

最后的退路：统计建模方法研讨

黎韬

中国人民大学

版本：0.0.1

更新：2022 年 3 月 22 日



感谢 ElegantL^AT_EX 提供的数学排版支持!

本书献给我的父亲和母亲。感谢父亲从小对我的严格要求，他让我明白了“劳动创造价值，知识就是力量”；感谢母亲自小对我无私的关爱，她让我明白了“无论外面风浪再大，家永远是我温暖的港湾”。

1 统计建模方法介绍

There are two cultures in the use of statistical modeling to reach conclusions from data. One assumes that the data are generated by a given stochastic data model. The other uses algorithmic models and treats the data mechanism as unknown(Breiman, 2001).

Leo Breiman

南京大学的周志华教授将机器学习定义为“致力于研究如何通过计算的手段，利用经验来改善系统自身的性能……如果说计算机科学是研究关于算法的学问，那么类似的，机器学习是研究关于学习算法的学问(周志华, 2016)。” Mitchell et al. (1997) 给了一个更形式化的定义：“假设用 P 来评估计算机程序在某任务类 T 上的性能，若一个程序通过利用经验 E 在 T 中任务上获得了性能 P 上改善，则我们就说关于 T 和 P ，该程序对 E 进行了学习。”

根据这个定义，机器学习的范围很宽。她不仅包括了随机数据生成模型(比如线性回归、逻辑回归、Cox 模型等)，还包括了最近热门的算法模型(比如支持向量机、决策树、神经网络等)。Breiman (2001) 认为，随机数据生成模型(下称“数据模型”)和算法模型在建模思路是有区别的。数据模型会先假定数据生成过程，即目标变量是关于特征、噪声和误差的一个函数($Y = f(X, \epsilon, \beta)$, $f(\cdot)$ 形式给定)，比

如：

$$Y = X\beta + \epsilon, \epsilon \sim N(0, \sigma^2) \quad (1.1)$$

然后通过合适的参数估计方法去得到参数估计值及其标准误差。而算法建模不去假定数据生成过程，它直接认为目标变量是特征的函数：

$$Y = f(X), f(\cdot) \text{形式不给定} \quad (1.2)$$

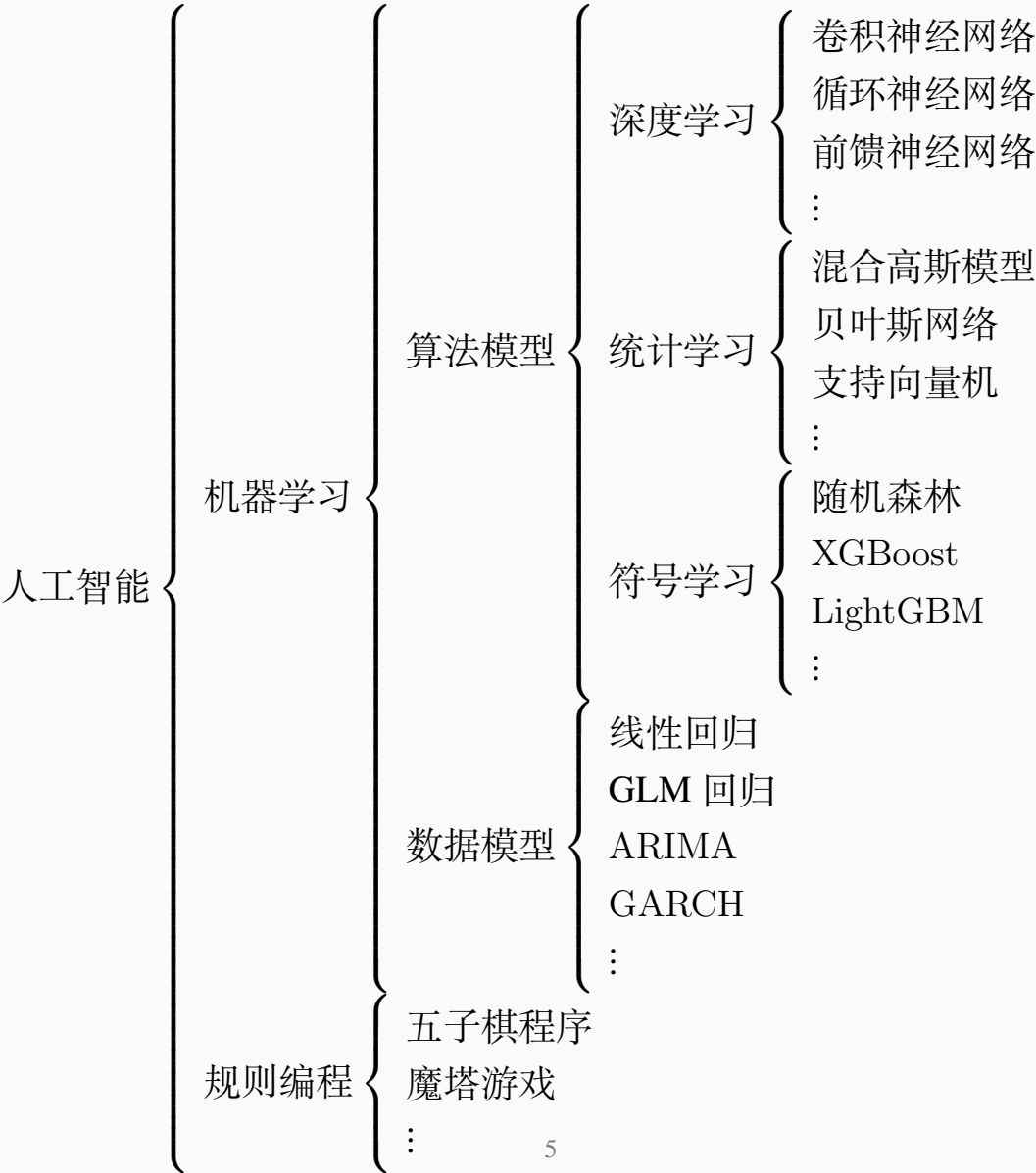
我们可以通过一定的规则（比如决策树、神经网络或支持向量机）去学习得到 $f(\cdot)$ 。

