Subject: Westlake University, Reinforce Learning, Lecture 6, Stochastic Approximation and Stochastic Gradient Decent

Date: from January 20, 2025 to January 24, 2025

Contents

Lecture 6, Stochastic Approximation and Stochastic Gradient Decent

Bilibili:Lecture 6, Stochastic Approximation and Stochastic Gradient Decent

Outline

本节并为介绍新的强化学习算法，而是为介绍下一节中的 TD Learning 算法做铺垫。为解决之前介绍的算法中计算效率低的问题，介绍了随机近似(stochastic approximation)中的 RM 算法，并分析了其收敛性．接着又介绍了机器学习中的常用算法：随机梯度下降算法，分析了其收敛的模式并依据 RM 定理对其进行收敛性分析.

Motivating example: mean estimation

Example – mean estimation

我们常常通过采样求平均的方式估计均值(期望)，例如采样了 N 个样本 $\{x_i\}_{i=1}^N$:

$$\mathbb{E}[X] \approx \bar{x} := \frac{1}{N}\sum_{i=1}^{N} x_i \tag{1}$$

根据大数定律(law of large numbers)，我们可以知道 $\bar{x} \xrightarrow{N\to\infty} \mathbb{E}[X]$.

可以看到式(1)的计算需要等到 N 个样本采样完成后才能进行，即非增量式计算(**non-incremental**)，这会造成算法效率的降低. 一种替代方式是立即使用新的采样数据，即增量式(**incremental**)地迭代更新，这样虽然在算法初期损失了部分精度，但提升了算法效率.

规定

$$w_{k+1} = \frac{1}{k}\sum_{i=1}^{k} x_i, \quad k = 1, 2, \dots \tag{2a}$$

$$w_k = \frac{1}{k-1}\sum_{i=1}^{k-1} x_i, \quad k = 2, 3, \dots \tag{2b}$$

那么就有如下 incremental 格式:

$$w_{k+1} = \frac{1}{k}\sum_{i=1}^{k} x_i = \frac{1}{k}\left(\sum_{i=1}^{k-1} x_i + x_k\right) = \frac{1}{k}\left((k-1)w_k + x_k\right) = w_k - \frac{1}{k}(w_k - x_k) \tag{3}$$

更一般地，可以将上述 incremental 格式总结为

$$w_{k+1} = w_k - \alpha_k(w_k - x_k) \tag{4}$$

此时 $\alpha_k > 0$，后续将说明当其满足一些条件时，仍然会有 $x_k \xrightarrow{k\to\infty} \mathbb{E}[X]$.

Note 1. 式*(2a)(2b)*只是为方便格式*(4)*中左右分别只有 $k+1$ 和 k. 当然也可将上述几个式子写为:

$$w_k = \frac{1}{k}\sum_{i=1}^{k} x_i, \quad w_{k+1} = \frac{1}{k+1}\sum_{i=1}^{k+1} x_i, \quad k = 1, 2, \dots \tag{5a}$$

$$w_{k+1} = w_k - \frac{1}{k+1}(w_k - x_{k+1}) = w_k - \alpha_k(w_k - x_{k+1}) \tag{5b}$$

Robbins-Monro algorithm(RM Alg.)

Robbins-Monro algorithm

随机近似(Stochastic approximation, SA) 是一大类算法，可用于方程的求根问题和优化问题。相较于例如牛顿法、梯度下降等算法，SA 可以在不知道函数表达式（但可以获得有噪/无噪输出）的情况下使用迭代更新的方式进行求解。其中，Robbins-Monro(RM) 算法是 SA 中具有开创性的工作，著名的随机梯度下降算法和 mean estimation 都是 RM 算法的特殊情况.

Problem statement: 考虑一个方程的求零点问题：

$$g(w) = 0$$

实际上优化问题和方程的求根问题都是这样的形式：

$$\begin{aligned} g(w) &= \nabla_w J(w) = 0 \\ g(w) &= c \rightarrow \tilde{g} = g(w) - c = 0 \end{aligned}$$

其中 $g(w)$ 为表达式未知的黑盒(black box)（例如神经网络就可以抽象为 $g(w) = y$ 的形式，但是 g 的形式是未知的），我们只能获得有噪输出

$$\tilde{g}(w, \eta) = g(w) + \eta$$

其中 η 为观测噪声.

