Robbins-Monro (RM) 算法学习笔记

szq (根据 PPT 内容整理)

2025 年 7 月 31 日

目录

1 问题陈述：寻找函数根

假设我们的目标是找到以下方程的根：

$$g(w) = 0,$$

其中 $w \in \mathbb{R}$ 是待求解的变量，而 $g : \mathbb{R} \to \mathbb{R}$ 是一个函数。

- 许多问题最终都可以转化为这种寻根问题。例如，假设 $J(w)$ 是一个需要最小化的目标函数，那么这个优化问题可以转化为求解：

$$g(w) = \nabla_w J(w) = 0$$

- 另外，像 $g(w) = c$（其中 c 是一个常数）这样的方程也可以通过将其重写为一个新函数 $g(w) - c = 0$ 来转化为上述标准形式。

2 Robbins-Monro (RM) 算法

2.1 算法提出的背景

如何求解 $g(w) = 0$ 的根？

- 如果函数 g 或其导数的**表达式已知**，那么有许多成熟的数值算法可以解决这个问题。
- **核心问题：**如果函数 g 的**表达式未知**，该怎么办？例如，函数 g 可能由一个我们无法获得其精确表达式的人工神经网络表示。

在这种场景下，我们无法直接计算 $g(w)$ 的值，只能通过某种方式（例如，与环境交互）获得其**带有噪声的观测值**。

2.2 算法描述

Robbins-Monro (RM) 算法正是为解决这类问题而设计的。其迭代更新公式如下：

$$w_{k+1} = w_k - a_k \tilde{g}(w_k, \eta_k), \quad k = 1, 2, 3, \ldots$$

其中：

- w_k 是对根的第 k 次估计。
- $\tilde{g}(w_k, \eta_k) = g(w_k) + \eta_k$ 是对 $g(w_k)$ 的第 k 次**带噪观测**，η_k 是随机噪声。
- a_k 是一个正系数，通常称为步长或学习率。

函数 $g(w)$ 在这里是一个**黑箱 (black box)**！该算法依赖于数据：

- **输入序列：** $\{w_k\}$
- **带噪输出序列：** $\{\tilde{g}(w_k, \eta_k)\}$

核心思想： 当模型（即函数的精确表达式）未知时，我们必须依赖**数据**。

2.3 收敛性分析

为什么 RM 算法能够找到 $g(w) = 0$ 的根？通常的解释分为两步：

1. 首先通过一个直观的例子来展示其工作原理。
2. 其次给出严格的收敛性分析。

下面的定理给出了 RM 算法收敛的严格数学证明。

定理 1: Robbins-Monro Theorem

在 Robbins-Monro 算法中，如果满足以下条件：

1) 函数 $g(w)$ 的梯度（斜率）被正数界定：

$$0 < c_1 \leq \nabla_w g(w) \leq c_2 \quad \text{for all } w;$$

2) 步长序列 a_k 满足：

$$\sum_{k=1}^{\infty} a_k = \infty \quad \text{并且} \quad \sum_{k=1}^{\infty} a_k^2 < \infty;$$

3) 噪声 η_k 在给定历史信息 $\mathcal{H}_k$ 的条件下，条件期望为零，且条件二阶矩有界：

$$\mathbb{E}[\eta_k | \mathcal{H}_k] = 0 \quad \text{并且} \quad \mathbb{E}[\eta_k^2 | \mathcal{H}_k] < \infty;$$

其中 $\mathcal{H}_k = \{w_k, w_{k-1}, \dots\}$ 代表到第 k 步为止的历史信息。

那么，序列 w_k 将以概率 1 (with probability 1, w.p.1) 收敛到方程的根 w^*，满足 $g(w^*) = 0$。

3 收敛条件详解

下面我们来详细解释 Robbins-Monro 定理中的三个核心条件。

3.1 条件一：关于函数 $g(w)$

$$0 < c_1 \leq \nabla_w g(w) \leq c_2 \quad \text{for all } w$$

这个条件表明：

- 函数 $g(w)$ 是**单调递增**的，这确保了方程 $g(w) = 0$ 的根存在且**唯**一。
- 函数的梯度（斜率）有一个上界，这可以防止函数值变化过快，有助于算法的稳定。

3.2 条件二：关于步长序列 a_k

$$\sum_{k=1}^{\infty} a_k^2 < \infty \quad \text{且} \quad \sum_{k=1}^{\infty} a_k = \infty$$

这个对步长的双重条件是收敛的关键，可以分为两部分来理解：

第一部分：$\sum_{k=1}^{\infty} a_k^2 < \infty$（保证收敛到稳定点） 这个条件（级数收敛）保证了步长 a_k 最终会趋向于零，即 $\lim_{k\to\infty} a_k = 0$。

