ОБ ОДНОМ ПРИМЕРЕ ОШИБОЧНОГО ОБУЧЕНИЯ МОДЕЛИ БАЙЕСОВСКОЙ ЛИНЕЙНОЙ РЕГРЕССИИ

А. Н. Курганский[1], И. В. Максименко[1,2]

[1]*Институт прикладной математики и механики, Донецк, Россия*
[2]*Донецкий государственный университет, Россия*
E-mail: topologia@mail.ru, igor.maksimenko_1967@mail.ru

Статья посвящена вопросу корректного применения математических методов в машинном обучении. Важно убедиться, что предложенный метод действительно подходит для решения выбранной задачи. Авторы делятся своим опытом неправильного применения байесовской линейной регрессии к конкретному случаю. Они построили вырожденный пример, который демонстрирует потенциальные проблемы при использовании байесовской линейной регрессии. Этот факт показывает, в байесовской линейной регрессии может быть важен порядок использования обучающих данных.

ONE EXAMPLE OF FAULTY TRAINING BAYESIAN LINEAR REGRESSION MODEL

A. N. Kurganskii, I. I. Maksimenko

The article is devoted to the issue of correct application of mathematical methods in machine learning. It is important to make sure that the proposed method is suitable for the solving of the chosen problem. The authors share their experience of incorrect application of Bayesian linear regression to a specific case. They built a degenerate example that demonstrates potential problems when using Bayesian linear regression. This fact shows that the order of using training data can be important in Bayesian linear regression.

Применение теории на практике требует понимания соответствия целей и границ применимости теории к анализируемому фрагменту реальности. Кажется важным в смысле доставляемых практических преимуществ, при прочих равных, отдавать себе отчёт, вызвано ли стремление применить конкретный метод для решения конкретной практической задачи популярностью метода или действительным соответствием инструмента решаемой задаче. Идея этой заметки возникла у авторов в результате их собственного неосторожного применения иерархической [1, с. 219] модели байесовской линейной регрессии [2, с. 127] к практической задаче. Цель данного сообщения – показать на примере, что встречающиеся определения байесовской линейной регрессии недостаточно полно объясняют условия ее применимости и вводят в заблуждение даже опытных пользователей математики и программирования.

Для краткости мы продемонстрируем ошибочное применение байесовской линейной регрессии на одном вырожденном примере. Вырожденные примеры возможно не появляются на практике, но высвечивают потенциальные проблемы. Предположим, требуется установить линейную зависимость между двумя переменными x и y и известно, что такая зависимость есть, а именно что

$y=ax+b+\varepsilon$, где случайная величина ε имеет равномерное распределение с математическим ожиданием 0 и неизвестной дисперсией.

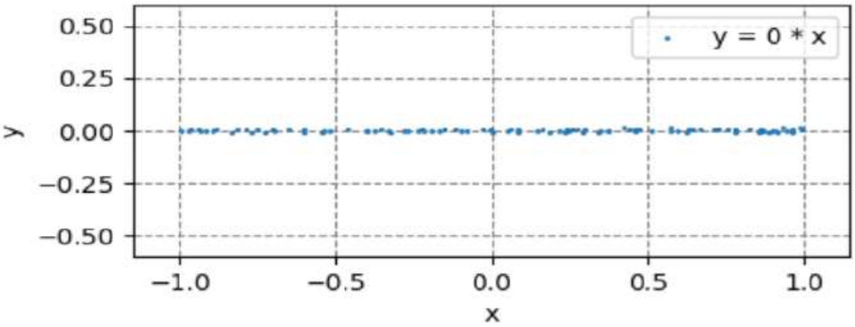

Рис. 1. Обучающая выборка T

Пример обучающего набора T показан на рис. 1 и представляет собой выборку размера 100 двумерной случайной величины (x,y), где x и y независимы, x имеет равномерную функцию распределения с параметрами $loc=-1$, $scale=2$, а y имеет равномерную функцию распределения с параметрами $loc=-\sigma$, $scale=2\sigma$ [2], где σ близко или равно 0. Другими словами, y является вырожденным случаем равномерно распределенной случайной величины и по сути не является случайной. Решение линейной регрессии $y=ax+b$ методом наименьших квадратов для обучающего набора данных T очевидно приведёт к: $a=b=0$.

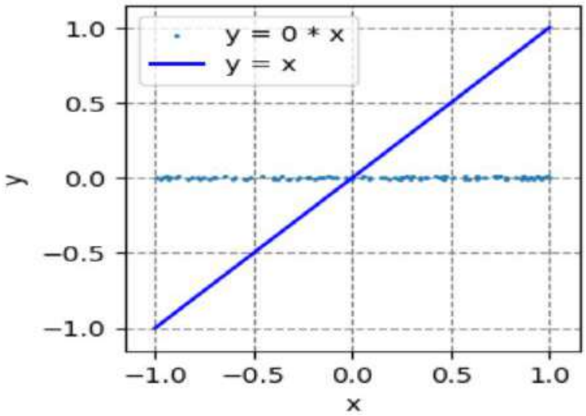

Рис. 2. Одно из правдоподобных решений модели байесовской линейной регрессии применительно к обучающему набору данных T.

