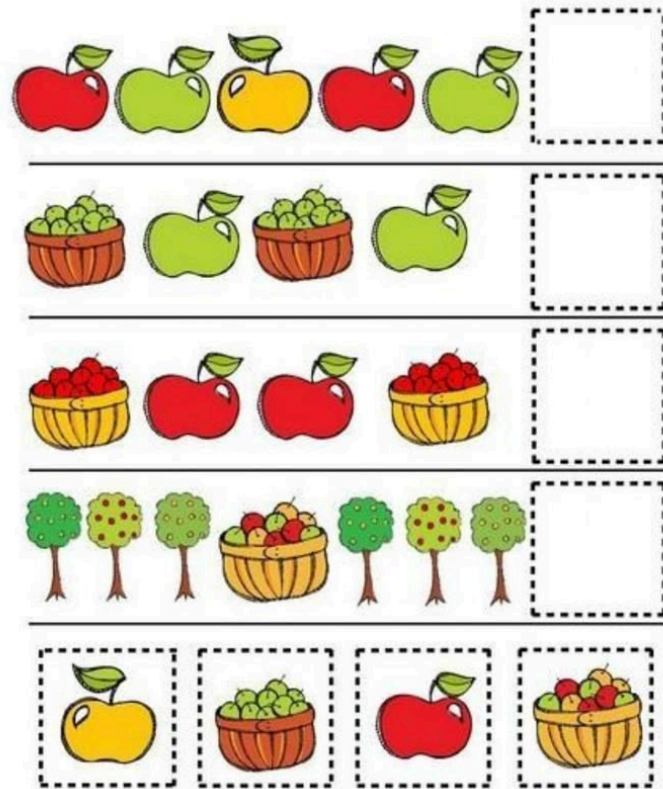


Индуктивный вывод Рея Соломонова

- 28, 31, 30, 33, 32, ...
- 50, 3, 47, 3, 44, ...



Задача: предсказать последовательность

abababab?

- $p(? = a \mid \text{abababab}) = ?$
- $p(? = b \mid \text{abababab}) = ?$

модель = вероятности $p(\text{следующий символ} \mid \text{предыдущие})$

Модель = мера на бесконечных строках

- Σ – алфавит, конечное множество
- Σ^* – конечные строки, $\Sigma^{\mathbb{N}}$ – бесконечные строки
- модель M : $p(a \mid s)$, $a \in \Sigma$, $s \in \Sigma^*$

Наблюдение. Модель определяет вероятностную меру на $\Sigma^{\mathbb{N}}$ с цилиндрической σ -алгеброй. И наоборот.

$$\mu(a_1 a_2 \dots a_n *) = p(a_1) p(a_2 | a_1) \dots p(a_n | a_1 \dots a_{n-1})$$

$a_1 a_2 \dots a_n * = \{s \in \Sigma^{\mathbb{N}} \mid s_{1:n} = a_1 a_2 \dots a_n\}$ – цилиндрическое множество

Мера на бесконечных строках $\approx$ мера на $[0;1]$

- символы Σ можно интерпретировать как цифры после запятой
- тогда $\Sigma^{\mathbb{N}}$ можно отождествить с $[0; 1]$
- цилиндрическая σ -алгебра = борелевская $\mathcal{B}$

Пример. Мера Лебега на $[0; 1]$ = модель подбрасывания монетки,
т.е. $p(a \mid s) = \frac{1}{|\Sigma|}$

Нюанс: строки $0.0999\dots$ и $0.1000\dots$ разные, а числа одинаковые.

Мера на $[0;1] = \uparrow$ функция $[0; 1] \rightarrow [0; 1]$

- вероятностная мера μ на $[0; 1]$ задаёт функцию распределения $f(x) = \mu([0; x))$
- f измерима, неубывает и непрерывна слева
- и наоборот, любая такая функция однозначно определяет свою меру μ

Известно, что такая f имеет не более чем счётное число точек разрыва. Эти точки соответствуют бесконечным строкам $s \in \Sigma^{\mathbb{N}}$, для которых $\mu(\{s\}) > 0$.

