半群学习

张俊东

2024 年 9 月 13 日

1 引言

抽象代数是数学的一个分支，主要研究代数结构的性质和运算规则。在过去，数学家将注意力集中在如何去分析一些已知的、规则明确的、性质良好的代数系统，如群、环、域等。在这篇文章中，我们会讨论这样一个有趣的问题，我们是否可以运用机器学习的方法来从数据中学习出一个我们尚不明确的代数系统?

2 半群

半群是一个集合配备了一种运算，这种运算满足以下基本条件：

1. **封闭性**：对于半群中的任意两个元素 a 和 b，运算 $a \cdot b$ 也在半群中。
2. **结合律**：对半群中的任意元素 a、b 和 c，有 $(a \cdot b) \cdot c = a \cdot (b \cdot c)$。
3. **单位元素**：存在一个单位元素 e 使得对半群中任意元素 a，有 $e \cdot a = a \cdot e = a$。

研究半群有非常多的好处，结合律的存在有利于进行高效的并行规约运算，可以帮助我们构建序列模型。

3 数据集

我们将要被建模的数据集进行如下描述，它是一个符号序列分类的问题。

Definition 1 所有可以在数据集中合法出现的符号的集合被符号集，记为 C 。它是一个离散的、有限的集合。

Definition 2 一个序列 x 被定义为由 C 中元素排列而成的，任意正整数长度的字符串。

例如，如果 $C = \{a, b, c\}$，合法的序列可能就会包括 $abc, aaabc, bcabc$ 等。在自然界中，会有这样一些序列映射的问题。如，我们会观察到某些 DNA 序列会展现某一性状，而另外一些 DNA 序列则不会展现这一性状；又例如，我们可以将一些时序决策的问题转化为序列分类的问题，如要求智能体在经历了某些关联事件后做出 A 选项，否则做出 B 选项等。

Definition 3 一个序列的映射 f 是从序列 x 到离散标签 y 的函数。将所有合法标签的集合记为 Y。

一个数据集就是有限对 (x, y) 的集合，记为 D，本文的目标就是利用机器学习的方法学习出泛化性好的映射 $\hat{f}$ 。

4 Setting

Definition 4 由数据集 D 定义的半群的作用集记为 T，它是这样被递归定义的：

1. 对于 $\forall c \in C$，有 $c \in T$。
2. 对于 $\forall a, b \in T$，有 $a \cdot b \in T$。

Theorem 5 对于任意的一个序列 $x = x_1 x_2 \cdots x_n$，可以唯一确定一个作用 $t \in T$。使得 $t = \prod_x x_i$。

这个命题的逆命题不成立，即对于一个作用 t 而言，它可以对应多个序列，而一个序列只能对应 T 中的一个元素。至于哪些序列公用一个 t，那是我们比较关心的问题，它反应了数据集所描绘的系统中所蕴含的结构信息。

Theorem 6 作用集 T 是一个有限集。

Definition 7 定义一个表现函数 $\phi : T \to Y$。它将一个作用元素 t 映射为一个标签 y。

标签 y 代表了外部对于系统作用结果的观察，不同的系统状态可能给予相同的 y，但这不代表这些序列对于将来输入会有一致的响应。

Definition 8 *对于数据集 D 而言，我们定义它的半群 G 为：*

$$G := (C, Y, D, T, \cdot, \phi)$$

其中对于 $\forall(x, y) \in D$ ，有

$$y = \phi(\prod_{x} x_i)$$

请注意，在半群的乘法运算中，往往是不满足交换律的。

5 布尔矩阵

Definition 9 *布尔代数系统 β 可以这样来描述，它的数域只有 0 和 1 两个值，并且两个双目算子“与”和“或”，和一个单目算子“非”*

$$\beta := (\{0, 1\}, \wedge, \vee, \neg) \tag{1}$$

$$a \wedge b = min(a, b) \tag{2}$$

$$a \vee b = max(a, b) \tag{3}$$

$$\neg a = 1 - a \tag{4}$$

当我们将布尔变量以矩阵的形式组织起来的时候，那么它就是一个布尔矩阵了。布尔矩阵与经典的复数矩阵有非常多相似之处。

Definition 10 我们定义两个布尔矩阵 A, B 的乘积为

$$A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1q} \\ a_{21} & a_{22} & \cdots & a_{2q} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mq} \end{pmatrix} \tag{5}$$

