26. Stochastic Approximation and SGD

26.1 Introduction

In previous chapters we studied maximum likelihood estimation (MLE) as a method for estimating unknown parameters in a statistical model. Given independent observations

$$Y_1, \ldots, Y_n \sim p(y; \theta),$$

the MLE is defined as the parameter value that maximizes the log-likelihood

$$\ell_n(\theta) = \frac{1}{n} \sum_{i=1}^{n} \log p(Y_i; \theta).$$

In many classical models the maximization can be performed analytically. For example, the MLE of the mean of a Gaussian distribution is the sample average, and the MLE of the rate parameter of an exponential distribution is the reciprocal of the sample mean. In more complicated models, however, the likelihood equation must be solved numerically.

Robbins and Monro (1951) proposed a remarkably different approach. Instead of maximizing the likelihood directly, they showed that one can estimate the parameter recursively by processing one observation at a time and using only the score function, that is, the derivative of the log-likelihood.

Their work initiated the field of *stochastic approximation*. Today, stochastic approximation is mainly remembered because it provides the theoretical foundation of stochastic gradient descent (SGD), one of the most important algorithms in modern machine learning.

Modern SGD is best viewed as the descendant of Robbins–Monro after more than seventy years of theoretical and algorithmic refinements. Contemporary optimization algorithms typically incorporate mini-batches, momentum terms, adaptive step sizes, averaging, preconditioning, projections, and reparametrizations.

For this reason, Robbins–Monro should not be regarded primarily as a practical algorithm for computing MLEs. Rather, it should be viewed as the theoretical prototype showing that estimating equations can be solved using only noisy observations. Most practical algorithms used today are sophisticated descendants of this original idea.

The goal of this chapter is twofold. First, we introduce the Robbins–Monro algorithm in the context of maximum likelihood estimation. Second, we explain how this classical method evolved into the stochastic gradient algorithms that are now ubiquitous in machine learning.

26.2 A Recursive Estimator of the Mean

To motivate the Robbins–Monro method, let us begin with a very simple problem.

Suppose that

$$Y_1, Y_2, \dots$$

are independent and identically distributed random variables with mean

$$\mu = E(Y).$$

The usual estimator of μ based on the first n observations is the arithmetic mean

$$\widehat{\mu}_n = \frac{1}{n}\sum_{i=1}^{n} Y_i.$$

When a new observation Y_n becomes available, we can update the estimate without recomputing the entire sum:

$$\begin{aligned}\widehat{\mu}_n &= \frac{Y_1 + \cdots + Y_{n-1} + Y_n}{n} \\ &= \frac{n-1}{n}\left(\frac{Y_1 + \cdots + Y_{n-1}}{n-1}\right) + \frac{1}{n}Y_n \\ &= \frac{n-1}{n}\widehat{\mu}_{n-1} + \frac{1}{n}Y_n.\end{aligned}$$

Subtracting $\widehat{\mu}_{n-1}$ from both sides gives

$$\widehat{\mu}_n = \widehat{\mu}_{n-1} + \tfrac{1}{n}\left(Y_n - \widehat{\mu}_{n-1}\right). \tag{26.1}$$

Equation (26.1) is remarkable because it updates the estimate using only:

- the previous estimate $\widehat{\mu}_{n-1}$,
- the new observation Y_n,
- a weight $1/n$.

No storage of past observations is required, only the last current estimate $\widehat{\mu}_{n-1}$.

The sequence of weights

$$a_n = \frac{1}{n}$$

satisfies three important conditions:

$$\lim_{n\to\infty} a_n = 0 \text{ (because } 1/n \to 0 \text{ as } n \to \infty) \tag{26.2}$$

$$\sum_{n=1}^{\infty} a_n = \infty \text{ (the harmonic series } A_N = \sum_{1}^{N} 1/n \text{ diverges to } \infty) \tag{26.3}$$

$$\sum_{n=1}^{\infty} a_n^2 < \infty \text{ (the Basel problem series } B_N = \sum_{1}^{N} 1/n^2 \text{ converges to } \pi^2/6) \tag{26.4}$$

The first condition ensures that the effect of a single observation eventually becomes negligible. The second condition ensures that new observations continue to influence the estimate indefinitely. The weights do not decrease too fast toward zero. However, the third condition ensures that it is not either too slow the convergence towards zero. Indeed, this condition ensures the cumulative effect of the random fluctuations remains bounded (maybe delete this sntence ???) I do not resist pointing out the beautiful history of the Basel problem series that was famously solved by Leonhard Euler in the 18th century (see the delightful book by William Dunham [2]).

These three conditions will reappear throughout the chapter and play a fundamental role in the convergence of the Robbins–Monro algorithm.

The recursive mean estimator is therefore our first example of a stochastic approximation procedure.

26.3 MLE as a Root-Finding Problem

Maximum likelihood estimation is usually presented as an optimization problem. Equivalently, it can be viewed as a root-finding problem.

Let

$$\ell_n(\theta) = \frac{1}{n}\sum_{i=1}^{n} \log p(Y_i;\theta)$$

be the average log-likelihood.

The MLE satisfies

$$\nabla_\theta \ell_n(\theta) = 0.$$

Define the *individual score function* based on a single observation Y_i:

$$s(Y_i,\theta) = \nabla_\theta \log p(Y_i;\theta). \tag{26.5}$$

Then the score of the complete sample is

$$U_n(\theta) = \nabla_\theta \ell_n(\theta) = \frac{1}{n}\sum_{i=1}^{n} s(Y_i,\theta). \tag{26.6}$$

The likelihood equation can therefore be written as

$$U_n(\theta) = 0.$$

Thus the MLE is the value of θ that makes the empirical score equal to zero.

