Chapter 6

Stochastic Approximation

In this next chapter, we will introduce the classical temporal-difference reinforcement learning (RL) algorithms. Before that, we need to press the pause button to get prepared better. That is because the ideas and expressions of temporal-difference algorithms are very different from what we have studied so far in this book. Many readers, who see the temporal-difference algorithms for the first time, often wonder why these algorithms were designed in the first place and why they work effectively. In fact, there is a knowledge gap between the contents in the previous and upcoming chapters. This chapter fills the gap by introducing basic stochastic approximation algorithms. Stochastic approximation refers to a broad class of stochastic iterative algorithms solving root finding or optimization problems. We will see in the next chapter that the temporal-difference algorithms are special stochastic approximation algorithms. As a result, it will be much easier to understand these algorithms.

6.1 Motivating example: mean estimation

Consider a random variable X which takes values in the finite set $\mathcal{X}$. Our aim is to estimate $\mathbb{E}[X]$. Suppose that we collected a sequence of iid samples $\{x_i\}_{i=1}^n$. The expectation of X can be approximated by

$$\mathbb{E}[X] \approx \bar{x} \doteq \frac{1}{n}\sum_{i=1}^{n} x_i. \tag{6.1}$$

The approximation in (6.1) is the basic idea of Monte Carlo estimation. We know that $\bar{x} \to \mathbb{E}[X]$ as $n \to \infty$ according to the Law of Large Numbers as introduced in the last chapter.

The problem that we would like to discuss is how to calculate the mean $\bar{x}$ in (6.1). There are two ways. The first way, which is trivial, is to collect all the samples and then calculate the average. The drawback of such a way is that, if the samples are collected one by one over some time, we have to wait until all the samples are collected. If the sampling time is long, such a wait is a waste of time. The second way can avoid this drawback because it calculates the average in an *incremental* and *iterative* manner. In particular, suppose

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

and hence

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

Then, w_{k+1} can be expressed in terms of w_k as

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

Therefore, we obtain the following iterative algorithm:

$$w_{k+1} = w_k - \frac{1}{k}(w_k - x_k). \tag{6.2}$$

We can use this iterative algorithm to calculate the mean $\bar{x}$ incrementally. It can be verified that

$$\begin{aligned}
w_1 &= x_1, \\
w_2 &= w_1 - \frac{1}{1}(w_1 - x_1) = x_1, \\
w_3 &= w_2 - \frac{1}{2}(w_2 - x_2) = x_1 - \frac{1}{2}(x_1 - x_2) = \frac{1}{2}(x_1 + x_2), \\
w_4 &= w_3 - \frac{1}{3}(w_3 - x_3) = \frac{1}{3}(x_1 + x_2 + x_3), \\
&\vdots \\
w_{k+1} &= \frac{1}{k}\sum_{i=1}^{k} x_i.
\end{aligned} \tag{6.3}$$

An advantage of (6.2) is that a mean estimate can be obtained immediately once a sample is received. Then, the mean estimate can be used for other purposes immediately. Of course, the mean estimate is not accurate in the beginning due to insufficient samples. However, it is better than nothing. As more samples are obtained, the estimation accuracy can be improved gradually according to the Law of Large Numbers.

One can also define $w_{k+1} = \frac{1}{1+k}\sum_{i=1}^{k+1} x_i$ and $w_k = \frac{1}{k}\sum_{i=1}^{k} x_i$. It would not make too much difference. In this case, the corresponding iterative algorithm is $w_{k+1} = w_k - \frac{1}{1+k}(w_k - x_{k+1})$. The details of the derivation is left as an exercise to the reader.

Furthermore, consider an algorithm with a more general expression:

$$w_{k+1} = w_k - \alpha_k (w_k - x_k), \tag{6.4}$$

which is exactly the same as (6.2) except that $1/k$ is replaced by $\alpha_k > 0$. Does this algorithm still converge to the mean $\mathbb{E}[X]$? In fact, the answer is yes if $\{\alpha_k\}$ satisfies some mild conditions as we show in the next section. We will also show that (6.4) is a special stochastic approximation algorithm and also a special stochastic gradient descent algorithm.

The algorithms in (6.2) and (6.4) are the first stochastic iterative algorithms ever introduced in this book. In the next chapter, the reader will see that the temporal-difference algorithms have similar (but more complex) expressions.

6.2 Robbins-Monro Algorithm

Stochastic approximation refers to a broad class of stochastic iterative algorithms solving root finding or optimization problems. Compared to many other root-finding algorithms such as gradient-based methods, stochastic approximation is powerful in the sense that it does not require knowing the expression of the objective function or its derivative.