Algorithm: RM 算法通过如下迭代方式逼近真实解：

$$w_{k+1} = w_k - a_k \tilde{g}(w_k, \eta_k), \quad k = 1, 2, 3, \ldots \tag{6}$$

其中 w_k是对解的第 k 次估计. RM 算法只依赖于数据：

1. Input sequence: $\{w_k\}$
2. Noisy output sequence: $\{\tilde{g}(w_k, \eta_k)\}$

一个简单的数值案例如下图1所示，图中求解了 $f(x) = x^3 - 2 = 0$ 的根，初始猜测 $x_0 = 0$，代码详见附录B.1

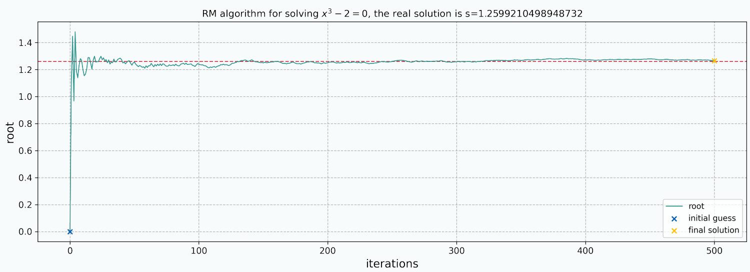

Figure 1: Experiment for RM alpgrithm

Robbins-Monro algorithm – Convergence properties

Theorem 1 (Robbins-Monro theorem)**.** 对于 *Robbins-Monro* 算法，若

1. $\forall w, 0 < c_1 \leqslant \nabla_w g(w) \leqslant c_2$（梯度的要求）
2. $\sum_{k=1}^{\infty} a_k = \infty, \sum_{k=1}^{\infty} a_k^2 < \infty$（系数的要求）
3. $\mathbb{E}[\eta_k|\mathcal{H}_k] = 0, \mathbb{E}[\eta_k^2|\mathcal{H}_k] < \infty$（测量误差的要求）

其中 $\mathcal{H}_k = \{w_k, w_{k-1}, \ldots\}$. 设方程 $g(w) = 0$ 的根为 w^*，满足 $g(w^*) = 0$，那么 w_k 以概率*1* 收敛*(with probability 1, w.p.1)*至 w^*. 证明需要使用 *Dvoretzky's theorem*，详见附录*A.2.*

Note 2. 对于上述三个条件，有如下详细解读：

1. 条件 *1* – 梯度的要求
   (a) 梯度 $\nabla_w g(w) > 0$，即原函数单调上升，保证了方程 $g(w) = 0$ 一定有唯一解
   (b) 梯度 $\nabla_w g(w) > 0$ 这一条件一般是可以接受的，例如当处理优化问题时，$g(w)$ 本身即梯度，$g(w)$ 梯度大于 *0* 表示原函数是凸的，这是较常见的凸优化问题
   (c) $\nabla_w g(w) \leqslant c_2$ 说明梯度有界，不过不满足此条件算法有时也能够收敛
2. 条件 2 – 系数的要求
   (a) $\sum_{k=1}^{\infty} a_k^2 < \infty$ 表明 $a_k \xrightarrow{k\to\infty} 0$. 这是由于 $w_{k+1} - w_k = -a_k \tilde{g}(w_k, \eta_k)$，因此为保证算法收敛需要 $a_k \xrightarrow{k\to\infty} 0$
   (b) $\sum_{k=1}^{\infty} a_k = \infty$ 表明 $a_k \xrightarrow{k\to\infty} 0$ 没有那么快，这是由于根据递推式 $w_2 = w_1 - a_1\tilde{g}(w_1, \eta_1), \cdots, w_{k+1} = w_k - a_k\tilde{g}(w_k, \eta_k), \cdots$，将两端累和消去可以得到
   $$w_1 - w_\infty = \sum_{k=1}^{\infty} a_k \tilde{g}(w_k, \eta_k)$$
   因此若 $\sum_{k=1}^{\infty} a_k < \infty$，由于 $\tilde{g}$ 有界，那么$|\sum_{k=1}^{\infty} a_k \tilde{g}(w_k, \eta_k)|$ 有界，这就要求初始猜测 w_1 不能距 w^* 过远，反之若收敛至零较慢则可放宽对初始猜测的要求
   (c) 取 $a_k = \frac{1}{k}$是可以满足要求的，因为有 $\sum\limits_{k=1}^{n} \frac{1}{k} \to \infty, \sum\limits_{k=1}^{\infty} \frac{1}{k} < \infty$, 原因在于
   $$\lim_{n\to\infty} \left(\sum_{k=1}^{n} \frac{1}{k} - \ln n \right) = \kappa, \ln n \xrightarrow{n\to\infty} \infty \Rightarrow \sum_{k=1}^{n} \frac{1}{k} \xrightarrow{n\to\infty} \infty$$
   $$\sum_{k=1}^{\infty} \frac{1}{k^2} = \frac{\pi^2}{6} < \infty (\textit{Basel problem})$$
   其中 $\kappa \approx 0.577$ 为 *Euler-Mascheroni* 常数[a].
   (d) 常见的做法并非令系数为 $\frac{1}{k}$，因为当 k 较大时，后面的数据在算法中起到的作用被无限缩小，这并不是我们想要的，因此常见的做法是开始时赋予较小的值再慢慢减小但不至于趋于 *0*
3. 条件 *3* – 测量误差的要求. 这里并不要求 η 满足高斯性, 常见的做法是取 $\{\eta_k\}$ 来自一个 *i.i.d.* 的随机序列，满足 $\mathbb{E}[\eta_k|\mathcal{H}_k] = \mathbb{E}[\eta_k] = 0, \mathbb{E}[\eta_k^2|\mathcal{H}_k] = \mathbb{E}[\eta_k^2] < \infty$