Example: Gaussian Mean with Known Variance

Suppose

$$Y_i \sim N(\mu,\sigma^2),$$

where σ^2 is known. The log-likelihood of a single observation is

$$\log p(y;\mu) = -\frac{1}{2}\log(2\pi\sigma^2) - \frac{(y-\mu)^2}{2\sigma^2}.$$

Differentiating with respect to μ, we have

$$s(y,\mu) = \frac{y-\mu}{\sigma^2}.$$

The likelihood equation becomes

$$\frac{1}{n}\sum_{i=1}^{n}\frac{Y_i-\mu}{\sigma^2}=0.$$

Multiplying by σ^2,

$$\frac{1}{n}\sum_{i=1}^{n}(Y_i-\mu)=0,$$

which yields

$$\widehat{\mu}_{MLE}=\overline{Y}.$$

In this example the MLE is available in closed form. In many models, however, solving the likelihood equation is considerably more difficult.

26.4 The Robbins–Monro Algorithm for MLE

The Robbins–Monro idea is deceptively simple. Instead of solving

$$U_n(\theta)=0$$

using all observations simultaneously, we process one observation at a time and update the estimate recursively. Let

$$\widehat{\theta}_{n-1}$$

be the current estimate. After observing the n-th data point Y_n, compute the score contribution

$$s(Y_n,\widehat{\theta}_{n-1})$$

and update the estimate according to

$$\widehat{\theta}_n=\widehat{\theta}_{n-1}+a_n s(Y_n,\widehat{\theta}_{n-1}), \tag{26.7}$$

where the sequence a_n satisfies conditions (26.2)–(26.4). Equation (26.7) is the Robbins–Monro estimator specialized to maximum likelihood estimation.

The intuition is straightforward. If the score evaluated at the current estimate is positive, the estimate should increase. If the score is negative, the estimate should decrease. The sequence a_n controls the magnitude of these corrections.

Example: Gaussian Mean with Known Variance

Suppose again that

$$Y_i\sim N(\mu,3^2).$$

The individual score is

$$s(Y_i,\mu)=\frac{Y_i-\mu}{9}.$$

Substituting into (26.7),

$$\widehat{\mu}_n=\widehat{\mu}_{n-1}+a_n\frac{Y_n-\widehat{\mu}_{n-1}}{9}.$$

Choosing $a_n=9/n$ we can eliminate the 9 in the denominator and obtain

$$\widehat{\mu}_n=\widehat{\mu}_{n-1}+\frac{1}{n}\left(Y_n-\widehat{\mu}_{n-1}\right),$$

which is exactly the recursive sample mean derived in the previous section. Thus, in this simple Gaussian model, Robbins–Monro reproduces the ordinary MLE.

Simulation in R

The following code illustrates the convergence of the Robbins–Monro estimator.

```
set.seed(123)

mu_true <- 5
sigma <- 3
n <- 5000

y <- rnorm(n, mu_true, sigma)

mu <- numeric(n)
mu[1] <- 0

for(k in 2:n){

a <- 9/k

score <- (y[k]-mu[k-1])/9

mu[k] <- mu[k-1] + a*score

}

plot(mu, type="l",
xlab="Iteration",
ylab=expression(hat(mu)))

abline(h=mu_true, lty=2)
```

Figure 26.1 shows a typical realization. The estimator fluctuates considerably during the first iterations but eventually stabilizes around the true parameter value.

In the next section we investigate why this simple recursion converges and what quantity it is actually estimating.

26.5 Foundations: Population and Empirical Score Functions

In the previous section we introduced the Robbins–Monro estimator

$$\widehat{\theta}_n = \widehat{\theta}_{n-1} + a_n s(Y_n, \widehat{\theta}_{n-1}),$$

where

$$s(Y, \theta) = \nabla_\theta \log p(Y; \theta)$$

is the individual score function (caculated with a single data point Y).

At this point a natural question arises:

Question

What quantity is Robbins–Monro actually trying to estimate?

To answer this question, we must distinguish between the empirical score function computed from a finite sample and its theoretical counterpart defined in the population.

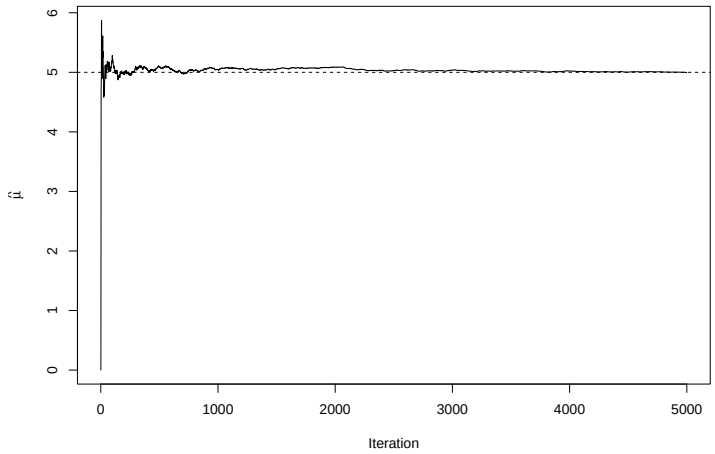

Figure 26.1: Robbins–Monro estimator for the mean of a Gaussian distribution with known variance.