Вывод

Следующие вещи эквивалентны:

- модель предсказания следующего символа
- вероятностная мера на $\Sigma^{\mathbb{N}}$

И это почти то же, что и:

- вероятностная мера на $[0; 1]$
- измеримая неубывающая непрерывная слева функция $[0; 1] \rightarrow [0; 1]$

Связь с ML

- пусть мы хотим по x предсказывать y
- имеем датасет $D = \{(x_i, y_i)\}$
- запишем x_i и y_i битами и положим на ленту
- положим на ленту новый x , и модель выдаст возможные варианты y

$$x_1 y_1 \dots x_n y_n x \ ?$$

- хотим учесть независимость разных $(x_i, y_i) \Rightarrow$ берём модель, инвариантную отн. перестановок блоков $x_i y_i$ в строке

Как сравнивать модели?

- предсказываем один символ $\Rightarrow$ лучше та модель, чья вероятность $p(a \mid s)$ оказалась больше
- предсказываем строку $\Rightarrow$ лучше та модель, чья вероятность $p(s)$ оказалась больше

В идеале хотелось бы дать такое определение:

- модель M_1 не хуже M_0 , если $\forall s \in \Sigma^* p_1(s) \geq p_0(s)$
- проблема: сумма вероятностей равна 1, поэтому если одна модель лучше на одних строках, то обязательно хуже на других

Определение. Мера μ_1 мажорирует меру μ_0 ($\mu_1 \succcurlyeq \mu_0$), если $\exists C > 0$, что $\forall$ измеримого X выполнено $\mu_1(X) \geq C\mu_0(X)$.

Равносильно: $\exists C \in \mathbb{R}$, что $\forall$ измеримого X выполнено
 $-\log(\mu_1(X)) \leq -\log(\mu_0(X)) + C$

Модель M_1 не хуже модели M_0 ($M_1 \succcurlyeq M_0$), если $\mu_1 \succcurlyeq \mu_0$.

Интуиция $M_1 \succcurlyeq M_0$:

- M_1 присваивает вероятности, не сильно меньшие, чем M_0
- таким образом, M_1 является более (или такой же) общей, т.к. рассматривает все варианты, которые рассматривает M_0

С точки зрения ML:

- функция потерь языковой модели (кросс-энтропия) для M_1 не более чем на константу больше чем у M_0

С точки зрения информации:

- M_1 позволяет получить длину сжатой строки не более чем на константу бит хуже чем позволяет M_0

Связь с производной Радона-Никодима

Теорема Радона-Никодима. Пусть мера ν абсолютно непрерывна относительно меры μ [т.е. из $\mu(A) = 0$ следует $\nu(A) = 0$]. Тогда существует измеримая $f : \dots \rightarrow [0; +\infty)$, что $\nu(A) = \int_A f(x) \mu(dx)$. Эта f называется *производной Радона-Никодима* и обозначается как $f = \frac{d\nu}{d\mu}$.

Утверждение. $\mu_1 \succcurlyeq \mu_0 \Leftrightarrow \frac{d\mu_0}{d\mu_1}$ существует и ограничена.

Утверждение. Для любой строки $s \in \Sigma^* \cup \Sigma^{\mathbb{N}}$ модель M совершит в среднем не больше ошибок в предсказании следующего символа, чем $-\ln p(s)$.

$$\begin{aligned}\mathbb{E}[\text{число ошибок на } s] &= \\ \sum_{a \in s} \mathbb{E}[\text{число ошибок в предсказании символа } a] &= \\ \sum [1 - p(s_{i+1} \mid s_{1:i})] &\leq \\ \sum [-\ln p(s_{i+1} \mid s_{1:i})] &= -\ln p(s)\end{aligned}$$

Вывод: пусть $s \in \Sigma^{\mathbb{N}}$ и $\mu(s) > 0$. Тогда ожидаемое число ошибок предсказаний модели на s ограничено, несмотря на бесконечность строки. При этом для любой $\mu_1 \succcurlyeq \mu$ число ошибок тоже ограничено.

Как сконструировать модель лучше?

- пусть имеем набор моделей $\mathcal{M} = \{M_1, M_2, \dots\}$
- существует ли модель M , которая не хуже каждой из них: $M \succcurlyeq M_i$?
- да! Подойдёт любая смесь семейства $\mathcal{M}$ с весами: $\mu = \sum w_i \mu_i$, где $w_i > 0$ и $\sum w_i = 1$.

Утверждение. *Любая смесь моделей с ненулевыми весами не хуже каждой из них.*

Ничего не напоминает?