[a]欧拉-马斯刻若尼常数是一个数学常数，定义为调和级数与自然对数的差值

Robbins-Monro algorithm – Apply to mean estimation

定义 $g(w) \triangleq w - \mathbb{E}[X]$，我们的目的是解出 $g(w) = 0$，我们可以定义有噪观测是

$\tilde{g}(w,x) \triangleq w-x$，其中 x 是对 X 的采样，那么可以得出

$$\tilde{g}(w,\eta)=(w-\mathbb{E}[X])+(\mathbb{E}[X]-x)\triangleq g(w)+\eta$$

其中 $\eta \triangleq \mathbb{E}[X]-x$. 代入 RM 算法(6)中就得到:

$$w_{k+1}=w_k-\alpha_k\tilde{g}(w_k,\eta_k)=w_k-\alpha_k(w_k-x_k)$$

再套用 Theorem1 即可得到相应的收敛性结论，并且此收敛性对于 X 的分布没有任何假设. 其收敛性分析也可以直接使用 Dvoretzkys convergence theorem，详细证明见附录A.3.

Dvoretzky's convergence theorem(optional)

Dvoretzkys convergence theorem

Dvoretzkys convergence theorem 是随机近似领域的经典结论，该定理可用于分析 RM 算法和许多强化学习算法的收敛性.

Theorem 2 (Dvoretzkys convergence theorem)**.** 考虑如下随机过程:

$$w_{k+1}=(1-\alpha_k)w_k+\beta_k\eta_k$$

其中 $\{\alpha_k\}_{k=1}^{\infty},\{\beta_k\}_{k=1}^{\infty},\{\eta_k\}_{k=1}^{\infty}$ 均为随机序列(*stochastic sequences*)，$\forall k,\alpha_k\geqslant 0,\beta_k\geqslant 0$，那么若如下条件满足则 $w_k\xrightarrow{w.p.1}0$:

1. $\sum_{k=1}^{\infty}\alpha_k=\infty,\sum_{k=1}^{\infty}\alpha_k^2<\infty;\sum_{k=1}^{\infty}\beta_k^2<\infty$ *uniformly w.p.1*
2. $\mathbb{E}[\eta_k|\mathcal{H}_k]=0,\mathbb{E}[\eta_k^2|\mathcal{H}_k]\leqslant C$ *w.p.1*

其中 $\mathcal{H}_k=\{w_k,w_{k-1},\ldots,\eta_{k-1},\ldots,\alpha_{k-1},\ldots,\beta_{k-1},\ldots\}$. 证明见附录*A.1*

Note 3. 关于 *Dvoretzkys convergence theorem(DCT)* 有如下几点说明:

1. *DCT* 中 $\{\alpha_k\},\{\beta_k\}$ 依赖于 $\mathcal{H}_k$，但在 *RM* 算法中 $\{\alpha_k\}$ 是确定性的，不依赖于 $\mathcal{H}_k$.
2. 由于 $\mathcal{H}_k$ 是随机序列，因此 $\mathbb{E}[\eta_k|\mathcal{H}_k],\mathbb{E}[\eta_k^2|\mathcal{H}_k]$ 均为随机变量
3. *DCT* 的推广版本可用于分析后续的 *Q-learning* 和 *TD learning* 算法

Dvoretzkys convergence theorem's application to mean estimation

M ean estimation 问题可以归约为 RM alg. 从而直接使用 **RM Theorem 1** 进行收敛性分析，其也可以使用 **Dvoretzky's Theorem 2**分析，详见附录A.3.

Stochastic gradient decent

Stochastic gradient decent – Algorithm

随机梯度下降(Stochastic gradient decent, **SGD**)是机器学习中的常用优化算法，我们将说明 1. SGD 是 RM 算法的特殊情况；2. Mean estimation 是 SGD 的特殊情况.

Problem statement: 我们需要解决如下优化问题:

$$\min_{w} \quad J(w) = \mathbb{E}[f(w, X)] \tag{7}$$

其中

1. w 为优化的目标参数/变量; X 为随机变量, 对 X 取期望
2. w, X 可为标量也可以为向量, $f(\cdot)$ 为标量

Gradient Decent(GD) 梯度下降法:

$$w_{k+1} = w_k - \alpha_k \nabla_w \mathbb{E}[f(w_k, X)] = w_k - \alpha_k \mathbb{E}[\nabla_w f(w_k, X)] \tag{8}$$

其中 α_k 为步长, 控制下降的快慢.

Batch Gradient Decent(BGD) 在实际使用中由于 X 的分布常常未知, 因此采用采样的方式进行估算（蒙特卡洛的思想）:

$$\mathbb{E}[\nabla_w f(w_k, X)] \approx \frac{1}{n}\sum_{i=1}^{n} \nabla_w f(w_k, x_i) \tag{9a}$$

$$w_{k+1} = w_k - \alpha_k \frac{1}{n}\sum_{i=1}^{n} \nabla_w f(w_k, x_i) \tag{9b}$$

此算法的缺陷是在每一次迭代中需要采样很多样本, 因此计算很困难.