26.5.1 The Empirical Score Function

Suppose that $Y_1, \dots, Y_n$ are independent observations from a parametric model $p(y;\theta)$ with $\theta \in \Theta$. The average log-likelihood is

$$\ell_n(\theta) = \frac{1}{n}\sum_{i=1}^{n} \log p(Y_i;\theta).$$

Its gradient is

$$U_n(\theta) = \nabla_\theta \ell_n(\theta) = \frac{1}{n}\sum_{i=1}^{n} s(Y_i,\theta),$$

where

$$s(Y_i,\theta) = \nabla_\theta \log p(Y_i;\theta)$$

is the score contribution of the (i)-th observation. The function $U_n(\theta)$ is called the *empirical score function*. It is random because it depends on the random sample. The maximum likelihood estimator is obtained by solving

$$U_n(\theta) = 0.$$

Because $U_n(\theta)$ depends on the observed sample, its root changes from one sample to another.

26.5.2 The Population Score Function

To understand the limiting behavior of the MLE, suppose that there exists a fixed parameter value θ^* that truly generates the data. Thus $Y_i \sim p(y;\theta^*)$. The value θ^* is unknown but fixed.

Now consider a single random variable $Y \sim p(y;\theta^*)$. For any candidate value $\theta \in \Theta$, define

$$G(\theta) = \mathbb{E}_{\theta^*}\left[s(Y,\theta)\right]. \tag{26.8}$$

The expectation is taken with respect to the true distribution $p(y;\theta^*)$, while the score is evaluated at an arbitrary value θ. The function $G(\theta)$ is called the *population score function*. Unlike the empirical score, $G(\theta)$ is not random.

26.5.3 The Difference Between θ and θ^*

Students often find the notation in (26.8) confusing because two different parameter symbols appear simultaneously. The distinction is fundamental.

- The symbol θ^* denotes the unknown parameter value that generated the data.
- The symbol θ denotes an arbitrary point in the parameter space where the score function is evaluated.

The expectation operator uses θ^* because it determines the distribution of the random variable Y. The argument of the score function is θ because this is the quantity we are varying in order to search for a root. Thus, throughout this section, θ^* is fixed (but unknown), and θ varies.

26.5.4 Example: Exponential Distribution

Suppose that

$$Y_i \sim \text{Exp}(\lambda^*)$$

where $\lambda^* > 0$ is the unknown true rate parameter.

The density is

$$p(y;\lambda) = \lambda e^{-\lambda y}, \qquad y > 0.$$

The log-density is

$$\log p(y;\lambda) = \log \lambda - \lambda y.$$

Differentiating with respect to λ,

$$s(y,\lambda) = \frac{1}{\lambda} - y.$$

Therefore, the population score function is

$$G(\lambda) = E_{\lambda^*}\left[\frac{1}{\lambda} - Y\right].$$

Since

$$E_{\lambda^*}(Y) = \frac{1}{\lambda^*},$$

we obtain

$$G(\lambda) = \frac{1}{\lambda} - \frac{1}{\lambda^*}. \tag{26.9}$$

The root of this function is obtained by solving

$$G(\lambda) = 0.$$

Using (26.9), we have

$$\frac{1}{\lambda} - \frac{1}{\lambda^*} = 0,$$

which implies

$$\lambda = \lambda^*.$$

Thus the true parameter value λ^* is the unique root of the population score equation $G(\lambda) = 0$. That is, the unique value of λ that solves $G(\lambda) = 0$ is λ^*.

26.5.5 Why the Population Score Matters

The empirical score function can be written as an arithmetic mean of i.i.d. random variables:

$$U_n(\theta) = \frac{1}{n}\sum_{i=1}^{n} s(Y_i, \theta).$$

For fixed θ, the Law of Large Numbers implies that this arithmetic mean converges to the population mean, the expected value of the random variables being averaged:

$$U_n(\theta) \to E_{\theta^*}[s(Y, \theta)] = G(\theta).$$

Therefore, the empirical score $U_n(\theta)$ converges to the population score $G(\lambda)$. This observation is one of the cornerstones of likelihood theory. As the sample size increases, the random function $U_n(\theta)$ approaches the deterministic function $G(\theta)$. Consequently, the root of the empirical score approaches the root of the population score. Since the unique root of $G(\theta)$ is θ^*, the maximum likelihood estimator converges to the true parameter value.

26.5.6 The Target of Robbins–Monro

We can now understand the objective of the Robbins–Monro algorithm. The recursion

$$\widehat{\theta}_n = \widehat{\theta}_{n-1} + a_n s(Y_n, \widehat{\theta}_{n-1})$$

does not attempt to solve the empirical equation

$$U_n(\theta) = 0$$

directly. Instead, it uses successive noisy observations of the score function to move toward the root of the population equation

$$G(\theta) = 0$$

which is the true parameter value θ^* that generates the data. In other words, Robbins–Monro replaces the average score

$$\frac{1}{n}\sum_{i=1}^{n} s(Y_i, \theta)$$

by a sequence of single-observation score contributions

$$s(Y_n, \theta),$$

and uses them to track the same population target. This point of view is fundamental because it transforms a deterministic root-finding problem into a stochastic dynamical system. The convergence of this dynamical system is the subject of the next section.