1.1 人工智能、机器学习和深度学习

人工智能是指由人制造出来的机器所表现出来的智能。通常人工智能是指通过普通计算机程序来呈现人类智能的技术 ([Wikipedia, 2021a](#))。深度学习是机器学习的分支，是一种以人工神经网络为架构，对数据进行表征学习的算法 ([Wikipedia, 2021b](#))。

机器学习是介于人工智能和深度学习之间的概念。机器学习是一种自动化学习、不需要人工显式编程的人工智能。相对于那些需要显式编程的人工智能（比如五子棋中指定不能让对手五子连结），机器学习只要指定学习方法就可以让机器不断迭代进化，自动学习出规则。深度学习是机器学习的分支，它特指使用神经网络的方法（如前馈神经网络 (NN)、卷积神经网络 (CNN)、循环神经网络 (RNN)），使

用海量的数据，针对特定的任务，为提高某项性能而学习。三者的关系总结如下：



2 深度学习

2.1 循环神经网络

虽然前馈神经网络已经足够强大，但面临某些特定问题 (如图像识别与语音识别) 时，她也同样面临着参数爆炸的问题。为了解决这个问题，研究者们发明了循环神经网络 ([Rumelhart et al., 1986](#))。正如卷积神经网络擅长处理超大型图片一样，循环神经网络擅长处理超长序列。

循环神经网络 (以下简称"RNN") 将权值共享的方法引入神经网络的建模当中。这种做法不仅大大降低了需要优化的参数量，使得参数校准更加精确，还赋予了神经网络接收不同长度的序列来训练的能力。以上两点优势是浅层的 (增大样本量就能解决)。更重要的是，因为权值共享，同一个词语对输出结果的影响，与其在句子中出现的位置无关。需要特别指出的是。RNN 并不能解决极性转移¹问题。要想解决这个问题需要借助其它的辅助模型。

2.1.1 前向传导

假设一个样本为一个长句子，所有长句子最大长度为 τ (即最多有 τ 个单词数)，每个单词都使用维度为 d 的词向量去表示，那么所有样本的形状为 (None, d, τ) ²。如果使用前馈神经网络去处理这个问题，假设有 N 个神经元，因为其只能接受一维的样本，所以需要将样本的形状展开，变成 $(\text{None}, d \times \tau)$ 。

¹句子中的一些否定词、程度副词等的使用都可能会使得句子的极性发生偏转

²None 表示样本的数量不定

那么一个前馈神经网络的参数就为 $d \times \tau \times N + N$ 个³，前馈神经网络试图学习的函数 $g(\cdot)$ 可以表示为如下形式：

$$\mathbf{h} = g(\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(\tau)}) \quad (2.1)$$