$$B = \begin{pmatrix} b_{11} & b_{12} & \cdots & b_{1n} \\ b_{21} & b_{22} & \cdots & b_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ b_{q1} & b_{q2} & \cdots & b_{qn} \end{pmatrix} \tag{6}$$

$$A \times B = \begin{pmatrix} \bigcup_q a_{1i} \wedge b_{i1} & \bigcup_q a_{1i} \wedge b_{i2} & \cdots & \bigcup_q a_{1i} \wedge b_{in} \\ \bigcup_q a_{2i} \wedge b_{i1} & \bigcup_q a_{2i} \wedge b_{i2} & \cdots & \bigcup_q a_{2i} \wedge b_{in} \\ \vdots & \vdots & \ddots & \vdots \\ \bigcup_q a_{mi} \wedge b_{i1} & \bigcup_q a_{mi} \wedge b_{i2} & \cdots & \bigcup_q a_{mi} \wedge b_{in} \end{pmatrix} \tag{7}$$

Theorem 11 布尔矩阵的乘法满足分配律和结合律。

当布尔矩阵的行数和列数相等时，我们说这是一个布尔方阵。

Theorem 12 布尔方阵和布尔矩阵乘法运算构成一个半群，它是封闭的。

Theorem 13 布尔方阵是可逆的，当且仅当它是一个置换矩阵。

Theorem 14 可以使用布尔方阵和布尔矩阵乘法运算来表达一个置换群。

6 实数逻辑运算

Definition 15 实数上的逻辑代数系统被定义为：

$$\gamma := (\mathbb{R}, \wedge, \vee, \neg) \tag{8}$$

$$a \wedge b = -\log(e^{-a} + e^{-b}) \tag{9}$$

$$a \vee b = \log(e^{a} + e^{b}) \tag{10}$$

$$\neg a = -a \tag{11}$$

Theorem 16 在 γ 中，有 $a \vee b = \neg(\neg a \wedge \neg b)$。

Theorem 17

$$\frac{1}{1+e^{-a\wedge b}} \leq \min(\frac{1}{1+e^{-a}}, \frac{1}{1+e^{-b}})$$

当且仅当 $\min(a,b) \to -\infty$ 时，等号成立

Theorem 18

$$\frac{1}{1+e^{-a\vee b}} \geq \max(\frac{1}{1+e^{-a}}, \frac{1}{1+e^{-b}})$$

当且仅当 $\max(a,b) \to \infty$ 时，等号成立

Theorem 19

$$\frac{1}{1+e^{-\neg a}} = 1 - \frac{1}{1+e^{-a}}$$

sigmoid 函数可以将任意实数映射到 0 和 1 之间，我们在 γ 上使用 sigmoid 函数进行映射可以表现出许多与经典布尔代数相似的特点。

Theorem 20 一般而言，γ 不符合分配律。

$$a \wedge (b \vee c) \neq (a \wedge b) \vee (a \wedge c)$$

当 $\min(|a|,|b|,|c|) \to \infty$ 时，等号成立

$$a \wedge (b \vee c) = (a \wedge b) \vee (a \wedge c)$$

Theorem 21 当所有变量的绝对值都趋向无穷时，γ 在 *sigmoid* 函数映射下与布尔代数等价。

Definition 22 我们定义两个实数逻辑矩阵 A, B 的乘积为