26.6 OPTIONAL: Why Robbins–Monro Works: a Proof Sketch

The Robbins–Monro estimator is defined recursively by

$$\widehat{\theta}_n = \widehat{\theta}_{n-1} + a_n s(Y_n, \widehat{\theta}_{n-1}),$$

where

$$a_n \to 0, \qquad \sum_{n=1}^{\infty} a_n = \infty, \qquad \sum_{n=1}^{\infty} a_n^2 < \infty.$$

Rather than starting with the general theorem, we first examine two simple models. The first one shows that Robbins–Monro reduces to an estimator that students already know. The second one shows the same idea in a nonlinear score equation.

26.6.1 Gaussian Mean with Known Variance

Suppose

$$Y_i \sim N(\mu^*, \sigma^2),$$

where σ^2 is known and μ^* is unknown.

The score for one observation is

$$s(y, \mu) = \frac{y - \mu}{\sigma^2}.$$

The Robbins–Monro recursion is

$$\widehat{\mu}_n = \widehat{\mu}_{n-1} + a_n \frac{Y_n - \widehat{\mu}_{n-1}}{\sigma^2}.$$

If we choose

$$a_n = \frac{\sigma^2}{n},$$

then

$$\widehat{\mu}_n = \widehat{\mu}_{n-1} + \frac{1}{n}(Y_n - \widehat{\mu}_{n-1}).$$

But this is exactly the recursive formula for the sample mean:

$$\widehat{\mu}_n = \overline{Y}_n.$$

Indeed, if $\widehat{\mu}_{n-1} = \overline{Y}_{n-1}$, then

$$\widehat{\mu}_n = \overline{Y}_{n-1} + \frac{1}{n}(Y_n - \overline{Y}_{n-1}) = \frac{n-1}{n}\overline{Y}_{n-1} + \frac{1}{n}Y_n = \overline{Y}_n.$$

By the Law of Large Numbers,

$$\overline{Y}_n \to \mu^*$$

with probability one.

Therefore, in this model, the convergence of Robbins–Monro follows directly from the Law of Large Numbers.

This example reveals the essential idea: Robbins–Monro updates the current estimate by moving it in the direction suggested by the newest score contribution. When the step size is chosen appropriately, the recursion coincides with a familiar consistent estimator.

26.6.2 A Nonlinear Example: Exponential Rate

Now suppose

$$Y_i \sim \text{Exp}(\lambda^*),$$

where $\lambda^* > 0$ is unknown.

The score for one observation is

$$s(y, \lambda) = \frac{1}{\lambda} - y.$$

The population score is

$$G(\lambda) = E_{\lambda^*}\left[\frac{1}{\lambda} - Y\right] = \frac{1}{\lambda} - \frac{1}{\lambda^*}.$$

The unique root of $G(\lambda) = 0$ is

$$\lambda = \lambda^*.$$

The Robbins–Monro recursion is

$$\widehat{\lambda}_n = \widehat{\lambda}_{n-1} + a_n \left(\frac{1}{\widehat{\lambda}_{n-1}} - Y_n \right).$$

This recursion is more delicate than the Gaussian case because the score is nonlinear and becomes unstable when $\widehat{\lambda}_{n-1}$ is close to zero.

To understand the deterministic force behind the recursion, ignore the random fluctuation temporarily and replace the observed score by its expectation:

$$\widehat{\lambda}_n \approx \widehat{\lambda}_{n-1} + a_n G(\widehat{\lambda}_{n-1}).$$

Since

$$G(\lambda) = \frac{1}{\lambda} - \frac{1}{\lambda^*},$$

we have:

$$G(\lambda) > 0 \quad \text{if} \quad \lambda < \lambda^*,$$

and

$$G(\lambda) < 0 \quad \text{if} \quad \lambda > \lambda^*.$$

Therefore, the deterministic part of the recursion pushes λ upward when it is below λ^* and downward when it is above λ^*.

This is the stabilizing mechanism behind Robbins–Monro.

However, the individual score

$$\frac{1}{\widehat{\lambda}_{n-1}} - Y_n$$

contains random noise. The step-size conditions ensure that this noise is gradually dampened while the cumulative deterministic drift remains strong enough to move the estimate toward the root.

This example also shows why practical implementations often require reparametrization. Setting

$$\eta = \log \lambda$$

gives the more stable recursion

$$\eta_n = \eta_{n-1} + a_n \left(1 - e^{\eta_{n-1}} Y_n\right).$$

The estimate of the original parameter is then

$$\widehat{\lambda}_n = e^{\eta_n}.$$

26.6.3 The General Mechanism

The previous examples show the two main ingredients in Robbins–Monro convergence.

First, the expected score must point toward the true parameter. In the scalar case this means that

$$(\theta - \theta^*)G(\theta) < 0$$

whenever $\theta \neq \theta^*$ and θ is close to θ^*.

Second, the random fluctuations must be controlled by the step sizes.

To see this, write

$$s(Y_n, \widehat{\theta}_{n-1}) = G(\widehat{\theta}_{n-1}) + \varepsilon_n,$$

where

$$\varepsilon_n = s(Y_n, \widehat{\theta}_{n-1}) - G(\widehat{\theta}_{n-1}).$$

Then

$$E(\varepsilon_n \mid Y_1, \ldots, Y_{n-1}) = 0.$$

Thus the recursion becomes

$$\widehat{\theta}_n = \widehat{\theta}_{n-1} + a_n G(\widehat{\theta}_{n-1}) + a_n \varepsilon_n.$$

The term $a_n G(\widehat{\theta}_{n-1})$ is the systematic drift toward the root.

The term $a_n \varepsilon_n$ is random noise.

The condition

$$\sum a_n = \infty$$

ensures that the systematic drift has enough cumulative strength to reach the root.