Stochastic Gradient Decent(SGD) 为降低 BDG 的计算量, 随机梯度下降仅使用一个随机的样本替代期望的计算, 即将 BGD 中的 $n = 1$:

$$w_{k+1} = w_k - \alpha_k \nabla_w f(w_k, x_k) \tag{10}$$

虽然 SGD 会较不精确, 但是可以提升算法的效率. 同时其也能够保证 $w_k \xrightarrow{k\to\infty} w^*$, 收敛性证明见 **Theorem 3**.

Stochastic gradient decent – Application to mean estimation

实际上 mean estimation 是特殊的 SGD 算法, 我们只需将其形式化为如下优化问题:

$$\min_{w} \quad J(w) = \mathbb{E}[f(w, X)] \triangleq \mathbb{E}\left[\frac{1}{2}\|w - X\|^2\right]$$

其中 $f(w, X) \triangleq \|w - X\|^2/2$, 其梯度 $\nabla_w f(w, X) = w - X$. 通过求解 $\nabla_w J(w) = 0$ 可以得到最优解 $w^* = \mathbb{E}[X]$. 因此此优化问题等价于 mean estimation 问题.

GD: 解决此问题的 GD 算法即

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

SGD: 由于 $\mathbb{E}[w_k - X]$ 无法计算, 因此使用样本代替期望, 得到 SGD 算法:

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

可以看到这与 mean estimation 的算法(4)一样, 因此 mean estimation 是特殊的 SGD.

Stochastic gradient decent – Convergence pattern

对比 GD 与 SGD 可以发现，变化就是梯度的计算从取期望变成了一次简单的采样：

$$\begin{aligned} w_{k+1} &= w_k - \alpha_k \mathbb{E}[\nabla_w f(w_k, X)] \\ &\Downarrow \\ w_{k+1} &= w_k - \alpha_k \nabla_w f(w_k, x_k) \end{aligned}$$

那么 $\nabla_w f(w_k, x_k)$ 就可以看作 $\mathbb{E}[\nabla_w f(w, X)]$ 的有噪估计：

$$\nabla_w f(w_k, x_k) = \mathbb{E}[\nabla_w f(w, X)] + \underbrace{\nabla_w f(w_k, x_k) - \mathbb{E}[\nabla_w f(w, X)]}_{\eta}$$

由于

$$\nabla_w f(w_k, x_k) \neq \mathbb{E}[\nabla_w f(w, X)]$$

因此我们需要考虑是否 $w_k \xrightarrow{k\to\infty} w^*$，并且考察收敛的快慢和是否带有随机性.

事实上 SGD 算法是收敛的(见 **Theorem 3**)，并且当估计值 w_k 与最优解 $w*$ 相去甚远时，其行为类似于 GD. 只有当 w_k 接近 $w*$ 时，SGD 的收敛才会表现出更多的随机性. 为简单分析，需要先定义梯度估计时带来的相对误差(relative error) δ_k:

$$\delta_k \triangleq \frac{|\nabla_w f(w_k, x_k) - \mathbb{E}[\nabla_w f(w_k, X)]|}{|\mathbb{E}[\nabla_w f(w_k, X)]|}$$

为简便起见，假设 $w, \nabla_w f(w_k, x_k)$ 均为标量，那么根据拉格朗日中值定理有：

$$\delta_k = \frac{|\nabla_w f(w_k, x_k) - \mathbb{E}[\nabla_w f(w_k, X)]|}{|\mathbb{E}[\nabla_w f(w_k, X)] - \mathbb{E}[\nabla_w f(w^*, X)]|} = \frac{|\nabla_w f(w_k, x_k) - \mathbb{E}[\nabla_w f(w_k, X)]|}{|\mathbb{E}[\nabla_w^2 f(\tilde{w}_k, X)(w_k - w^*)]|}$$

其中 $\tilde{w}_k \in][w_k, w^*]$. 由于已经假设函数 $f(\cdot)$ 是严格凸的，即 $\exists c > 0, \forall w, X, \nabla_w^2 f \geqslant c \geqslant 0$，因此有

$$\left|\mathbb{E}\left[\nabla_w^2 f(\tilde{w}_k, X)(w_k - w^*)\right]\right| = \left|\mathbb{E}\left[\nabla_w^2 f(\tilde{w}_k, X)\right]\right| |(w_k - w^*)| \geqslant c\,|w_k - w^*|$$

带入相对误差定义式中得到

$$\delta_k \leqslant \frac{|\overbrace{\nabla_w f(w_k, x_k)}^{\text{stochastic gradient}} - \overbrace{\mathbb{E}[\nabla_w f(w_k, X)]}^{\text{true gradient}}|}{\underbrace{c|w_k - w^*|}_{\text{distance to the optimal solution}}}. \tag{11}$$

从式(11)可以看出

1. 当 $w_k - w^*$ 较大，即离最优解较远时，SGD 会表现得更像 GD，梯度下降更新的更准确
2. 当 $w_k - w^*$ 较小，即离最优解较近时，确实会存在一定的随机性，但是此时已经接近了最优解