The Robbins-Monro (RM) algorithm [6] is a pioneering work in the field of stochastic approximation. The famous stochastic gradient descent algorithm is a special form of the RM algorithm. We next introduce its details. Suppose we would like to find the root of the equation

$$g(w) = 0,$$

where $w \in \mathbb{R}$ is the variable to be solved and $g : \mathbb{R} \to \mathbb{R}$ is a function. Many problems can be eventually converted to this root-finding problem. For example, suppose $J(w)$ is an objective function to be minimized. Then, the optimization problem can be converted to $g(w) = \nabla_w J(w) = 0$. Note that an equation like $g(w) = c$ with c as a constant can also be converted to the above equation by rewriting $g(w) - c$ as a new function.

If the expression of g or its derivative is known, there are many numerical algorithms that can solve this problem. However, the problem we are facing is firstly *the expression of the function g is unknown* (for example, the function is represented by an artificial neuron network) and secondly *g cannot be measured or observed accurately.* Suppose we can only obtain a noisy observation of g:

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

where $\eta \in \mathbb{R}$ is the observation error. Here, η may be white noise or structural error [7]. Therefore, the system can be viewed as a *black-box* problem, because only the input w and the noisy output $\tilde{g}(w, \eta)$ are known. Our aim is to solve $g(w) = 0$ from w and $\tilde{g}$.

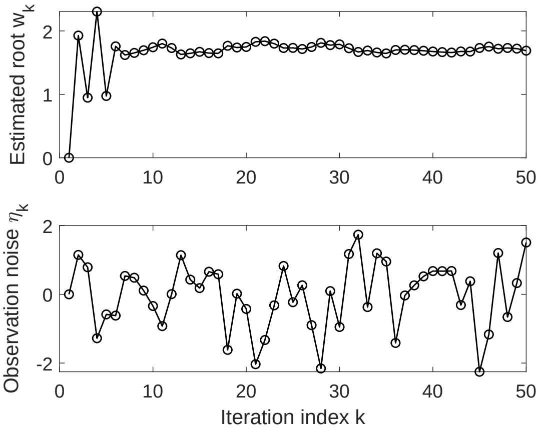

Figure 6.1: An illustrative example of the RM algorithm.

The RM algorithm that can solve the above problem is

$$w_{k+1} = w_k - a_k \tilde{g}(w_k, \eta_k), \qquad k = 1, 2, 3, \dots \tag{6.5}$$

where w_k is the kth estimate of the root, $\tilde{g}(w_k, \eta_k)$ is the kth noisy observation, and a_k is a positive coefficient. As can be seen, the RM algorithm does not require knowing any information about the function. It only requires knowing the input and output data.

To illustrate the RM algorithm, consider an example where $g(w) = w^3 - 5$. The true root is $5^{1/3} \approx 1.71$. If we can only observe the input w and the output $\tilde{g}(w) = g(w) + \eta$, we can use the RM to find the root. In particular, suppose η_k is iid and obeys a standard normal distribution with a mean of zero and standard deviation of 1. The initial guess is $w_1 = 0$ and a_k is selected to be $a_k = 1/k$. The evolution of w_k is shown in Figure 6.1. As can be seen, even though the observation is corrupted by a noise η_k, the estimate w_k can still converge to the true root.

6.2.1 Convergence properties

Why can the RM algorithm in (6.5) find the root of $g(w) = 0$? We first illustrate the idea by an example and then give a rigorous convergence analysis.

Consider the example shown in Figure 6.2. In this example, $g(w) = \tanh(w - 1)$. The true root of $g(w) = 0$ is $w^* = 1$. We apply the RM algorithm with $w_1 = 3$ and $a_k = 1/k$. To better illustrate the reason of convergence, we simply set $\eta_k \equiv 0$. The RM algorithm in this case is $w_{k+1} = w_k - a_k g(w_k)$ since $\tilde{g}(w_k, \eta_k) = g(w_k)$ when $\eta_k = 0$. The resulting $\{w_k\}$ is shown in Figure 6.2. As can be seen, w_k converges to the true root $w^* = 1$.

This simple example can illustrate why the RM algorithm converges.

– When $w_k > w^*$, we have $g(w_k) > 0$. Then, $w_{k+1} = w_k - a_k g(w_k) < w_k$ and hence w_{k+1} is closer to w^* than w_k.

– Similarly, when $w_k < w^*$, we have $g(w_k) < 0$. Then, $w_{k+1} = w_k - a_k g(w_k) > w_k$ and

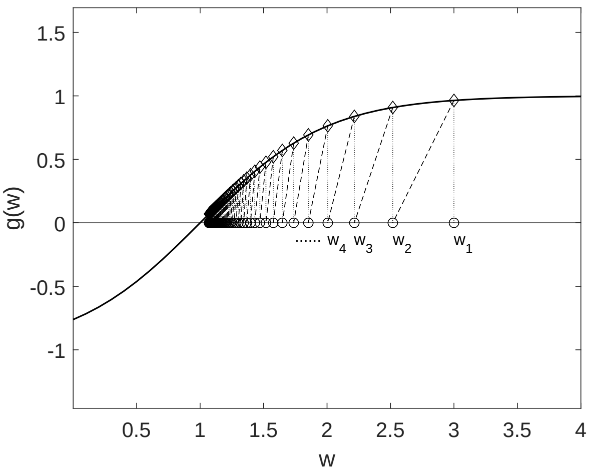

Figure 6.2: An example to illustrate the convergence of the RM algorithm.

w_{k+1} is closer to w^* than w_k.