$$A \times B = \begin{pmatrix} \bigcup_q a_{1i} \wedge b_{i1} & \bigcup_q a_{1i} \wedge b_{i2} & \cdots & \bigcup_q a_{1i} \wedge b_{in} \\ \bigcup_q a_{2i} \wedge b_{i1} & \bigcup_q a_{2i} \wedge b_{i2} & \cdots & \bigcup_q a_{2i} \wedge b_{in} \\ \vdots & \vdots & \ddots & \vdots \\ \bigcup_q a_{mi} \wedge b_{i1} & \bigcup_q a_{mi} \wedge b_{i2} & \cdots & \bigcup_q a_{mi} \wedge b_{in} \end{pmatrix} \tag{12}$$

$$= \begin{pmatrix} \log(\sum_q \frac{1}{e^{-a_{1i}}+e^{-b_{i1}}}) & \log(\sum_q \frac{1}{e^{-a_{1i}}+e^{-b_{i2}}}) & \cdots & \log(\sum_q \frac{1}{e^{-a_{1i}}+e^{-b_{in}}}) \\ \log(\sum_q \frac{1}{e^{-a_{2i}}+e^{-b_{i1}}}) & \log(\sum_q \frac{1}{e^{-a_{2i}}+e^{-b_{i2}}}) & \cdots & \log(\sum_q \frac{1}{e^{-a_{2i}}+e^{-b_{in}}}) \\ \vdots & \vdots & \ddots & \vdots \\ \log(\sum_q \frac{1}{e^{-a_{mi}}+e^{-b_{i1}}}) & \log(\sum_q \frac{1}{e^{-a_{mi}}+e^{-b_{i2}}}) & \cdots & \log(\sum_q \frac{1}{e^{-a_{mi}}+e^{-b_{in}}}) \end{pmatrix} \tag{13}$$

Theorem 23 实数逻辑矩阵乘法一般不符合结合律，当矩阵元素的最小绝对值趋向于无穷时，结合律成立。

Theorem 24 $\forall \epsilon \in \mathbb{R}$，如果 $\exists M < +\infty, |\epsilon| \leq M$，那么

$$\lim_{|a|\to\infty} \frac{1}{1+e^{-(a+\epsilon)}} = \lim_{|a|\to\infty} \frac{1}{1+e^{-a}}$$

上面式子表明，当参数收敛到正负无穷时，任何有界噪声都无法对预测结果造成任何影响，模型参数会在无穷处趋于稳定。

7 优化算法

我们来将前面讨论的多个概念串联起来。首先，我们提出一个问题，是否可以将序列建模问题看作是一个半群学习的过程，也就是从数据中构建出我们的半群来。然后，我们找到了一个实现半群的好的载体，也就是布尔矩阵。我们可以让每一个作用集的元素映射到一个具体的布尔方阵。然而，布尔矩阵始终是由离散数值构成的，它对于许多基于连续实数的优化算法并不友好。后来，我们又讨论了实数逻辑矩阵，并且知道了当实数逻辑矩阵在极限情况下是会与布尔矩阵是等价的。

现在我们着手要解决的是，如何使用优化算法来使得实数逻辑矩阵收敛到我们想要的布尔矩阵。我们定义的实数逻辑矩阵拥有许多布尔矩阵的性质，也保持了较好的连续性，可以使用随机有界噪声 + 梯度下降的方式进行优化。训练时，我们会为每一个合法符号配置一个可学习的实数逻辑方阵。可以预见的是，当模型收敛到一个稳定状态时，每一个参数都会趋向于正负无穷。

在实现表现函数 ϕ 时，我们也使用布尔矩阵来实现，并且会为它设置对应实数逻辑矩阵来进行学习。如果元素 t 是作用域 T 内的一个布尔矩阵，那么我们将 ϕ 定义为

$$\phi(t) = L \times t \times R$$

其中，L 和 R 是关于 ϕ 的要被学习的布尔矩阵。它们的作用是用了改变矩阵的大小。