Stochastic gradient decent – Convergence

Theorem 3 (Convergence of SGD). 对于 *SGD* 算法*(10)*，定义满足 $\nabla_w \mathbb{E}[f(w,X)] = 0$ 的根为 w^*。若满足以下条件则 $w_k \xrightarrow{w.p.1} w^*$：

1. $0 < c_1 \leqslant \nabla_w^2 f(w,X) \leqslant c_2$（严格凸性）
2. $\sum_{k=1}^{\infty} a_k = \infty, \quad \sum_{k=1}^{\infty} a_k^2 < \infty$(与 *RM* 算法相同)
3. $\{x_k\}_{k=1}^{\infty}$ *i.i.d.*（常见要求）

证明的思路是说明 SGD 是一个特殊的 RM 算法(实际上直接证明也可以，但会很复杂). 首先优化问题(7)在严格凸的条件下可以转化为如下求根问题：

$$J(w) = \mathbb{E}[f(w,X)] \Rightarrow g(w) \triangleq \nabla_w J(w) = \mathbb{E}[\nabla_w f(w,X)] = 0$$

令

$$\tilde{g}(w,\eta) = \nabla_w f(w,x) = \underbrace{\mathbb{E}[\nabla_w f(w,X)]}_{g(w)} + \underbrace{\nabla_w f(w,x) - \mathbb{E}[\nabla_w f(w,X)]}_{\eta}$$

那么解决 $g(w) = 0$ 的 RM 算法为：

$$w_{k+1} = w_k - a_k \tilde{g}(w_k,\eta_k) = w_k - a_k \nabla_w f(w_k,x_k)$$

这就是 SGD，这样根据 Theorem 1就可以得到上述结论. 证明详见附录A.4.

Stochastic gradient decent – Experiment

实验视频 here

Stochastic gradient decent – A deterministic formulation

可以看到梯度下降算法的问题设置中包含了求期望，但是在深度学习常常涉及到的是最小化训练样本的误差，即：

$$\min_w \quad J(w) = \frac{1}{n}\sum_{i=1}^{n} f(w,x_i)$$

其中 $f(w,x_i)$ 是参数化的函数，w 为需优化的目标变量，$\{x_i\}_{i=1}^{n}$ 为采样得到的随机变量.

此时求解此问题的 GD 算法和 SGD 算法分别为：

$$GD: w_{k+1} = w_k - \alpha_k \nabla_w J(w_k) = w_k - \alpha_k \frac{1}{n}\sum_{i=1}^{n} \nabla_w f(w_k,x_i) \tag{12a}$$

$$SGD: w_{k+1} = w_k - \alpha_k \nabla_w f(w_k,x_k) \tag{12b}$$

如果我们将 x_i 的选取看作是从 $p_k = \frac{1}{n}$ 的均匀分布($p(X = x_i) = \frac{1}{n}$)中取样得到的，那么显然算法(12b)仍然是一种 SGD 算法；并且此时由于随机性，那么 x_k 的选取是随机的，可能反复取到，因此不需要进行排序后依次选择.

BGF, MBGD, and SGD

Example

梯度下降算法根据使用的样本量不同可以分为三种:

1. 批量梯度下降(batch gradient descent, **BGD**):每次迭代都使用所有样本

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

2. 小批量梯度下降(mini-batch gradient descent, **MBGD**):仅使用一小部分样本

$$w_{k+1} = w_k - \alpha_k \frac{1}{m} \sum_{j\in\mathcal{I}_k} \nabla_w f(w_k, x_j), \quad \text{(MBGD)}$$

   其中 $\mathcal{I}_k$ 是 k 时刻获取的 batch 下标集，在 $\{1, 2, \cdots, n\}$ 中随机抽取，$|\mathcal{I}_k| = m \leqslant n$.

3. 随机梯度下降(stochastic gradient descent, **SGD**):仅使用一个样本

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

1. 可以认为 MBGD 囊括了 BGD 和 SGD:当 batch $|\mathcal{I}_k| \to n$ 时，MBGD $\to$ BGD，当 batch $|\mathcal{I}_k| \to 1$ 时，MBGD $\to$ SGD
2. 直观上来说随机性 GD > MBGD > BGD，梯度估计准确性 BGD > MBGD > GD
3. MBGD 中 $\mathcal{I}_k$ 是一个采样集合，有可能采样到重复的，因此即使是 $|\mathcal{I}_k| = n$ BGD 与 MBGD 也有细微差别，因为 MBGD 是可能重复采样，无法覆盖整个 $\{1, 2, \cdots, n\}$.