Therefore, w_{k+1} gets closer to w^* in either case.

As illustrated by the above example, the convergence of the RM algorithm is intuitively straightforward to see. However, it is nontrivial to prove it rigorously in the presence of stochastic observation errors. A rigorous convergence result is given below.

Theorem 6.1 (Robbins-Monro Theorem)**.** *In the Robbins-Monro algorithm, if*

1) $0 < c_1 \leq \nabla_w g(w) \leq c_2$ *for all* w*;*

2) $\sum_{k=1}^{\infty} a_k = \infty$ *and* $\sum_{k=1}^{\infty} a_k^2 < \infty$*;*

3) $\mathbb{E}[\eta_k|\mathcal{H}_k] = 0$ *and* $\mathbb{E}[\eta_k^2|\mathcal{H}_k] < \infty$*;*

where $\mathcal{H}_k = \{w_k, w_{k-1}, \dots\}$*, then* w_k *converges with probability 1 (w.p.1) to the root* w^* *satisfying* $g(w^*) = 0$*.*

We postpone the proof of this result to the next section after a powerful tool for analyzing the convergence of stochastic sequences is introduced. This theorem relies on the notion of "convergence with probability 1", which is introduced in the appendix of this book.

The interpretations of the three conditions in Theorem 6.1 are given below.

– The first condition requires g to be monotonically increasing (or nondecreasing). This condition ensures that the root of $g(w) = 0$ exists and is unique.

– The second condition of $\{a_k\}$ is interesting. We often see conditions like this in RL algorithms. The condition of $\sum_{k=1}^{\infty} a_k^2 < \infty$ requires that a_k must converge to zero as $k \to \infty$. The condition of $\sum_{k=1}^{\infty} a_k = \infty$ requires that a_k should not converge to zero too fast.

– The third condition is a mild condition. The observation error η_k is not required to be Gaussian. A special yet common case is that $\{\eta_k\}$ is an iid stochastic sequence satisfying

$\mathbb{E}[\eta_k] = 0$ and $\mathbb{E}[\eta_k^2] < \infty$. This special case implies the third condition because η_k is independent to $\mathcal{H}_k$ and hence we have $\mathbb{E}[\eta_k|\mathcal{H}_k] = \mathbb{E}[\eta_k] = 0$ and $\mathbb{E}[\eta_k^2|\mathcal{H}_k] = \mathbb{E}[\eta_k^2]$.

Furthermore, we examine the second condition in Theorem 6.1 more closely.

– Why is the second condition important for the convergence of the algorithm?

This question of course can be answered when we present the rigorous proof of the theorem later. Here, we would like to give some insightful intuition.

First, $\sum_{k=1}^{\infty} a_k^2 < \infty$ indicates that $a_k \to 0$ as $k \to \infty$. Why is this condition important? Since

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

if $a_k \to 0$, then $a_k \tilde{g}(w_k, \eta_k) \to 0$ and hence $w_{k+1} - w_k \to 0$, indicating that w_{k+1} and w_k get close to each other when $k \to \infty$. Otherwise, if a_k does not converge, then w_k may still fluctuate when $k \to \infty$.

Second, $\sum_{k=1}^{\infty} a_k = \infty$ indicates that a_k should not converge to zero too fast. Why is this condition important? Summarizing $w_2 = w_1 - a_1 \tilde{g}(w_1, \eta_1)$, $w_3 = w_2 - a_2 \tilde{g}(w_2, \eta_2)$, $\dots$, $w_{k+1} = w_k - a_k \tilde{g}(w_k, \eta_k)$ leads to

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

Suppose $w_\infty = w^*$. If $\sum_{k=1}^{\infty} a_k < \infty$, then $\sum_{k=1}^{\infty} a_k \tilde{g}(w_k, \eta_k)$ may be bounded. Then, if the initial guess w_1 is chosen arbitrarily far away from w^*, then the above equality would be invalid. Therefore, the condition $\sum_{k=1}^{\infty} a_k = \infty$ can make sure that the algorithm converges given an arbitrary initial guess.

– What kinds of sequences satisfy $\sum_{k=1}^{\infty} a_k = \infty$ and $\sum_{k=1}^{\infty} a_k^2 < \infty$?