Байесовский вывод = смесь моделей

Утверждение. Рассмотрим семейство моделей $\mathcal{M} = \{M_1, M_2, \dots\}$, и введём над ними априорное распределение $p(M_i)$. Тогда получив строку s можно найти апостериорное распределение $p(M_i \mid s)$, и вычислить предсказательное распределение $p(a \mid s) = \sum p(a \mid s, M_i)p(M_i \mid s)$. Тогда модель M , получающаяся из предсказательного распределения, является смесью семейства $\mathcal{M}$ с априорными весами: $\mu = \sum p(M_i)\mu_i$.

Байесовский вывод = построение модели не хуже каждой рассматриваемой.

Доказательство.

- для любой строки s : $p_{\text{смеси}}(s) = \sum p(s \mid M_i)p(M_i)$

Распишем байесовское предсказательное распределение:

$$p(a \mid s) = \sum p(a \mid s, M_i)p(M_i \mid s) = \frac{\sum p(a \mid s, M_i)p(s \mid M_i)p(M_i)}{\sum p(s \mid M_i)p(M_i)} =$$
$$\frac{\sum p(sa \mid M_i)p(M_i)}{\sum p(s \mid M_i)p(M_i)} = \frac{p_{\text{смеси}}(sa)}{p_{\text{смеси}}(s)} = p_{\text{смеси}}(a \mid s)$$

Пример: байесовский вывод для монетки

- пусть $\Sigma = \{0; 1\}$
- $\bowtie$ семейство моделей $M_\theta: \begin{cases} p(0 \mid s, \theta) = \theta \\ p(1 \mid s, \theta) = 1 - \theta \end{cases}$
- вероятность каждого символа фиксирована и не зависит от наблюдаемой строки
- $\bowtie \mu := \int_{\theta \in [0;1]} \mu_\theta d\theta$ – это байесовская смесь семейства $\{M_\theta\}$

Утверждение. Для этой смеси распределение следующего символа такое: $p(a \mid s) = \frac{k_a + 1}{n + 2}$.

[$n = |s|$ – длина строки, k_a – количество символов a в строке s]

Мораль: смесь даёт более умную модель!

Существует ли лучшая модель?

- проблема: можем брать смесь лишь **счётного** семейства
- всевозможные модели образуют несчётное семейство!
(=измеримые неубывающие непр.слева функции $[0; 1] \rightarrow [0; 1]$)

Упражнение. Не существует вероятностной меры на $[0; 1]$, которая мажорирует любую другую такую меру.

- *Лучшей модели среди всех нет!*
- Но нас интересуют не все модели. В более узких семействах лучшая модель может существовать.

- нас точно не интересуют *невычислимые* модели
- а вот вычислимых моделей счётное количество (т.к. компьютерных программ счётное кол-во)

Возьмём смесь всех вычислимых моделей.

- она *не хуже* любой вычислимой модели!
- является ли эта смесь вычислимой?

Вычислимые модели

Модель можно задать двумя способами:

- программой сэмплирования
- программой, вычисляющей вероятности строк

Проблема остановки: не существует программы, которая принимает код произвольной программы и отвечает на вопрос, остановится ли та.

Определение меры через сэмплирование

Программа сэмплирования (ПС) – это бесконечно работающая программа, содержащая такие функции:

- `random_bit()` – возвращает случайный бит
- `write(символ)` – выводит следующий символ

Наблюдение. Любая ПС задаёт распределение над $\Sigma^{\mathbb{N}} \cup \Sigma^*$.

Программа может уйти в бесконечный цикл и напечатать только конечную строку, поэтому $\cup \Sigma^*$.

Вычисление вероятностей строк

Определение. Мера μ над $\Sigma^{\mathbb{N}} \cup \Sigma^*$ вычислима, если $\exists$ программа $f : \Sigma^* \times \mathbb{Q}_{>0} \rightarrow \mathbb{Q}$, что $\forall s \in \Sigma^*$ и $\varepsilon \in \mathbb{Q}_{>0} |\mu(s *) - f(s, \varepsilon)| < \varepsilon$.
Т.е. меру можно вычислить с любой заранее заданной точностью.

Определение. Мера μ над $\Sigma^{\mathbb{N}} \cup \Sigma^*$ полувычислима, если $\exists$ программа $f : \Sigma^* \times \mathbb{N} \rightarrow \mathbb{Q}$, что $\forall s \in \Sigma^* f(s, t) \xrightarrow{t \rightarrow \infty} \mu(s *)$.
Т.е. мы можем сколь угодно точно вычислить μ увеличивая число итераций t .

Утверждение. Мера μ вычислима $\Leftrightarrow$ она полувычислима, и функция $N(\varepsilon)$ из определения предела вычислима.