A Proof

A.1 Proof for Dvoretzkys convergence theorem 2

Proof for Dvoretzkys convergence theorem

Proposition 1 (Dvoretzkys convergence theorem)**.** 考虑如下随机过程:

$$w_{k+1} = (1-\alpha_k)w_k + \beta_k\eta_k$$

其中 $\{\alpha_k\}_{k=1}^{\infty}, \{\beta_k\}_{k=1}^{\infty}, \{\eta_k\}_{k=1}^{\infty}$ 均为随机序列*(stochastic sequences)*，$\forall k, \alpha_k \geqslant 0, \beta_k \geqslant 0$，那么若如下条件满足则 $w_k \xrightarrow{w.p.1} 0$:

1. $\sum_{k=1}^{\infty}\alpha_k = \infty, \sum_{k=1}^{\infty}\alpha_k^2 < \infty; \sum_{k=1}^{\infty}\beta_k^2 < \infty$ *uniformly w.p.1*
2. $\mathbb{E}[\eta_k|\mathcal{H}_k] = 0, \mathbb{E}[\eta_k^2|\mathcal{H}_k] \leqslant C$ *w.p.1*

其中 $\mathcal{H}_k = \{w_k, w_{k-1}, \ldots, \eta_{k-1}, \ldots, \alpha_{k-1}, \ldots, \beta_{k-1}, \ldots\}$.

Proof. 令 $h_k \triangleq w_k^2$，那么就有

$$\begin{aligned} h_{k+1} - h_k &= w_{k+1}^2 - w_k^2 = (w_{k+1}+w_k)(w_{k+1}-w_k) \\ &= [(2-\alpha_k)w_k + \beta_k\eta_k]\,[-\alpha_k w_k + \beta_k\eta_k] = (\alpha_k^2 - 2\alpha_k)w_k^2 + 2(1-\alpha_k)\beta_k w_k\eta_k + \beta_k^2\eta_k^2 \end{aligned}$$

对等号两边同时取条件期望，由于 w_k 依赖于 $\mathcal{H}_k$，因此可以将其从期望中提出来，再考虑 α_k, β_k 仅依赖于 $\mathcal{H}_k$ 的简单情形（复杂情形暂不讨论），就有

$$\begin{aligned} \mathbb{E}[h_{k+1} - h_k|\mathcal{H}_k] &= \mathbb{E}\left[(\alpha_k^2 - 2\alpha_k)w_k^2 + 2(1-\alpha_k)\beta_k w_k\eta_k + \beta_k^2\eta_k^2\right] \\ &= \mathbb{E}\left[(\alpha_k^2 - 2\alpha_k)w_k^2|\mathcal{H}_k\right] + \mathbb{E}\left[2(1-\alpha_k)\beta_k w_k\eta_k|\mathcal{H}_k\right] + \mathbb{E}\left[\beta_k^2\eta_k^2|\mathcal{H}_k\right] \\ &= -\alpha_k(2-\alpha_k)w_k^2 + 2(1-\alpha_k)\beta_k w_k\mathbb{E}[\eta_k|\mathcal{H}_k] + \beta_k^2\mathbb{E}[\eta_k^2|\mathcal{H}_k] \end{aligned}$$

根据条件 $\sum_{k=1}^{\infty}\alpha_k^2 < \infty$ 可知 $\alpha_k \to 0$，那么 $\exists\, n \in \mathbb{N}_+$，$s.t.\ 1 < 2-\alpha < 2$，因此不妨就设 $0 \leqslant \alpha \leqslant 1$，那么 $-\alpha_k(2-\alpha_k)w_k^2 \leqslant 0$，又由于 $\mathbb{E}[\eta_k^2|\mathcal{H}_k] \leqslant C, \mathbb{E}[\eta_k|\mathcal{H}_k] = 0$，故:

$$\mathbb{E}[h_{k+1} - h_k|\mathcal{H}_k] = -\alpha_k(2-\alpha_k)w_k^2 + \beta_k^2\mathbb{E}[\eta_k^2|\mathcal{H}_k] \leqslant \beta_k^2 C \tag{13}$$

通过累和 h_k，同时考虑到 $\sum_{k=1}^{\infty}\beta_k^2 < \infty$，有:

$$\sum_{k=1}^{\infty}\mathbb{E}[h_{k+1} - h_k|\mathcal{H}_k] \leqslant \sum_{k=1}^{\infty}\beta_k^2 C < \infty$$

根据 quasimartingale convergence theorem 可知 h_k 收敛(w.p.1).

由式(13)累和可得

$$\sum_{k=1}^{\infty}\alpha_k(2-\alpha_k)w_k^2 = \sum_{k=1}^{\infty}\beta_k^2\mathbb{E}[\eta_k^2|\mathcal{H}_k] - \sum_{k=1}^{\infty}\mathbb{E}[h_{k+1} - h_k|\mathcal{H}_k] < \infty$$

由于已经假设 $0 \leqslant \alpha \leqslant 1$，因此

$$\sum_{k=1}^{\infty} \alpha_k w_k^2 \leqslant \sum_{k=1}^{\infty} \alpha_k (2-\alpha_k) w_k^2 < \infty$$