One typical sequence is

$$\alpha_k = \frac{1}{k}.$$

Why is that? On the one hand, it holds that

$$\lim_{n \to \infty} \left(\sum_{k=1}^{n} \frac{1}{k} - \ln n \right) = \kappa,$$

where $\kappa \approx 0.577$ is called the Euler-Mascheroni constant (also called Euler's constant) [8]. Since $\ln n \to \infty$ as $n \to \infty$, we have

$$\sum_{k=1}^{\infty} \frac{1}{k} = \infty.$$

In fact, $H_n = \sum_{k=1}^{n} 1/k$ has a specific name in the number theory: Harmonic number

[9]. On the other hand, it is notable that

$$\sum_{k=1}^{\infty} \frac{1}{k^2} = \frac{\pi^2}{6} < \infty.$$

The limit $\sum_{k=1}^{\infty} 1/k^2$ also has a specific name in the number theory: Basel problem. Therefore, the sequence $\{1/k\}$ satisfies the second condition in Theorem 6.1. Of course, a slight modification, such as $a_k = 1/k + 1$ or $a_k = c_k/k$ where c_k is bounded and maybe varying, also preserves the condition.

While the RM algorithm is guaranteed to converge when the three conditions in Theorem 6.1 are satisfied, it may still be effective to a certain extent even though some of the conditions are not satisfied. For example, in the example in Figure 6.1, $g(x) = x^3 - 5$ does not satisfy the first condition on gradient boundedness. Nevertheless, the RM algorithm can still find the root if the initial guess is adequately (not arbitrarily) selected. More importantly, we will see that a_k is often selected as a sufficiently small constant in many RL algorithms. Although the second condition is not satisfied in this case, the algorithm can still work effectively.

6.2.2 Application to mean estimation

We next apply the Robbins-Monro Theorem to analyze the mean estimation problem discussed in the first section of this chapter. Recall that

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

is the mean estimation algorithm in (6.4). It was mentioned that while α_k satisfies some mild conditions, this algorithm can converge to $\mathbb{E}[X]$. The convergence proof was not given. We now show that it is a special RM algorithm and hence its convergence naturally follows.

Consider

$$g(w) \doteq w - \mathbb{E}[X].$$

Our aim is to solve $g(w) = 0$. If we can do that, then we obtain the value of $\mathbb{E}[X]$. The observation we can get is

$$\tilde{g}(w, x) \doteq w - x,$$

because we can only obtain a sample x of X. Note that

$$\begin{aligned}\tilde{g}(w,\eta) &= w - x \\ &= w - x + \mathbb{E}[X] - \mathbb{E}[X] \\ &= (w - \mathbb{E}[X]) + (\mathbb{E}[X] - x) \doteq g(w) + \eta,\end{aligned}$$

where $\eta \doteq \mathbb{E}[X] - x$. Therefore, the observation $\tilde{g}(w,\eta)$ is the sum of $g(w)$ and an observation error η. The RM algorithm for solving $g(x) = 0$ is

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

which is exactly the algorithm in (6.4). It is guaranteed by Theorem 6.1 that w_k converges $\mathbb{E}[X]$ with probability 1 if $\sum_{k=1}^{\infty} \alpha_k = \infty$ and $\sum_{k=1}^{\infty} \alpha_k^2 < \infty$ and $\{x_k\}$ is iid.

It should be noted that there is no requirement for the distribution of X. The convergence is guaranteed as long as the samples of X are iid and the variance is bounded according to Theorem 6.1. Moreover, while we can obtain the analytical expression of w_{k+1} as $w_{k+1} = 1/k \sum_{i=1}^{k} x_i$ when $\alpha_k = 1/k$, we would not be able to write out its analytical expression when α_k is general. The convergence analysis is nontrivial in this case and the Robbins-Monro Theorem provides an elegant tool to analyze the convergence.

6.3 Dvoretzky's convergence theorem

Dvoretzky's Theorem, published in 1956 [10], is a classic result in the area of stochastic approximation. It can be used to prove the convergence of the RM algorithm and many RL algorithms.

This section is a little mathematically intensive. Readers that are interested in the convergence analysis of stochastic algorithms are recommended to study this section. Otherwise, this section can be skipped.

Theorem 6.2 (Dvoretzky's Theorem)**.** *Consider a stochastic process*

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

where $\{\alpha_k\}_{k=1}^{\infty}, \{\beta_k\}_{k=1}^{\infty}, \{\eta_k\}_{k=1}^{\infty}$ are stochastic sequences. Here $\alpha_k \geq 0, \beta_k \geq 0$ for all k. Then, w_k would converge to zero with probability 1 if the following conditions are satisfied:

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) $\mathbb{E}[\eta_k | \mathcal{H}_k] = 0$ and $\mathbb{E}[\eta_k^2 | \mathcal{H}_k] \leq C$ w.p.1;

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

Before presenting the proof of this theorem, we first clarify some problems.

- In the RM algorithm, the coefficient sequence $\{\alpha_k\}$ is deterministic. However, Dvoretzky's Theorem allows $\{\alpha_k\}, \{\beta_k\}$ to be random variables depending on $\mathcal{H}_k$. Thus, it is more general and powerful because we may encounter the case where α_k or β_k is a function of w_k or η_k.
- In the first condition, it is stated as "uniformly w.p.1". That is because α_k and β_k may be random variables and hence the definition of their limits must be in the stochastic sense. In the second condition, it is also stated as "w.p.1". This is because $\mathcal{H}_k$ is a sequence of random variables instead of specific values. The definition of the conditional expectation in this case is in the sense of w.p.1.
- The statement of Theorem 6.2 is slightly different from [11] in the sense that Theorem 6.2 does not require $\sum_{k=1}^{\infty} \beta_k = \infty$ in the first condition. Even in the extreme case where $\beta_k = 0$ for all k, the sequence can still converge.