Определение. Мера μ над $\Sigma^{\mathbb{N}} \cup \Sigma^*$ полувычислима снизу (в литературе: *перечислима снизу*), если существует программа $f(s, t) : \Sigma^* \times \mathbb{N} \rightarrow \mathbb{Q}$, что $\forall s \in \Sigma^*$ значение f :

- неубывает: $f(s, 0) \leq f(s, 1) \leq \dots$
- стремится к $\mu(s *)$: $\lim_{t \rightarrow \infty} f(s, t) = \mu(s *)$

вычислимые меры $\subset$ полувычислимые снизу $\subset$
полувычислимые

Связь сэмплирования и вычисления вероятности

Теорема 1. Вероятностная мера μ на $\Sigma^{\mathbb{N}} \cup \Sigma^*$: задаётся некоторой ПС $\Leftrightarrow \mu$ полувывчислима снизу.

Теорема 2. Вероятностная мера μ на $\Sigma^{\mathbb{N}}$: задаётся некоторой ПС (мера задана только на бесконечных строках, т.е. вероятность заиклиться и выдать конечную строку должна быть равна 0!) $\Leftrightarrow \mu$ вычислима.

*Строгие доказательства см. в теоремах 75, 76 и 77 из книги
“Колмогоровская сложность и алгоритмическая случайность” (2012)
– Н. К. Верецагин, В. А. Успенский, А. Шень*

Лучшая модель

Теорема. Среди мер на $\Sigma^{\mathbb{N}} \cup \Sigma^*$, задаваемых ПС ($\Leftrightarrow$ полувывчислимых снизу), существует наибольшая, т.е. мажорирующая все остальные. Она называется универсальной мерой.

Доказательство. Построим такую ПС:

1. через `random_bit()` она сэмплирует код случайной ПС
2. эмулирует выбранную ПС

Если ПС выбирается с вероятностью w , то эта w и будет константой мажорирования мер.

Универсальная мера порождается универсальной машиной Тьюринга, которой на вход в `random_bit()` подаётся случайная бесконечная битовая строка.

- универсальных мер много, и все они мажорируют друг друга

Определение. (Колмогоровская) априорная сложность строки:
 $KA(s) := -\log_2 \mu(s *)$, где μ – универсальная мера.

- $KA(s)$ определена с точностью до аддитивной константы

Теорема. Любая мера, мажорирующая все вычислимые меры, сама не может являться вычислимой!

Доказательство аналогично доказательству невычислимости обычной колмогоровской сложности. (это теорема 6 в книге Верещагина)

Оно проводится на основании парадокса Берри: “наименьшее натуральное число, которое нельзя описать менее чем 20-ю словами”.

Если предположить, что такой алгоритм существует, то Колмогоровская сложность не будет наименьшей длиной описания, что противоречит её определению.

Детерминированные ПС

- Оказывается, универсальную меру можно получить и из детерминированных ПС!
- В общем случае мы могли брать любые ненулевые веса
- Если используем только детерминированные ПС, то веса нужны определённые:

$$w_p = 2^{-|p|}$$

где p – код программы для любой префиксно-свободной машины Тьюринга.

Теорема. Пусть U – любая префиксно-свободная машина Тьюринга. Пусть $M(s) := \sum_{p: U(p) \text{ начинается на } s} 2^{-|p|}$. Тогда для любой вероятностной полувывчислимой снизу меры μ над $\Sigma^* \cup \Sigma^{\mathbb{N}}$ существует константа $C > 0$, что $\forall s \in \Sigma^*$ справедливо $M(s) \geq C\mu(s *)$.

Индуктивный вывод Соломонова: алгоритм

$$p(a \mid s) = \frac{\sum_{p: \text{программа } p \text{ генерирует строку 'sa'}} 2^{-|p|}}{\sum_{p: \text{программа } p \text{ генерирует строку 's'}} 2^{-|p|}}$$

- суммы сверху и снизу приближаются перебором всё большего количества компьютерных программ на всё большем времени вычислений
- в пределе это даст оптимальное предсказание – лучшее, которое можно получить каким-либо алгоритмом

Итоги

- существует модель предсказания последовательности, которая *не хуже* любой вычислимой модели
- любая такая модель является невычислимой из-за проблемы остановки
- однако среди таких моделей есть полувывислимая, т.е. мы можем сколь угодно точно вычислить её вероятности, но не с заранее заданной точностью

Интуиция: какой бы среда ни была, универсальная модель сделает число ошибок равное $O(\text{длина её описания})$, и полностью её выучит.