The condition

$$\sum a_n^2 < \infty$$

ensures that the accumulated random noise remains controlled.

This is the central idea of the Robbins–Monro theorem.

26.6.4 A Simplified Convergence Theorem

Theorem 26.6.1 — Simplified Robbins–Monro Theorem. Suppose that θ is scalar and that the population score

$$G(\theta) = E_{\theta^*}[s(Y, \theta)]$$

has a unique root at θ^*. Assume that, in a neighborhood of θ^*,

$$(\theta - \theta^*)G(\theta) < 0$$

whenever $\theta \neq \theta^*$. Also assume that the score has finite variance and that

$$a_n \to 0, \qquad \sum a_n = \infty, \qquad \sum a_n^2 < \infty.$$

Then, under suitable regularity conditions,

$$\widehat{\theta}_n \to \theta^*$$

with probability one.

This theorem formalizes the intuition developed above. The expected score points toward the true parameter, while the decreasing step sizes prevent the random noise from dominating the recursion.

26.7 OPTIONAL: A More Formal Proof Sketch

The Robbins–Monro estimator is defined recursively by

$$\widehat{\theta}_n = \widehat{\theta}_{n-1} + a_n s(Y_n, \widehat{\theta}_{n-1}),$$

where

$$a_n \to 0, \qquad \sum_{n=1}^{\infty} a_n = \infty, \qquad \sum_{n=1}^{\infty} a_n^2 < \infty.$$

The objective of this section is to explain why this recursion converges to the true parameter value. Rather than presenting a fully rigorous proof, we provide a proof sketch that captures the main ideas.

26.7.1 The Error Process

Define the estimation error

$$e_n = \widehat{\theta}_n - \theta^*.$$

Subtracting θ^* from both sides of the Robbins–Monro recursion gives

$$e_n = e_{n-1} + a_n s(Y_n, \widehat{\theta}_{n-1}).$$

??? The behavior of the algorithm is therefore completely determined by the behavior of the score function near the true parameter value.

26.7.2 Separating Signal and Noise

Recall that the population score function is

$$G(\theta) = E_{\theta^*}[s(Y, \theta)].$$

Define the random fluctuation

$$\varepsilon_n = s(Y_n, \widehat{\theta}_{n-1}) - G(\widehat{\theta}_{n-1}).$$

Then

$$E(\varepsilon_n \mid \mathscr{F}_{n-1}) = 0,$$

where $\mathscr{F}_{n-1}$ denotes all information available up to iteration $n-1$. This is a sophisticated way to say that we are conditioning on all previous data $\mathscr{D}_{n-1} = (Y_1, \ldots, Y_{n-1})$. Thus,

$$s(Y_n, \widehat{\theta}_{n-1}) = G(\widehat{\theta}_{n-1}) + \varepsilon_n.$$

Substituting into the recursion yields

$$e_n = e_{n-1} + a_n G(\widehat{\theta}_{n-1}) + a_n \varepsilon_n. \tag{26.10}$$

This decomposition is fundamental. The term $G(\widehat{\theta}_{n-1})$ is the deterministic component that pushes the estimator toward the true parameter. The term ε_n is the random noise introduced by observing a single data point.

26.7.3 Linearization Around the True Parameter

Since

$$G(\theta^*) = 0,$$

a first-order Taylor expansion gives

$$G(\widehat{\theta}_{n-1}) = G(\theta^*) + G'(\theta^*)(\widehat{\theta}_{n-1} - \theta^*) + o(e*n-1).$$

Therefore,

$$G(\widehat{\theta}*n-1) \approx Me_{n-1},$$

where

$$M = G'(\theta^*).$$

Substituting into (26.10) gives

$$e_n \approx (1 + a_n M)e_{n-1} + a_n \varepsilon_n.$$

This equation has the same structure as a stable autoregressive process with a decreasing noise term.

26.7.4 Role of the Step Sizes

The conditions imposed on the sequence a_n have a natural interpretation. The condition $\sum a_n = \infty$ ensures that the deterministic drift toward θ^* never disappears. The condition $\sum a_n^2 < \infty$ ensures that the accumulated variance of the noise remains finite. Roughly speaking, $\sum a_n$ controls the signal, while $\sum a_n^2$ controls the noise. The Robbins–Monro conditions guarantee that the signal dominates the noise in the long run.

26.7.5 Convergence Theorem

We now state a simplified convergence theorem.

Theorem 26.7.1 — Robbins–Monro Estimator. Suppose that:

1. $G(\theta)$ has a unique root at θ^*;
2. $G(\theta)$ is continuously differentiable in a neighborhood of θ^*;
3. $G'(\theta^*) < 0$;
4. the score function has finite variance;
5. the sequence a_n satisfies

$$a_n \to 0, \qquad \sum a_n = \infty, \qquad \sum a_n^2 < \infty.$$

Then

$$\widehat{\theta}_n \longrightarrow \theta^*$$

with probability one.

The proof requires martingale arguments and is beyond the scope of this book. The important point is that the Robbins–Monro estimator converges to the root of the population score equation even though it observes only one noisy score contribution at a time.

26.7.6 Interpretation

The Robbins–Monro recursion can be viewed as a compromise between two competing goals. On the one hand, the algorithm must continue learning indefinitely. This requires $\sum a_n = \infty$ On the other hand, the effect of the random fluctuations must eventually disappear. This requires $\sum a_n^2 < \infty$ The remarkable insight of Robbins and Monro was that both goals can be achieved simultaneously.