6.3.1 Proof of Dvoretzky's Theorem

The original proof of Dvoretzky's theorem was given in 1956 [10]. There are quite a few ways to prove it. We next present a proof based on quasimartingales inspired by [12].

Proof of Dvoretzky's Theorem. Let $h_k \doteq w_k^2$. Then

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

Taking expectation on both sides of the above equation gives

$$\mathbb{E}[h_{k+1} - h_k|\mathcal{H}_k] = \mathbb{E}[-\alpha_k(2 - \alpha_k)w_k^2|\mathcal{H}_k] + \mathbb{E}[\beta_k^2\eta_k^2|\mathcal{H}_k] + \mathbb{E}[2(1 - \alpha_k)\beta_k r_k w_k|\mathcal{H}_k]. \tag{6.6}$$

Since w_k is determined by $\mathcal{H}_k$, it can be taken out from the expectation. Moreover, suppose α_k, β_k is totally determined by $\mathcal{H}_k$. This is valid if, for example, $\{\alpha_k\}$ and $\{\beta_k\}$ are deterministic sequences or functions of w_k. Then, they can also be taken out from the expectation. Therefore, (6.6) becomes

$$\mathbb{E}[h_{k+1} - h_k|\mathcal{H}_k] = -\alpha_k(2 - \alpha_k)w_k^2 + \beta_k^2\mathbb{E}[\eta_k^2|\mathcal{H}_k] + 2(1 - \alpha_k)\beta_k\mathbb{E}[\eta_k w_k|\mathcal{H}_k]. \tag{6.7}$$

For the first term, since $\sum_{k=1}^{\infty} \alpha_k^2 < \infty$ implies $\alpha_k \to 0$ w.p.1. As a result, there exists a finite n such that $\alpha_k \leq 1$ w.p.1 for all $k \geq n$. Without loss of generality, we can simply consider the case of $k \geq n$ and hence $\alpha_k \leq 1$ w.p.1. Then, $-\alpha_k(2-\alpha_k)w_k^2 \leq 0$. For the second term, we have $\beta_k^2\mathbb{E}[\eta_k^2|\mathcal{H}_k] \leq \beta_k^2 C$ as assumed. For the third term, we have $2(1-\alpha_k)\beta_k\mathbb{E}[\eta_k w_k|\mathcal{H}_k] = 2(1-\alpha_k)\beta_k w_k\mathbb{E}[\eta_k|\mathcal{H}_k] = 0$ since $\mathbb{E}[\eta_k|\mathcal{H}_k] = 0$. Therefore, (6.7) becomes

$$\mathbb{E}[h_{k+1} - h_k|\mathcal{H}_k] = -\alpha_k(2-\alpha_k)w_k^2 + \beta_k^2\mathbb{E}[\eta_k^2|\mathcal{H}_k] \leq \beta_k^2 C \tag{6.8}$$

and hence

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

The last inequality is due to the condition $\sum_{k=1}^{\infty}\beta_k^2 < \infty$. Then, based on the theorem of quasimartingales,we conclude that h_k converges w.p.1.

While we now know that h_k is convergent and so is w_k, we next determine what value w_k converges to. It follows from (6.8) that

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

The first term on the right hand side is bounded as assumed. The second term is bounded because h_k converges and hence $h_{k+1} - h_k$ is summable. Hence, $\sum_{k=1}^{\infty}\alpha_k(2-\alpha_k)w_k^2$ on the left-hand side is also bounded. Since we consider the case of $\alpha_k \leq 1$, we have

$$\infty > \sum_{k=1}^{\infty}\alpha_k(2-\alpha_k)w_k^2 \geq \sum_{k=1}^{\infty}\alpha_k w_k^2 \geq 0.$$

Therefore, $\sum_{k=1}^{\infty}\alpha_k w_k^2$ is bounded. Since $\sum_{k=1}^{\infty}\alpha_k = \infty$, we must have $w_k \to 0$ w.p.1. □

6.3.2 Application to mean estimation

While the mean estimation algorithm, $w_{k+1} = w_k + \alpha_k(x_k - w_k)$, has been analyzed based on the RM Theorem, we next show that its convergence can also be directly proven based on the Dvoretzky's theorem.

