо описывает данные, как изменятся эти величины при дальнейших экспериментах и т.д.
- Поэтому центральную роль в машинном обучении играет теория вероятностей – и мы тоже будем её активно применять.

- Christopher M. Bishop, *Pattern Recognition and Machine Learning*, Springer, 2007.
- Kevin Murphy, *Machine Learning: A Probabilistic Perspective*, MIT Press, 2013.
- Trevor Hastie, Robert Tibshirani, and Jerome Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2nd ed., Springer, 2009.

БАЙЕСОВСКИЙ ПОДХОД

- Нам не понадобятся математические определения сигма-алгебры, вероятностной меры, борелевских множеств и т.п.
- Достаточно понимать, что бывают дискретные случайные величины (неотрицательные вероятности исходов в сумме дают единицу) и непрерывные случайные величины (интеграл неотрицательной функции плотности равен единице).

ОСНОВНЫЕ ОПРЕДЕЛЕНИЯ

- *Совместная вероятность* – вероятность одновременного наступления двух событий, $p(x, y)$; маргинализация:

$$p(x) = \sum_y p(x, y).$$

- *Условная вероятность* – вероятность наступления одного события, если известно, что произошло другое, $p(x | y)$:

$$p(x, y) = p(x | y)p(y) = p(y | x)p(x).$$

- *Теорема Байеса* – из предыдущей формулы:

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

- *Независимость*: x и y независимы, если

$$p(x, y) = p(x)p(y).$$

- Приведём классический пример из классической области применения статистики — медицины.
- Пусть некий тест на какую-нибудь болезнь имеет вероятность успеха 95% (т.е. 5% — вероятность как позитивной, так и негативной ошибки).
- Всего болезнь имеется у 1% респондентов (отложим на время то, что они разного возраста и профессий).
- Пусть некий человек получил позитивный результат теста (тест говорит, что он болен). С какой вероятностью он действительно болен?

- Приведём классический пример из классической области применения статистики — медицины.
- Пусть некий тест на какую-нибудь болезнь имеет вероятность успеха 95% (т.е. 5% — вероятность как позитивной, так и негативной ошибки).
- Всего болезнь имеется у 1% респондентов (отложим на время то, что они разного возраста и профессий).
- Пусть некий человек получил позитивный результат теста (тест говорит, что он болен). С какой вероятностью он действительно болен?
- Ответ: 16%.

- Обозначим через t результат теста, через d — наличие болезни.
- $p(t = 1) = p(t = 1|d = 1)p(d = 1) + p(t = 1|d = 0)p(d = 0)$.
- Используем теорему Байеса:

$$\begin{aligned} p(d = 1|t = 1) &= \\ &= \frac{p(t = 1|d = 1)p(d = 1)}{p(t = 1|d = 1)p(d = 1) + p(t = 1|d = 0)p(d = 0)} = \\ &= \frac{0.95 \times 0.01}{0.95 \times 0.01 + 0.05 \times 0.99} = 0.16. \end{aligned}$$

- Вот такие задачи составляют суть вероятностного вывода (probabilistic inference).
- Поскольку они обычно основаны на теореме Байеса, вывод часто называют байесовским (Bayesian inference).
- Но не только поэтому.

БАЙЕСОВСКИЙ ПОДХОД

- Нам не понадобятся математические определения сигма-алгебры, вероятностной меры, борелевских множеств и т.п.
- Достаточно понимать, что бывают дискретные случайные величины (неотрицательные вероятности исходов в сумме дают единицу) и непрерывные случайные величины (интеграл неотрицательной функции плотности равен единице).

ОСНОВНЫЕ ОПРЕДЕЛЕНИЯ

- *Совместная вероятность* – вероятность одновременного наступления двух событий, $p(x, y)$; маргинализация:

$$p(x) = \sum_y p(x, y).$$

- *Условная вероятность* – вероятность наступления одного события, если известно, что произошло другое, $p(x | y)$:

$$p(x, y) = p(x | y)p(y) = p(y | x)p(x).$$

- *Теорема Байеса* – из предыдущей формулы:

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

- *Независимость*: x и y независимы, если

$$p(x, y) = p(x)p(y).$$

ПРЯМЫЕ И ОБРАТНЫЕ ЗАДАЧИ

- Прямая задача: в урне лежат 10 шаров, из них 3 чёрных. Какова вероятность выбрать чёрный шар?
- Или: в урне лежат 10 шаров с номерами от 1 до 10. Какова вероятность того, что номера трёх последовательно выбранных шаров дадут в сумме 12?
- Обратная задача: перед нами две урны, в каждой по 10 шаров, но в одной 3 чёрных, а в другой — 6. Кто-то взял из какой-то урны шар, и он оказался чёрным. Насколько вероятно, что он брал шар из первой урны?
- Заметьте, что в обратной задаче вероятности сразу стали байесовскими (хоть здесь и можно переформулировать через частоты).

- Иначе говоря, прямые задачи теории вероятностей описывают некий вероятностный процесс или модель и просят подсчитать ту или иную вероятность (т.е. фактически по модели предсказать поведение).
- Обратные задачи содержат *скрытые переменные* (в примере — номер урны, из которой брали шар). Они часто просят по известному поведению построить вероятностную модель.
- Задачи машинного обучения обычно являются задачами второй категории.

- Запишем теорему Байеса:

$$p(\theta|D) = \frac{p(\theta)p(D|\theta)}{p(D)}.$$

- Здесь $p(\theta)$ – *априорная вероятность* (prior probability), $p(D|\theta)$ – *правдоподобие* (likelihood), $p(\theta|D)$ – *апостериорная вероятность* (posterior probability), $p(D) = \int p(D | \theta)p(\theta)d\theta$ – *вероятность данных* (evidence).
- Вообще, *функция правдоподобия* имеет вид

$$a \mapsto p(y|x = a)$$

для некоторой случайной величины y .

- В статистике обычно ищут *гипотезу максимального правдоподобия* (maximum likelihood):

$$\theta_{ML} = \arg \max_{\theta} p(D | \theta).$$

- В байесовском подходе ищут *апостериорное распределение* (posterior)

$$p(\theta|D) \propto p(D|\theta)p(\theta)$$

и, возможно, *максимальную апостериорную гипотезу* (maximum a posteriori):

$$\theta_{MAP} = \arg \max_{\theta} p(\theta | D) = \arg \max_{\theta} p(D | \theta)p(\theta).$$

ПОСТАНОВКА ЗАДАЧИ

- Простая задача вывода: дана нечестная монетка, она подброшена N раз, имеется последовательность результатов падения монетки. Надо определить её «нечестность» и предсказать, чем она выпадет в следующий раз.
- Гипотеза максимального правдоподобия скажет, что вероятность решки равна числу выпавших решек, делённому на число экспериментов.

ПОСТАНОВКА ЗАДАЧИ

- Простая задача вывода: дана нечестная монетка, она подброшена N раз, имеется последовательность результатов падения монетки. Надо определить её «нечестность» и предсказать, чем она выпадет в следующий раз.
- Гипотеза максимального правдоподобия скажет, что вероятность решки равна числу выпавших решек, делённому на число экспериментов.
- То есть если вы взяли незнакомую монетку, подбросили её один раз и она выпала решкой, вы теперь ожидаете, что она всегда будет выпадать только решкой, правильно?
- Странно получается... давайте поговорим об этом поподробнее позже.

Спасибо за внимание!