The simplest sequence satisfying these conditions is $a_n = c/n$ where $c > 0$ is any positive constant. This choice remains widely used in stochastic optimization today.

26.8 Multivariate Robbins–Monro

The Robbins–Monro algorithm extends naturally to vector-valued parameters. Suppose that $\theta = (\theta_1, \ldots, \theta_p)^T \in \Theta \subseteq \mathbb{R}^p$ is an unknown parameter vector and that $Y_1, Y_2, \ldots$ are independent observations from a distribution $p(y;\theta)$. The individual score function is now a vector:

$$\mathbf{s}(Y,\theta) = \nabla_\theta \log p(Y;\theta) = \left(\frac{\partial}{\partial \theta_1}, \ldots, \frac{\partial}{\partial \theta_p} \right)^T \log p(Y;\theta).$$

The Robbins–Monro estimator becomes

$$\widehat{\theta}_n = \widehat{\theta}_{n-1} + a_n \mathbf{s}\left(Y_n, \widehat{\theta}_{n-1}\right). \tag{26.11}$$

The interpretation is identical to the scalar case. Each new observation contributes a score vector that pushes the current estimate in the direction suggested by the likelihood. Under regularity conditions, $\widehat{\theta}_n \to \theta^*$ with probability one.

The examples below illustrate how this simple recursion specializes to several familiar models.

26.8.1 Example: Gaussian Distribution with Unknown Mean and Variance

Suppose

$$Y_i \sim N(\mu, \sigma^2),$$

where both parameters are unknown. The log-likelihood of a single observation is

$$\log p(y;\mu,\sigma^2) = -\frac{1}{2}\log(2\pi\sigma^2) - \frac{(y-\mu)^2}{2\sigma^2}.$$

Direct application of Robbins–Monro to σ^2 is problematic because variance must remain positive. To avoid this difficulty we introduce the reparametrization

$$\gamma = \log(\sigma^2).$$

Then

$$\sigma^2 = e^\gamma.$$

The parameter vector becomes

$$\theta = (\mu, \gamma)^T.$$

The score with respect to μ is

$$s_\mu(y,\mu,\gamma) = \frac{y-\mu}{e^\gamma}.$$

Using the chain rule, the score with respect to γ is

$$s_\gamma(y,\mu,\gamma) = \frac{1}{2}\left(\frac{(y-\mu)^2}{e^\gamma} - 1\right).$$

The Robbins–Monro recursion becomes

$$\mu_n = \mu_{n-1} + a_n \frac{Y_n - \mu_{n-1}}{e^{\gamma_{n-1}}},$$

and

$$\gamma_n = \gamma_{n-1} + \frac{a_n}{2}\left(\frac{(Y_n-\mu_{n-1})^2}{e^{\gamma_{n-1}}} - 1\right).$$

The estimate of the variance is then obtained through

$$\widehat{\sigma}_n^2 = e^{\gamma_n}.$$

This example illustrates an important practical lesson. The theoretical Robbins–Monro theorem is stated in terms of arbitrary parameters θ, but numerical stability often requires a suitable reparametrization. In this case, working with $\gamma = \log(\sigma^2)$ automatically guarantees positive variance estimates.

26.8.2 Example: Linear Regression

Consider the linear model $Y_i = \mathbf{x}_i^T\beta + \varepsilon_i$, where $\varepsilon_i \sim N(0,\sigma^2)$ and the covariate vector $\mathbf{x}_i = (x_{i1},\dots,x_{ip})^T$ is observed. Assuming σ^2 is known, the log-likelihood of a single observation is

$$\log p(y_i;\beta) = -\frac{1}{2\sigma^2}\left(y_i - \mathbf{x}_i^T\beta\right)^2 + C.$$

Differentiating,

$$\mathbf{s}(y_i,\beta) = \frac{1}{\sigma^2}\mathbf{x}_i\left(y_i - \mathbf{x}_i^T\beta\right).$$

The Robbins–Monro estimator is therefore

$$\widehat{\beta}_n = \widehat{\beta}_{n-1} + \frac{a_n}{\sigma^2}\mathbf{x}_n\left(y_n - \mathbf{x}*n^T\widehat{\beta}_{n-1}\right). \tag{26.12}$$

This formula has a striking interpretation. The quantity

$$y_n - \mathbf{x}*n^T\widehat{\beta}_{n-1}$$

is simply the prediction error of the current model on the newest observation. The estimate is updated by moving in the direction of the covariate vector, weighted by this prediction error. Notice that equation (26.12) never requires computation of $(X^TX)^{-1}$. This is particularly attractive when the dataset is very large or when observations arrive sequentially. In fact, equation (26.12) can be viewed as an online version of ordinary least squares.

26.8.3 Example: Logistic Regression

Suppose now that $Y_i \in 0, 1$ and that

$$P(Y_i = 1|\mathbf{x}_i) = p_i = \frac{\exp(\mathbf{x}_i^T \beta)}{1 + \exp(\mathbf{x}_i^T \beta)}.$$

The log-likelihood contribution of a single observation is

$$\ell_i(\beta) = y_i \log p_i + (1 - y_i) \log(1 - p_i).$$

Differentiation gives the familiar score function

$$\mathbf{s}(y_i, \beta) = \mathbf{x}_i(y_i - p_i).$$

Substituting into the Robbins–Monro recursion,

$$\widehat{\beta}_n = \widehat{\beta}_{n-1} + a_n \mathbf{x}_n (y_n - \hat{p}_n), \tag{26.13}$$

where

$$\hat{p}_n = \frac{\exp(\mathbf{x}_n^T \widehat{\beta}_{n-1})}{1 + \exp(\mathbf{x}_n^T \widehat{\beta}_{n-1})}.$$

Equation (26.13) should look familiar. It is essentially the stochastic gradient descent algorithm used to fit logistic regression models in machine learning. Each observation produces a prediction error

$$y_n - p_n,$$

and the coefficient vector is updated in the direction of the covariates. Historically, this is one of the clearest examples of how modern stochastic gradient methods emerged from the Robbins–Monro framework.