Proof. Let $w^* = \mathbb{E}[X]$. The mean estimation algorithm $w_{k+1} = w_k + \alpha_k(x_k - w_k)$ can be rewritten as

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

Let $\Delta \doteq w - w^*$. Then, we have

$$\begin{aligned}\Delta_{k+1} &= \Delta_k + \alpha_k(x_{k+1} - w^* - \Delta_k)\\ &= (1-\alpha_k)\Delta_k + \alpha_k \underbrace{(x_k - w^*)}_{\eta_k}.\end{aligned}$$

Since $\{x_k\}$ is iid, then $\mathbb{E}[x_k|\mathcal{H}_k] = \mathbb{E}[x_k] = w^*$. Moreover, $\mathbb{E}[\eta_k] = \mathbb{E}[x_k - w^*] = 0$ and $\mathbb{E}[\eta_k^2] = \mathbb{E}[x_k^2] - \mathbb{E}[X]^2$ is bounded. Following Dvoretzky's theorem, we conclude that Δ_k converges to zero and hence w_k converges to $w^* = \mathbb{E}[X]$ w.p.1. □

6.3.3 Application to the Robbins-Monro theorem

We are now ready to prove the Robbins-Monro theorem by using Dvoretzky's theorem.

Proof of the Robbins-Monro theorem. The RM algorithm aims to find the root of $g(w) = 0$. Suppose the root is w^* such that $g(w^*) = 0$. The RM algorithm is

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

Then, we have

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

Due to the mean value theorem (Appendix x), we have $g(w_k)-g(w^*) = \nabla_w g(w_k')(w_k - w^*)$, where $w_k' \in [w_k, w^*]$. Let $\Delta_k \doteq w_k - w^*$. The above equation becomes

$$\begin{aligned}\Delta_{k+1} &= \Delta_k - a_k[\nabla_w g(w_k')(w_k - w^*) + \eta_k]\\ &= \Delta_k - a_k\nabla_w g(w_k')\Delta_k + a_k r_k\\ &= [1 - \underbrace{a_k\nabla_w g(w_k')}_{\alpha_k}]\Delta_k + a_k r_k.\end{aligned}$$

Note that $\nabla_w g(w)$ is always bounded by $0 < c_1 \leq \nabla_w g(w) \leq c_2$ as assumed. Since $\sum_{k=1}^{\infty} a_k = \infty$ and $\sum_{k=1}^{\infty} a_k^2 < \infty$ as assumed, we know $\sum_{k=1}^{\infty} \alpha_k = \infty$ and $\sum_{k=1}^{\infty} \alpha_k^2 < \infty$. Thus, all the conditions in Dvoretzky's theorem are satisfied and hence Δ_k converges to zero w.p.1. □

The proof of the RM theorem demonstrates the powerfulness of Dvoretzky's theorem. In particular, α_k in the proof is a stochastic sequence depending on w_k' and w_k rather than a deterministic sequence. In this case, Dvoretzky's theorem is still applicable.

6.3.4 An extension of Dvoretzky's Theorem

We next extend Dvoretzky's theorem to a more general theorem that can handle multiple variables. This general theorem, proposed by [11], can be used to analyze the convergence of stochastic iterative algorithms such as Q-learning and temporal-difference RL algorithms.

Theorem 6.3. *Let x be any element of a set $\mathcal{X}$. For the stochastic iterative process*

$$\Delta_{k+1}(x) = (1 - \alpha_k(x))\Delta_k(x) + \beta_k(x)e_k(x), \tag{6.9}$$

$\Delta_k(x)$ converges to zero w.p.1 if

1) The set $\mathcal{X}$ is finite;

2) $\sum_k \alpha_k(x) = \infty$, $\sum_k \alpha^2(x) < \infty$, $\sum_k \beta_k^2(x) < \infty$, and $\mathbb{E}[\beta_k(x)|\mathcal{H}_k] \leq \mathbb{E}[\alpha_k(x)|\mathcal{H}_k]$ uniformly w.p.1;

3) $\|\mathbb{E}[e_k(x)|\mathcal{H}_k]\|_\infty \leq \gamma\|\Delta_k\|_\infty$, where $\gamma \in (0,1)$;

4) $\mathrm{var}[e_k(x)|\mathcal{H}_k] \leq C(1 + \|\Delta_k(x)\|_\infty)^2$, where C is a constant.

Here, $\mathcal{H}_k = \{\Delta_k, \Delta_{k-1}, \dots, e_{k-1}, \dots, \alpha_{k-1}, \dots, \beta_{k-1}, \dots\}$ stands for the history. The notation $\|\cdot\|_\infty$ refers to the maximum norm.

Proof. As an extension, it can be proven based on Dvoretzky's theorem. Details can be found in [11] and are omitted here. □

Some remarks about this theorem are given below.