不同于前馈神经元，RNN 神经元可以直接接受形状为 (None, d, τ) 的样本。它的方法是在每一步 $t = 1, 2, \dots, \tau$ 分别处理一个输入 \mathbf{x}_t 和一个状态 \mathbf{h}_{t-1}

$$\mathbf{h}^{(t)} = g(\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(\tau)}) \quad (2.2)$$

$$= f(\mathbf{h}^{(t-1)}, \mathbf{x}^{(t)}; \boldsymbol{\theta}) \quad (2.3)$$

通过这种变换，我们可以将学习 $g(\cdot)$ 的任务简化为学习 $f(\cdot)$ ，而 $f(\cdot)$ 中的参数 $\boldsymbol{\theta}$ 因为权值共享，其参数量已经大大减少了。

给定一个 $\mathbf{h}^{(0)}$ ，我们就可以计算出任意 $\mathbf{h}^{(t)}$ ，然后将 $\mathbf{h}^{(t)}$ 传入下一个输出层（一般是全连接层）中，就可以给出预测值。我们假设输出层中的神经元有 M 个。

$$\mathbf{a}^{(t)} = \mathbf{b} + \mathbf{W}\mathbf{h}^{(t-1)} + \mathbf{U}\mathbf{x}^{(t)} \quad (2.4)$$

$$\mathbf{h}^{(t)} = \tanh(\mathbf{a}^{(t)}) \quad (2.5)$$

$$\mathbf{o}^{(t)} = \mathbf{c} + \mathbf{V}\mathbf{h}^{(t)} \quad (2.6)$$

$$\hat{\mathbf{y}}^{(t)} = \text{softmax}(\mathbf{o}^{(t)}) \quad (2.7)$$

³其中有 N 个是偏置项

损失函数一般设置为交叉熵损失函数：

$$L\left(\left\{\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(\tau)}\right\}, \left\{\mathbf{y}^{(1)}, \dots, \mathbf{y}^{(\tau)}\right\}\right) = \sum_t L^{(t)} \quad (2.8)$$

$$= - \sum_t \log p_{\text{model}}\left(\mathbf{y}^{(t)} \mid \left\{\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(t)}\right\}\right) \quad (2.9)$$

$$= - \sum_{t=1}^{\tau} \sum_{i=1}^M y_i^{(t)} \log(\hat{y}_i^{(t)}) \quad (2.10)$$

2.1.2 反向传播

RNN 中需要校准的参数有 $\mathbf{b}, \mathbf{W}, \mathbf{U}, \mathbf{c}, \mathbf{V}$ ：

$$\nabla_{\mathbf{c}} L = \sum_t (\nabla_{\mathbf{o}^{(t)}} L) \frac{\partial \mathbf{o}^{(t)}}{\partial \mathbf{c}} \quad (2.11)$$

$$(\text{vec}(\nabla_{\mathbf{V}} L))^T = \sum_t (\nabla_{\mathbf{o}^{(t)}} L) \frac{\partial \mathbf{o}^{(t)}}{\partial \text{vec}(\mathbf{V})} \quad (2.12)$$

$$\nabla_{\mathbf{b}} L = \sum_t (\nabla_{\mathbf{h}^{(t)}} L) \frac{\partial \mathbf{h}^{(t)}}{\partial \mathbf{b}} \quad (2.13)$$

$$(\text{vec}(\nabla_{\mathbf{W}} L))^T = \sum_t (\nabla_{\mathbf{h}^{(t)}} L) \frac{\partial \mathbf{h}^{(t)}}{\partial \text{vec}(\mathbf{W})} \quad (2.14)$$

$$(\text{vec}(\nabla_{\mathbf{U}} L))^T = \sum_t (\nabla_{\mathbf{h}^{(t)}} L) \frac{\partial \mathbf{h}^{(t)}}{\partial \text{vec}(\mathbf{U})} \quad (2.15)$$

其中

$$(\nabla_{\mathbf{o}^{(t)}} L) = \frac{\partial L}{\partial L^{(t)}} \frac{\partial L^{(t)}}{\partial \mathbf{o}^{(t)}} = (\hat{y}^{(t)} - I_{\{y^{(t)}=1\}})^T \quad (2.16)$$

注意 $\nabla_{\mathbf{o}^{(t)}} L$ (以及之后的所有梯度) 都是行向量。推导出上述式子需要借助于:

1. $\partial L / \partial L^{(t)} = 1$
2. $L^{(t)} = \sum_i y_i^{(t)} \log(\hat{y}_i^{(t)}) = \log(\hat{y}_{i^*}^{(t)})$, $y_i^{(t)} = 1$ 如果 $i = i^*$
3. $y_{i^*}^{(t)} = \exp[o_{i^*}^{(t)}] / \sum_j \exp(o_j^{(t)})$
4. 矩阵求导使用分子布局的方式 (之后都使用该方式)

因为第 $\mathbf{h}^{(\tau)}$ 只会决定 $\mathbf{o}^{(\tau)}$, 而不会决定 $\mathbf{h}^{(\tau+1)}$, 所以:

$$\nabla_{\mathbf{h}^{(\tau)}} L = (\nabla_{\mathbf{o}^{(\tau)}} L) \frac{\partial \mathbf{o}^{(\tau)}}{\partial \mathbf{h}^{(\tau)}} = (\nabla_{\mathbf{o}^{(\tau)}} L) V \quad (2.17)$$

对于 $t = 1, 2, \dots, (\tau - 1)$, 我们知道 $\mathbf{h}^{(t)}$ 不仅可以决定 $\mathbf{o}^{(t)}$, 还可以决定 $\mathbf{h}^{(t+1)}$, 所以:

$$\nabla_{\mathbf{h}^{(t)}} L = (\nabla_{\mathbf{h}^{(t+1)}} L) \frac{\partial \mathbf{h}^{(t+1)}}{\partial \mathbf{a}^{(t+1)}} \frac{\partial \mathbf{a}^{(t+1)}}{\partial \mathbf{h}^{(t)}} + (\nabla_{\mathbf{o}^{(t)}} L) \frac{\partial \mathbf{o}^{(t)}}{\partial \mathbf{h}^{(t)}} = (\nabla_{\mathbf{h}^{(t+1)}} L) \text{diag}[1 - (\mathbf{h}^{(t+1)})^2] W + (\nabla_{\mathbf{o}^{(t)}} L) V \quad (2.18)$$

$$\frac{\partial \mathbf{o}^{(t)}}{\partial \mathbf{c}} = I_M \quad (2.19)$$

$$\frac{\partial \mathbf{o}^{(t)}}{\partial \text{vec}(\mathbf{V})} = (\mathbf{h}^{(t)})^T \otimes I_M \quad (2.20)$$

$$\frac{\partial \mathbf{h}^{(t)}}{\partial \mathbf{b}} = \frac{\partial \mathbf{h}^{(t)}}{\partial \mathbf{a}^{(t)}} \frac{\partial \mathbf{a}^{(t)}}{\partial \mathbf{b}} = \text{diag}[1 - (\mathbf{h}^{(t)})^2] I_N \quad (2.21)$$

$$\frac{\partial \mathbf{h}^{(t)}}{\partial \text{vec}(\mathbf{W})} = \text{diag}[1 - (\mathbf{h}^{(t)})^2](\mathbf{h}^{(t-1)})^T \otimes I_N \quad (2.22)$$

$$\frac{\partial \mathbf{h}^{(t)}}{\partial \text{vec}(\mathbf{U})} = \text{diag}[1 - (\mathbf{h}^{(t)})^2](\mathbf{x}^{(t)})^T \otimes I_N \quad (2.23)$$

参考文献

- WIKIPEDIA, 2021a. 人工智能 — Wikipedia, The Free Encyclopedia[EB/OL]. <https://zh.wikipedia.org/wiki/%E4%BA%BA%E5%B7%A5%E6%99%BA%E8%83%BD>.
- WIKIPEDIA, 2021b. 深度学习 — Wikipedia, The Free Encyclopedia[EB/OL]. <https://zh.wikipedia.org/wiki/%E6%B7%B1%E5%BA%A6%E5%AD%A6%E4%B9%A0>.
- 周志华, 2016. 机器学习 [M]. 北京: 清华大学出版社: 1.
- BREIMAN L, 2001. Statistical modeling: The two cultures (with comments and a rejoinder by the author)[J]. Statistical science, 16(3): 199-231.
- MITCHELL T M, et al., 1997. Machine learning[J].
- RUMELHART D E, HINTON G E, WILLIAMS R J, 1986. Learning representations by back-propagating errors[J]. nature, 323(6088): 533-536.