Модель байесовской линейной регрессии рассматривает коэффициенты a и b как случайные величины. Для простоты предположим, что априори известно $b=0$ и линейная зависимость имеет вид $y=ax+\varepsilon$. Положим, что $a=tg(\varphi)$, где случайная величина φ априори равномерно распределена на следующем интервале $(-\pi/2, \pi/2)$, то есть мы ничего не знаем о a. Далее предположим, что величина σ, определяющая границы $[-\sigma, \sigma]$ значений величины ε равномерно распределена, к примеру, на интервале $[0, 2]$. Было бы естественным ожидать, что в результате обучения модели на выборке T апостериорные распределения величин как φ, так и σ наиболее плотны в точках близких к 0, а исходные данные T противоречили бы любым другим итогам обучения. Но простые рассуждения показывают, что эти ожидания ошибочны.

Согласно модели: $\varepsilon = y\text{-}ax$. Предположим φ=π/4, а значит $\varepsilon = y\text{-}x$. Из определения синтетического обучающего набора T={(x_i, y_i)}, где, напомним, $y_i=0$, а x_i представляют собой выборку равномерно распределенной случайной величины на интервале [-1,1], следует, что потенциально возможный результат байесовского вывода о том, что φ≈π/4 и $\sigma \approx 1$ (рис. 2) не противоречил бы обучающей выборке T, так как для неё $\varepsilon = x$. Аналогичные рассуждения показывают, что обучающей выборке T не противоречит любое решение для σ и φ, если $\sigma \approx |tg(\varphi)|$. Результаты соответствующих формальных вычислений в рамках байесовского моделирования апостериорной совместной плотности распределения для σ и φ показаны на рис. 3 в форме тепловой карты. Очевидно, что этот вводящий в заблуждение результат не соответствует практическим ожиданиям.

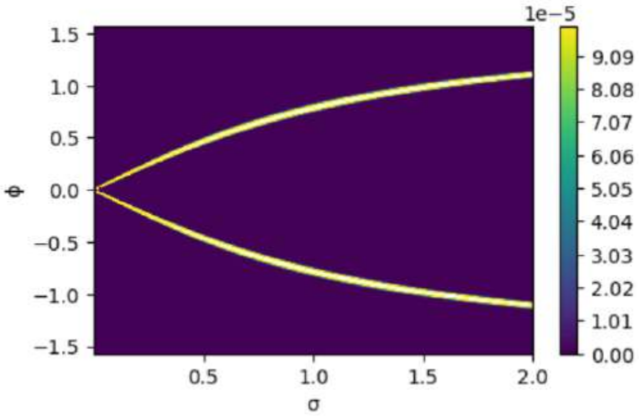

Рис. 3. Тепловая карта совместной апостериорной плотности распределения параметров σ и φ модели байесовской линейной регрессии

Однако, если разбить $T=T_0 \cup T_1$ в нашем случае на две непересекающиеся обучающие выборки $T_0 = \{(x_i, y_i)|\ x_i \leq 0\}$ и $T_1 = \{(x_i, y_i)|\ x_i > 0\}$, и обучить модель $y=ax+b+\varepsilon$ в два этапа: сначала на выборке T_0, затем на выборке T_1, то очевидно результат обучения будет соответствовать ожиданиям от модели доставлять практически верные результаты.

Решение, представленное на рис. 3, получено авторами в результате программирования на языке Python байесовской линейной регрессии «вручную», непосредственно исходя из математической модели, то есть без использования специализированных библиотек вероятностного программирования. Ниже уже представлен готовый к запуску в подходящей среде код модели рассматриваемого в работе примера на Python, использующий библиотеку вероятностного программирования PyMC v5.16 [4]. Библиотека PyMC специально предназначена для моделирования байесовского вывода. Как и ожидалось, обучение PyMC модели не сходится.

```
import pymc as pm
import numpy as np
from numpy.random import default_rng
from scipy.stats import uniform

rng = default_rng(0)
x = uniform.rvs(loc=-1, scale=2, size=2000, random_state=rng)
y = 0*x+uniform.rvs(loc=-0.001, scale=2*0.001, size=2000, random_state=rng)
with pm.Model() as model_blr:
    sigma = pm.Uniform('sigma', lower=0.0001, upper=2)
    phi = pm.Uniform('phi', lower=-np.pi/2, upper=np.pi/2)
    a = pm.Deterministic('a', pm.math.tan(phi))
    likelihood = pm.Uniform(
        'Y',
        lower=a*x-sigma,
        upper=a*x+sigma,
        observed=y
    )
    traces_s = pm.sample(1000, chains=4, random_seed=rng)
```

Листинг. PyMC-модель рассматриваемого примера

Неформальным следствием рассуждений и фактов выше, является важность порядка, последовательности использования обучающих данных в байесовской линейной регрессии.

Работа выполнена при финансовой поддержке Минобрнауки России в рамках научной темы "Разработка и совершенствование интеллектуальных методов классификации и прогнозирования для задач распознавания образов и моделирования информационных процессов" (Регистрационный номер 1023111000141-9-1.2.1).

СПИСОК ЛИТЕРАТУРЫ

1. *Kruschke J. K.* Doing Bayesian Data Analysis. A Tutorial with R, JAGS, and Stan, Second Editon, 2014.
2. *Гудфеллоу Я., Бенджио И., Курвилль А.* Глубокое обучение / пер. с анг. А. А. Слинкина. 2-е изд., испр. М. : ДМК Пресс, 2018. 652 с.
3. NIST/SEMATECH e-Handbook of Statistical Methods [Электронный ресурс]. URL: http://www.itl.nist.gov/div898/handbook/eda/section3/eda364.htm (дата обращения: 29.09.2024).
4. PyMC. [Электронный ресурс]. URL: https://www.pymc.io/ (дата обращения: 29.09.2024).