- We first clarify some notations in the theorem. The variable x can be viewed as an index. In the context of RL, it often represents a state or state-action pair. Although there is a maximum norm in the conditions, every variable in this theorem is scalar. The maximum norm is defined as $\|\mathbb{E}[e_k(x)|\mathcal{H}_k]\|_\infty = \max_x |\mathbb{E}[e_k(x)|\mathcal{H}_k]|$ and $\|\Delta_k(x)\|_\infty = \max_x |\Delta_k(x)|$. If we put them for different states in a vector, this norm is actually the L_∞ norm of a vector:

$$\|\mathbb{E}[e_k(x)|\mathcal{H}_k]\|_\infty \doteq \max_x |\mathbb{E}[e_k(x)|\mathcal{H}_k]| = \left\| \begin{bmatrix} \vdots \\ \mathbb{E}[e_k(x)|\mathcal{H}_k] \\ \vdots \end{bmatrix} \right\|_\infty .$$

  where different rows of the vectors correspond to different x.

- This theorem is more general than Dvoretzky's theorem. First, it can handle the case of multiple variables due to the maximum norm. This is important for RL problems that have multiple states. Second, while Dvoretzky's theorem requires that $\mathbb{E}[e_k(x)|\mathcal{H}_k] = 0$ and $\text{var}[e_k(x)|\mathcal{H}_k] \leq C$, this theorem only requires that the expectation and variance are bounded by the error Δ_k.

- While (6.9) is merely for a single state, the reason that it can handle multiple states is because of conditions 3 and 4, which are for the entire state space. Moreover, when applying this theorem to prove the convergence of RL algorithms, we need to show that (6.9) is valid for every state.

6.4 Stochastic gradient descent

This section introduces stochastic gradient descent (SGD) algorithms, which are widely used in the field of machine learning. We will show that SGD is a special RM algorithm and the mean estimation algorithm is a special SGD algorithm.

Suppose we would like to solve the following optimization problem:

$$\min_{w} \quad J(w) = \mathbb{E}[f(w, X)], \tag{6.10}$$

where w is the parameter to be optimized and X is a random variable. The expectation is with respect to X. Here, w and X can be either scalars or vectors. The function $f(\cdot)$ is a scalar.

A straightforward method to solve (6.10) is *gradient descent*. Let the gradient of $\mathbb{E}[f(w, X)]$ be $\nabla_w \mathbb{E}[f(w, X)] = \mathbb{E}[\nabla_w f(w, X)]$. Then, the gradient-descent algorithm is

$$w_{k+1} = w_k - \alpha_k \nabla_w \mathbb{E}[f(w_k, X)] = w_k - \alpha_k \mathbb{E}[\nabla_w f(w_k, X)] \tag{6.11}$$

The gradient-descent algorithm can find the optimal solution under some mild conditions such as the convexity of f. An introduction to gradient-descent algorithms is given in Appendix B.

The problem of the gradient descent algorithm in (6.11) is that the expected value on the right-hand side is difficult to calculate. One potential way to calculate the expected value is based on the probability distribution of X. However, the distribution is often unknown in practice. Another way is to collect a large number of iid samples $\{x_i\}_{i=1}^n$ of X so that the expected value can be approximated as

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

Then, (6.11) becomes

$$w_{k+1} = w_k - \frac{1}{n}\sum_{i=1}^{n} \nabla_w f(w_k, x_i). \tag{6.12}$$

This algorithm is called *batch gradient descent (BGD)* because it uses all the samples as a single batch in every iteration.

The problem of the BGD algorithm in (6.12) is that it requires all the samples in each iteration. In practice, since the samples may be collected incrementally, it is favorable to optimize w instantly every time a sample is collected. To that end, we can use the following algorithm:

$$w_{k+1} = w_k - \alpha_k \nabla_w f(w_k, x_k), \tag{6.13}$$

where x_k is the sample collected at time step k. This is the well-known *stochastic gradient descent* algorithm. The reason that this algorithm is called stochastic is that it relies on stochastic samplings $\{x_k\}$.

Compared to the gradient descent algorithm in (6.11), SGD replaces the true gradient $\mathbb{E}[\nabla_w f(w, X)]$ by the *stochastic gradient* $\nabla_w f(w_k, x_k)$. Since $\nabla_w f(w_k, x_k) \neq \mathbb{E}[\nabla_w f(w, X)]$, whether such a replacement can still ensure $w_k \to w^*$ as $k \to \infty$? The answer is yes. We next show some intuitive explanation and will give the rigourous proof of the convergence in Section 6.4.5. Because

$$\begin{aligned} \nabla_w f(w_k, x_k) &= \mathbb{E}[\nabla_w f(w, X)] + \Big(\nabla_w f(w_k, x_k) - \mathbb{E}[\nabla_w f(w, X)]\Big) \\ &\doteq \mathbb{E}[\nabla_w f(w, X)] + \eta_k, \end{aligned}$$

the SGD algorithm can be rewritten as

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

Therefore, SGD is the same as gradient descent except it has a perturbation term $\alpha_k \eta_k$. Since $\{x_k\}$ is iid, we have $\mathbb{E}_{x_k}[\nabla_w f(w_k, x_k)] = \mathbb{E}_X[\nabla_w f(w, X)]$. As a result,

$$\mathbb{E}[\eta_k] = \mathbb{E}\Big[\nabla_w f(w_k, x_k) - \mathbb{E}[\nabla_w f(w, X)]\Big] = \mathbb{E}_{x_k}[\nabla_w f(w_k, x_k)] - \mathbb{E}_X[\nabla_w f(w, X)] = 0.$$

Therefore, the perturbation term η_k has zero mean, which intuitively suggests that it would not jeopardize the convergence. A rigourous proof of the convergence of SGD will be given in Section 6.4.5.