由于 $\sum_{k=1}^{\infty} \alpha_k = \infty$，因此 $w_k \to 0$ w.p.1. □

A.2 Proof for Robbins-Monro theorem 1

Proof for Robbins-Monro theorem

Proposition 2 (Robbins-Monro theorem)**.** 对于 *Robbins-Monro* 算法，若

1. $\forall w, 0 < c_1 \leqslant \nabla_w g(w) \leqslant c_2$（梯度的要求）
2. $\sum_{k=1}^{\infty} a_k = \infty, \sum_{k=1}^{\infty} a_k^2 < \infty$（系数的要求）
3. $\mathbb{E}[\eta_k|\mathcal{H}_k] = 0, \mathbb{E}[\eta_k^2|\mathcal{H}_k] < \infty$（测量误差的要求）

其中 $\mathcal{H}_k = \{w_k, w_{k-1}, \ldots\}$. 设方程 $g(w) = 0$ 的根为 w^*，满足 $g(w^*) = 0$，那么 w_k 以概率*1* 收敛*(with probability 1, w.p.1)*至 w^*.

Proof. RM 算法是为求方程 $g(w) = 0$ 的根 w^*，其算法可以写为

$$w_{k+1} = w_k - a_k \tilde{g}(w_k, \eta_k) = w_k - a_k[g(w_k) + \eta_k]$$

其可以被改写为

$$w_{k+1} - w^* = w_k - w^* - a_k[g(w_k) - g(w^*) + \eta_k].$$

根据拉格朗日中值定理可知 $\exists w_k' \in (w_k, w^*)$，$s.t.\ g(w_k) - g(w^*) = \nabla_w g(w_k')(w_k - w^*)$，令 $\Delta_k \triangleq w_k - w^*$，上式可被改写为

$$\Delta_{k+1} = \Delta_k - a_k[\nabla_w g(w_k')\Delta_k + \eta_k] = [1 - \underbrace{a_k \nabla_w g(w_k')}_{\alpha_k}]\Delta_k + \underbrace{(-a_k)}_{\beta_k}\eta_k$$

根据 RM 算法的条件可知，

$$c_1 a_k \leqslant \alpha_k \leqslant c_2 a_k$$

因此

$$\sum_{k=1}^{\infty} \alpha_k \geqslant c_1 \sum_{k=1}^{\infty} a_k \Rightarrow \sum_{k=1}^{\infty} \alpha_k = \infty (\text{满足DCT条件1第1点})$$
$$\sum_{k=1}^{\infty} \alpha_k^2 \leqslant c_2 \sum_{k=1}^{\infty} a_k^2 < \infty (\text{满足DCT条件1第2点})$$
$$\sum_{k=1}^{\infty} \beta_k^2 = \sum_{k=1}^{\infty} a_k^2 < \infty (\text{满足DCT条件1第3点})$$

DCT的条件2直接可以由RM算法条件3进行保证，因此根据 DCT 可知 $w_k \xrightarrow{w.p.1} w^*$. □

A.3 DCT's application to mean estimation

Dvoretzkys convergence theorem's application to mean estimation

Proposition 3 (Mean estimation)**.** 令 $\tilde{g}(w,\eta) = (w - \mathbb{E}[X]) + (\mathbb{E}[X] - x) \triangleq g(w) + \eta$，其中 $\eta \triangleq \mathbb{E}[X] - x$. 代入 *RM* 算法*(6)*中就得到:

$$w_{k+1} = w_k - \alpha_k \tilde{g}(w_k, \eta_k) = w_k - \alpha_k(w_k - x_k)$$

此算法在一定条件下收敛.

Proof. 令 $w^* \triangleq \mathbb{E}[X]$，那么 mean estimation 的 RM 算法可以重新写为

$$w_{k+1} - w^* = w_k - w^* + \alpha_k(x_k - w^* + w^* - w_k)$$

令 $\Delta \doteq w - w^*$，有

$$\Delta_{k+1} = \Delta_k + \alpha_k(x_k - w^* - \Delta_k) = (1-\alpha_k)\Delta_k + \alpha_k \underbrace{(x_k - w^*)}_{\eta_k}$$

下面逐条验证 DCT 的条件:

1. 只需要将参数 α_k 的设置满足 **Theorem 1** 中的条件即可满足条件1.
2. 由于 $\{x_k\}$ 是 i.i.d.，因此有 $\mathbb{E}[x_k|\mathcal{H}_k] = \mathbb{E}[x_k] = \mathbb{E}[X] = w^*$，因此显然有 $\mathbb{E}[\eta_k|\mathcal{H}_k] = 0$ w.p.1. 此外，$\mathbb{E}[\eta_k^2|\mathcal{H}_k] = \mathbb{E}\left[x_k^2 + w^{*2} - 2x_k \cdot w^*|\mathcal{H}_k\right] = \mathbb{E}\left[x_k^2\right] - w^{*2}$，因此若 $\{x_k\}$ 的方差有界则 $\mathbb{E}[\eta_k^2|\mathcal{H}_k]$ 有界，即可满足条件2.