Индуктивный вывод

- следствия $\longrightarrow$ предпосылки
- наука и ML:
 - следствия = наблюдения
 - предпосылки = гипотезы о мире или данных
- универсальный алгоритм – полный перебор компьютерных программ (= вывод Соломонова):
 - даст наилучший прогноз
 - требует ∞ вычислений
- есть практически полезные алгоритмы

Дедуктивный вывод

- предпосылки $\longrightarrow$ следствия
- математика:
 - предпосылки = аксиомы + правила логики
 - следствия = теоремы
- универсальный алгоритм – полный перебор всех доказательств
 - докажет любую теорему (если доказательство существует)
 - требует ∞ вычислений
- нет практически полезных алгоритмов

Индуктивный вывод на практике

- любой алгоритм обучения = распределение на $\Sigma^{\mathbb{N}}$
- универсальный алгоритм обучения невычислим
- хотим построить хороший алгоритм обучения

Инструменты:

- байесовская смесь – гарантированное улучшение, но вычислительно дорого
- всё остальное ML ;)

SGD задаёт меру на $\Sigma^{\mathbb{N}}$

- зададим нейросетью $p_{\theta}(y|x)$
- получив новый x в процессе обучения, сначала предсказываем $p_{\theta}(y|x)$, потом обновляем параметры $\theta \leftarrow \text{lr} * \text{grad}$

Так мы задаём меру на $\Sigma^{\mathbb{N}}$:

- биты x предсказываем равномерно (они нас не интересуют)
- биты y предсказываем согласно $p_{\theta}(y|x)$, где θ – результат SGD на всей прошлой истории

$$x_1 y_1 \dots x_n y_n x \ ?$$

Индуктивный вывод vs no-free-lunch

- индуктивный вывод: *универсальный алгоритм обучения есть*
- NFL теоремы: *универсального алгоритма нет, и вообще все алгоритмы в среднем одинаковы*

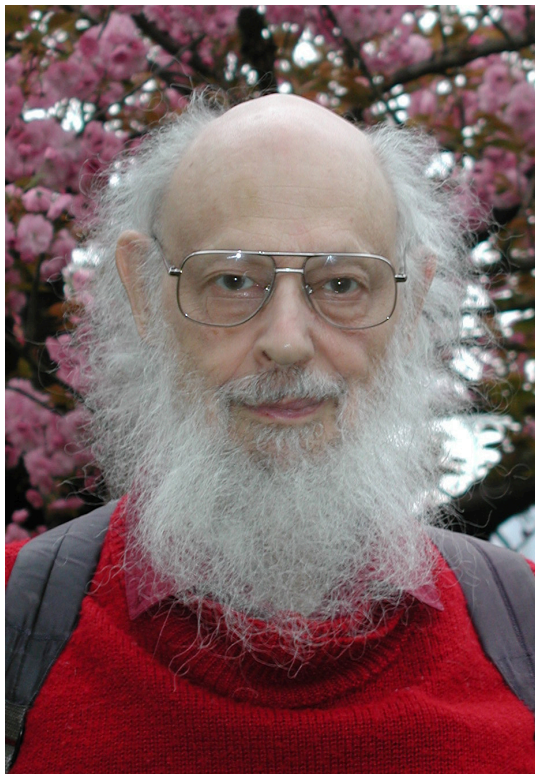
Причина:

- индуктивный вывод рассматривает обучение при произвольных размерах данных
- NFL рассматривает обучение при датасете фиксированного размера

Литература

- лучшая книга где есть всё: “An Introduction to Kolmogorov Complexity and Its Applications” – *Ming Li, Paul Vitányi*
- введение в теорию: “A Philosophical Treatise of Universal Induction” (2011) – *Samuel Rathmanner, Marcus Hutter*
- строгая математика, без акцента на индуктивный вывод: “Колмогоровская сложность и алгоритмическая случайность” (2013) – *Н. К. Верещагин, В. А. Успенский, А. Шень*
- основополагающая работа: “A Formal Theory of Inductive Inference” (Parts I-II, 1964) – *Ray Solomonoff*

Ray Solomonoff (1926-2009)



Леонид Левин (род. 1948)



Присоединяйтесь к клубу!

если вас заинтересовало 😊

клуб Соломонова

- делимся книгами / статьями
- находим людей, которым это тоже интересно!