- **重要性：**考虑更新公式 $w_{k+1} - w_k = -a_k \tilde{g}(w_k, \eta_k)$。为了让序列 w_k 最终收敛，其更新步长 $w_{k+1} - w_k$ 必须趋于 0。当 w_k 接近根 w^* 时，$g(w_k)$ 会趋于 0，但噪声项 η_k 不会。因此，必须通过让 $a_k \to 0$ 来抑制噪声的持续影响，确保更新量最终消失，使得算法稳定在根附近。

第二部分：$\sum_{k=1}^{\infty} a_k = \infty$（保证能到达根） 这个条件（级数发散）保证了步长 a_k **收敛到零的速度不能太快**。

- **重要性：**将更新规则从 $k = 1$ 开始累加，形式上可以理解为：

  $$w_\infty - w_1 = \sum_{k=1}^{\infty} (w_{k+1} - w_k) = -\sum_{k=1}^{\infty} a_k \tilde{g}(w_k, \eta_k)$$

  如果步长序列的和 $\sum a_k$ 是一个有限值，那么总的“探索距离”也将是有限的。这意味着，如果初始猜测值 w_1 离真正的根 w^* 非常远，算法可能永远无法到达它。而“和发散”的条件确保了算法有能力跨越任意有限的距离，不会因为步长衰减过快而“半途而废”。

3.3 条件三：关于噪声 η_k

$$\mathbb{E}[\eta_k|\mathcal{H}_k] = 0 \quad \text{并且} \quad \mathbb{E}[\eta_k^2|\mathcal{H}_k] < \infty$$

这个条件要求噪声在给定历史信息的条件下，其**条件期望为零**，且条件二阶矩有界（即方差有界）。

- **条件期望为零**意味着，平均而言，噪声不会系统性地将估计值推向某个错误的方向，它是无偏的。
- **方差有界**确保了噪声不会出现极端离群值，从而破坏收敛进程。
- 一个常见但更强的特例是，噪声序列 $\{\eta_k\}$ 是一个独立同分布 (i.i.d.) 的随机序列，满足 $\mathbb{E}[\eta_k] = 0$ 和 $\mathbb{E}[\eta_k^2] < \infty$。
- 需要注意的是，该定理并不要求噪声服从高斯分布 (Gaussian)。

4 随机梯度下降 (Stochastic Gradient Descent)

接下来，我们介绍随机梯度下降（SGD）算法。

- SGD 算法在机器学习和强化学习领域被广泛使用。
- SGD 是一种特殊的 RM 算法。
- （之前讨论的）均值估计算法也是一种特殊的 SGD 算法。

假设我们的目标是解决以下优化问题：

$$\min_w J(w) = \mathbb{E}_X[f(w, X)]$$

这是一个目标函数包含对随机变量求期望的优化问题。

- w 是需要优化的参数。
- X 是一个随机变量，期望是针对 X 的分布计算的。
- w 和 X 既可以是标量也可以是向量，而函数 $f(\cdot)$ 的输出是标量。

为了解决这个优化问题，有几种不同的梯度下降方法。

4.1 方法一：梯度下降 (Gradient Descent, GD)

标准的梯度下降法使用目标函数对参数 w 的**真实梯度**进行更新。

$$w_{k+1} = w_k - \alpha_k \nabla_w \mathbb{E}_X[f(w_k, X)] = w_k - \alpha_k \mathbb{E}_X[\nabla_w f(w_k, X)]$$

缺点 (Drawback): 真实梯度的期望值 $\mathbb{E}_X[\nabla_w f(w_k, X)]$ 通常难以直接计算，因为它可能需要对 X 的整个概率分布进行积分。

4.2 方法二：批量梯度下降 (Batch Gradient Descent, BGD)

批量梯度下降通过在一批样本上计算梯度的平均值来**近似**真实的期望梯度。

$$\mathbb{E}_X[\nabla_w f(w_k, X)] \approx \frac{1}{n}\sum_{i=1}^{n} \nabla_w f(w_k, x_i)$$

其更新规则为：

$$w_{k+1} = w_k - \alpha_k \frac{1}{n}\sum_{i=1}^{n} \nabla_w f(w_k, x_i)$$

缺点 (Drawback): 在每一次迭代更新 w_k 时，都需要采集并计算大量的样本（一个 batch），计算成本可能很高。

4.3 方法三：随机梯度下降 (Stochastic Gradient Descent, SGD)

随机梯度下降是 GD 和 BGD 的一种简化，它在每次迭代时仅使用**单个**随机样本来估计梯度。

$$w_{k+1} = w_k - \alpha_k \nabla_w f(w_k, x_k)$$

其中 x_k 是在第 k 步从 X 的分布中随机采样的一个样本。

- **与梯度下降 (GD) 的比较：** SGD 用**随机梯度** (stochastic gradient) $\nabla_w f(w_k, x_k)$ 替代了**真实梯度** (true gradient) $\mathbb{E}_X[\nabla_w f(w_k, X)]$。这个随机梯度是真实梯度的一个**无偏估计**。
- **与批量梯度下降 (BGD) 的比较：** SGD 相当于将批量大小设置为 $n = 1$ 的 BGD，极大地降低了单次迭代的计算复杂度。