□

A.4 Proof for Convergence of SGD theorem 3

Proof for Convergence of SGD

Proposition 4 (Convergence of SGD)**.** 对于 *SGD* 算法*(10)*，定义满足 $\nabla_w \mathbb{E}[f(w,X)] = 0$ 的根为 w^*。若满足以下条件则 $w_k \xrightarrow{w.p.1} w^*$:

1. $0 < c_1 \leqslant \nabla_w^2 f(w,X) \leqslant c_2$（严格凸性）
2. $\sum_{k=1}^{\infty} a_k = \infty, \quad \sum_{k=1}^{\infty} a_k^2 < \infty$*(*与 *RM* 算法相同*)*
3. $\{x_k\}_{k=1}^{\infty}$ *i.i.d.*（常见要求）

Proof. 首先优化问题(7)在严格凸的条件下可以转化为如下求根问题:

$$J(w) = \mathbb{E}[f(w,X)] \Rightarrow g(w) \triangleq \nabla_w J(w) = \mathbb{E}[\nabla_w f(w,X)] = 0$$

令

$$\tilde{g}(w,\eta) = \nabla_w f(w,x) = \underbrace{\mathbb{E}[\nabla_w f(w,X)]}_{g(w)} + \underbrace{\nabla_w f(w,x) - \mathbb{E}[\nabla_w f(w,X)]}_{\eta}$$

那么解决 $g(w)=0$ 的 RM 算法为:

$$w_{k+1} = w_k - a_k\tilde{g}(w_k,\eta_k) = w_k - a_k\nabla_w f(w_k,x_k)$$

这就是 SGD 算法(10)，因此 SGD 是特殊的 RM 算法. 因此下面逐条验证 **Theorem 1** 的条件能够满足:

1. 根据强凸性有

$$c_1 \leqslant \nabla_w g(w) = \nabla_w \mathbb{E}[\nabla_w f(w,X)] = \mathbb{E}[\nabla_w^2 f(w,X)] \leqslant c_2$$

因此条件1满足

2. 条件2二者相同，自然满足

3. 由于 $\eta = \nabla_w f(w,x) - \mathbb{E}[\nabla_w f(w,X)]$，故有

$$\begin{aligned}\mathbb{E}[\eta_k|\mathcal{H}_k] &= \mathbb{E}[\nabla_w f(w_k,x_k) - \mathbb{E}[\nabla_w f(w_k,X)]|\mathcal{H}_k] \\ &= \mathbb{E}_{x_k}[\nabla_w f(w_k,x_k)] - \mathbb{E}[\nabla_w f(w_k,X)] = 0\end{aligned}$$

$$\mathbb{E}[\eta_k^2|\mathcal{H}_k] = \mathbb{E}\left[(\nabla_w f(w,x))^2\right] - (\mathbb{E}[\nabla_w f(w_k,X)])^2$$

因此若 $|\nabla_w f(w,x)| < \infty$，则 $\mathbb{E}[\eta_k^2|\mathcal{H}_k] < \infty, \forall w,x$，故条件3成立.

综上所述，根据 **Theorem 1** 可知 SGD 算法收敛. □

B Codes

B.1 RM algorithm

Introduction

```python
import numpy as np
import matplotlib.pyplot as plt
import sympy

def RM(initial_guess, noisy_f, step_fun, ite):
    S = [initial_guess]
    for i in range(ite):
        s = S[i] - step_fun(i+1)*noisy_f(S[i])
        S.append(s)
    return S

real_f = lambda x: x**3 - 2
noisy_f = lambda x: x**3 - 2 + np.random.uniform(low=-1, high=1.0)
step_fun = lambda x: 1 / x

x = sympy.Symbol("x")
real_solution = float(sympy.solve(x**3 - 2)[0])

initial_guess = 0.0
ite = 500
solotions = RM(initial_guess, noisy_f, step_fun, ite)

qing = (36/255, 144/255, 135/255)
hong = (200/255, 29/255, 49/255)
huang = '#FFC200'
```

```
lan = '#005DAF'

plt.figure(figsize=(16, 5), dpi = 200)
plt.axhline(y=real_solution, color=hong, alpha = 0.9, linestyle='--',
    linewidth=1.2, zorder=0)
plt.plot(solotions, color=qing, alpha = 0.9, label='root', linewidth=1.2,
    zorder=1)
plt.scatter(0, solotions[0], marker = 'x', zorder=2, color = lan, label='
    initial guess')
plt.scatter(len(solotions)-1, solotions[-1], marker = 'x', color = huang,
    zorder=2, label='final solution')


plt.xlabel('iterations', fontsize=14, color='black')
plt.ylabel('root', fontsize=14, color='black')
plt.grid(True, which='both', axis='both', linestyle='--', linewidth=0.8,
    alpha=0.6, color='gray')
plt.tick_params(axis='both', which='major', labelsize=11, colors='black')
plt.xticks(range(0, len(solotions)+1, 100))
plt.title(f'RM algorithm for solving $x^3-2=0$, the real solution is s={
    real_solution}')
plt.legend()
plt.savefig('RM.png', dpi=300, transparent=True)
plt.show()
```

First updated: 24 January, 2025
Last updated: 12 May, 2025

References