26.8.4 A Unifying Perspective

The previous examples may appear quite different, but they all share the same structure. Given a statistical model $p(y; \theta)$,

1. Compute the score contribution of a single observation:

   $$\mathbf{s}(y, \theta) = \nabla_\theta \log p(y; \theta).$$

   Usually, it is easy to obtain derivatives.
2. Choose a sequence of step sizes a_n (typically, $a_n = 1/n$).
3. Update the estimate according to

   $$\widehat{\theta}_n = \widehat{\theta}_{n-1} + a_n \mathbf{s}\left(Y_n, \widehat{\theta}_{n-1}\right).$$

The Robbins–Monro algorithm therefore provides a universal recipe for constructing recursive maximum likelihood estimators. In the next section we discuss several practical issues that arise when implementing these recursions in real applications.

26.9 Practical Issues in Robbins–Monro Estimation

The Robbins–Monro theorem establishes that, under suitable assumptions, the recursion

$$\widehat{\theta}_n = \widehat{\theta}_{n-1} + a_n s(Y_n, \widehat{\theta}_{n-1})$$

converges to the true parameter value. However, convergence theorems are asymptotic results. In practical applications, several numerical issues arise that strongly affect the performance of the algorithm. This section discusses some of the most important considerations.

26.9.1 Choice of the Step Size Sequence

The sequence a_n controls how aggressively the estimator reacts to new observations. The classical Robbins–Monro conditions require $a_n \to 0$, and $\sum_{n=1}^{\infty} a_n = \infty$, and $\sum_{n=1}^{\infty} a_n^2 < \infty$. A common choice is $a_n = c/n$ where c is a positive constant.

The value of (c) can have a substantial impact on convergence. If c is too small, the algorithm converges very slowly. If c is too large, the algorithm may oscillate substantially before settling near the solution. Figure **??** illustrates the effect of different choices of c. In practice, the optimal value of c depends on the scale of the score function and the local curvature of the likelihood surface. Since these quantities are usually unknown, some trial and error is often required to identify a value of c that provides a satisfactory compromise between stability and speed of convergence.

26.9.2 Sensitivity to Parameterization

The Robbins–Monro theorem is invariant to smooth reparametrizations. Numerical implementations are not. This distinction is important. Consider the exponential distribution $Y \sim \text{Exp}(\lambda)$. The score function is

$$s(y, \lambda) = \frac{1}{\lambda} - y.$$

Applying Robbins–Monro directly gives

$$\lambda_n = \lambda_{n-1} + a_n \left(\frac{1}{\lambda_{n-1}} - Y_n \right).$$

The problem is that the score becomes arbitrarily large when λ approaches zero. A single unfavorable update may push the estimate close to zero, after which the score becomes enormous and numerical instability follows.

A more stable approach is to work with a reparametrization. Rather than λ, we wrok with

$$\eta = \log \lambda.$$

The score for this new parameter η becomes

$$s_\eta(y, \eta) = 1 - e^{\eta} y.$$

The corresponding recursion is

$$\eta_n = \eta_{n-1} + a_n (1 - e^{\eta_{n-1}} Y_n).$$

The estimate of λ is recovered through $\lambda_n = e^{\eta_n}$. This transformation automatically guarantees positivity and removes the singularity at $\lambda = 0$. The same idea is commonly used when estimating variances, covariance matrices, and other constrained parameters.

26.9.3 Projection Methods

Sometimes the parameter space is naturally constrained. Examples include $\lambda > 0$, or $\sigma^2 > 0$, or probability vectors satisfying

$$p_j \geq 0, \qquad \text{and} \qquad \sum_j p_j = 1.$$

One possibility is to project each update back into the admissible parameter space.

The projected Robbins–Monro recursion is

$$\widehat{\theta}_n = \Pi_\Theta \left(\widehat{\theta}_{n-1} + a_n s(Y_n, \widehat{\theta}_{n-1}) \right),$$

where Π_Θ denotes projection onto the parameter space Θ.

Projection methods are widely used in optimization and machine learning. However, projections must be applied with care. For example, simply truncating an estimate of an exponential rate parameter at a very small positive number can create extremely large scores and lead to numerical instability. For this reason, reparametrization is often preferable whenever a natural transformation is available.

26.9.4 Scaling and Preconditioning

The Robbins–Monro update uses the score directly. Unfortunately, different coordinates may have very different scales. Suppose that

$$\theta = (\theta_1, \theta_2)^T.$$

If the likelihood is much more sensitive to θ_1 than to θ_2, then a single step size a_n may be inappropriate. A common modification is to replace the update $\theta_n = \theta_{n-1} + a_n \mathbf{s}_n$ by

$$\theta_n = \theta_{n-1} + a_n \mathbf{M}_n \mathbf{s}_n,$$

where $\mathbf{M}_n$ is a positive-definite matrix. This idea is closely related to Newton's method and Fisher scoring. Many modern optimization algorithms can be interpreted as adaptive choices of the matrix $\mathbf{M}_n$.

26.9.5 Polyak–Ruppert Averaging

A remarkable discovery made several decades after Robbins and Monro is that averaging successive iterates can substantially improve statistical efficiency. Instead of using the final estimate $\widehat{\theta}_n$, consider the averaged estimator

$$\bar{\theta}_n = \frac{1}{n} \sum_{i=1}^{n} \widehat{\theta}_i.$$

Polyak and Ruppert showed that the averaged estimator often has a smaller asymptotic variance than the original Robbins–Monro estimator. In many problems, $\bar{\theta}_n$ achieves the optimal asymptotic efficiency predicted by classical likelihood theory. For this reason, averaging has become a standard component of stochastic optimization algorithms.

26.9.6 Mini-Batches

The original Robbins–Monro algorithm uses one observation at a time. Modern machine learning algorithms often use a small batch of observations. Let

$$B_n = Y_{n1}, \ldots, Y_{nm}$$

be a batch containing m observations. The update becomes

$$\widehat{\theta}_n = \widehat{\theta}_{n-1} + a_n \frac{1}{m} \sum_{j=1}^{m} s(Y_{nj}, \widehat{\theta}_{n-1}).$$

The resulting estimate of the score has lower variance than a single-observation update while remaining computationally inexpensive. Mini-batches are one of the key ingredients of modern stochastic gradient methods.

26.10 Robbins–Monro and Modern Stochastic Optimization

The Robbins–Monro algorithm was introduced in 1951 as a method for solving equations involving unknown expectations. In the context of this chapter, it provides a recursive procedure for solving the population score equation

$$G(\theta) = E_{\theta^*}[s(Y, \theta)] = 0,$$

whose unique solution is the true parameter value θ^*.

From a modern perspective, the most remarkable aspect of Robbins–Monro is not the specific recursion itself but the principle that it established. The algorithm demonstrated that a deterministic root-finding problem can be solved using only noisy observations collected sequentially. Rather than computing an average score based on the entire dataset, Robbins–Monro updates the estimate using one observation at a time.

This idea was revolutionary. It showed that estimation and optimization can be performed without storing the complete dataset and without repeatedly solving large deterministic problems. These features are particularly attractive when observations arrive continuously or when the dataset is too large to fit comfortably in memory.

26.10.1 The Original Robbins–Monro Problem

Historically, Robbins and Monro did not formulate their algorithm in terms of likelihoods or score functions. Their original objective was to solve the equation

$$m(\theta) = \alpha,$$

where $m(\theta)$ is an unknown function that cannot be observed directly. Instead, for each value of θ, one can observe a random variable $Z(\theta)$ satisfying

$$E[Z(\theta)] = m(\theta).$$

The Robbins–Monro recursion uses successive observations of $Z(\theta)$ to approximate the root of the equation $m(\theta) = \alpha$. The maximum likelihood problem studied in this chapter is a special case obtained by setting $m(\theta) = E_{\theta^*}[s(Y, \theta)]$ and $\alpha = 0$. Thus, likelihood-based estimation represents only one application of a much broader stochastic approximation framework.

26.10.2 From Robbins–Monro to Stochastic Gradient Descent

Modern stochastic gradient descent (SGD) can be viewed as a direct descendant of the Robbins–Monro algorithm. Suppose that we wish to minimize an objective (or loss) function

$$L(\theta) = E[\ell(Y, \theta)]$$

The minimum satisfies

$$\nabla L(\theta) = 0.$$

Using a single observation Y_n, the gradient can be approximated by

$$\nabla_\theta \ell(Y_n, \theta).$$

This leads to the recursion

$$\theta_n = \theta_{n-1} + a_n \nabla_\theta \ell(Y_n, \theta_{n-1}),$$

which is precisely the stochastic gradient descent algorithm.

The similarity with Robbins–Monro is evident. In both cases, a deterministic equation is replaced by noisy observations of its components, and the estimate is updated sequentially.

26.10.3 Why Modern Algorithms Look Different

Although the conceptual foundations remain the same, modern optimization algorithms differ substantially from the original Robbins–Monro procedure.

Several improvements have been developed over the past seventy years:

- mini-batches replace single observations;
- adaptive learning rates adjust the step size automatically;
- momentum terms reduce oscillations;
- averaging techniques improve statistical efficiency;
- preconditioning and second-order methods accelerate convergence;
- reparametrizations and projections improve numerical stability.

As we saw in the exponential and Gaussian examples, even simple likelihood models may require careful choices of parameterization and step size in order to obtain stable numerical behavior. These practical considerations become even more important in high-dimensional optimization problems. Consequently, modern machine learning algorithms should not be viewed as alternatives to Robbins–Monro. Rather, they are sophisticated descendants of the same fundamental idea.

26.10.4 Final Remarks

The Robbins–Monro algorithm occupies a special place in statistics and machine learning. It was one of the first algorithms to exploit randomness as a computational tool rather than treating it merely as a source of uncertainty.

Today, stochastic approximation remains one of the central ideas underlying online estimation, adaptive control, reinforcement learning, and large-scale machine learning. Although modern algorithms are considerably more elaborate than the original recursion proposed in 1951, they continue to embody the same basic principle: a deterministic estimation problem can be solved using noisy observations collected sequentially.

For this reason, Robbins–Monro should be viewed not merely as a historical algorithm, but as the conceptual foundation of modern stochastic optimization.