6.4.1 Application to mean estimation

We next apply SGD to analyze the mean estimation problem and show that the mean estimation algorithm in (6.4) is a special SGD algorithm. To that end, formulate the mean estimation problem as an optimization problem:

$$\min_{w} \quad J(w) = \mathbb{E}\left[\frac{1}{2}\|w - X\|^2\right] \doteq \mathbb{E}[f(w, X)], \tag{6.14}$$

where $f(w, X) = \|w - X\|^2/2$ and $\nabla_w f(w, X) = w - X$. It can be verified that the optimal solution is $w^* = \mathbb{E}[X]$.

The gradient descent algorithm for solving (6.14) is

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

The gradient descent algorithm is not applicable here because $\mathbb{E}[w_k - X]$ or $\mathbb{E}[X]$ on the right-hand side is unknown (in fact, it is what we need to solve).

The SGD algorithm for solving (6.14) is

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

where x_k is a sample obtained at time step k. It is noticed this SGD algorithm is the same as the iterative mean estimation algorithm in (6.4). Therefore, (6.4) is an SGD algorithm specifically for solving the mean estimation problem.

6.4.2 Convergence pattern of SGD

SGD uses the stochastic gradient $\nabla_w f(w_k, x_k)$ to approximate the true gradient $\mathbb{E}[\nabla_w f(w_k, X)]$. Since the stochastic gradient is random and hence the approximation is inaccurate, an important question is whether the convergence of SGD is slow or random.

To answer this question, we consider the *relative error* between the stochastic and batch gradients:

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

For the sake of simplicity, we assume all the variables are *scalar*. If δ_k is small, we can expect that SGD behaves similarly to the standard gradient descent. Since w^* is assumed to be the optimal solution and hence $\mathbb{E}[\nabla_w f(w^*, X)] = 0$, the relative error can be written as

$$\delta_k = \frac{|\nabla_w f(w_k, x_k) - \mathbb{E}[\nabla_w f(w_k, X)]|}{|\mathbb{E}[\nabla_w f(w_k, X)] - \mathbb{E}[\nabla_w f(w^*, X)]|} = \frac{|\nabla_w f(w_k, x_k) - \mathbb{E}[\nabla_w f(w_k, X)]|}{|\mathbb{E}[\nabla_w^2 f(\tilde{w}_k, X)(w_k - w^*)]|}. \tag{6.15}$$

where the last equality is due to the mean value theorem and $\tilde{w}_k \in [w_k, w^*]$. Suppose f is strictly convex such that $\nabla_w^2 f \geq c > 0$ for all w, X, where c is a positive bound. Then, the denominator of δ_k in (6.15) becomes

$$\begin{aligned}\big|\mathbb{E}[\nabla_w^2 f(\tilde{w}_k, X)(w_k - w^*)]\big| &= \big|\mathbb{E}[\nabla_w^2 f(\tilde{w}_k, X)](w_k - w^*)\big| \\ &= \big|\mathbb{E}[\nabla_w^2 f(\tilde{w}_k, X)]\big|\big|(w_k - w^*)\big| \\ &\geq c|w_k - w^*|.\end{aligned}$$

Substituting the above inequality to (6.15) gives

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

The above equation suggests an interesting convergence pattern of SGD. In particular, the relative error δ_k is inversely proportional to $|w_k - w^*|$. As a result, when $|w_k - w^*|$ is large, δ_k is small. In this case, the SGD algorithm behaves like the gradient descent algorithm and hence w_k approach w^* quickly. However, when w_k is close to w^*, the relative error δ_k may be large and the convergence exhibits more randomness.

A good example to demonstrate the above analysis is the mean estimation problem. Consider the mean estimation problem in (6.14). When w and X are both scalar, we have $f(w, X) = |w - X|^2/2$ and hence

$$\begin{aligned}\nabla_w f(w, x_k) &= w - x_k, \\ \mathbb{E}[\nabla_w f(w, x_k)] &= w - \mathbb{E}[X] = w - w^*.\end{aligned}$$

As a result, the relative error is

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

The expression of the relative error clearly shows that δ_k is inversely proportional to $|w_k - w^*|$. As a result, when w_k is far away from w^*, the relative error is small and SGD behaves like gradient descent. On the other hand, δ_k is proportional to $|\mathbb{E}[X] - x_k|$. As a result, the mean of δ_k is proportional to the variance of X.

The simulation results in Figure 6.3 illustrate the above analysis. In this simulation, the random variable $X \in \mathbb{R}^2$ represents a two-dimensional position in the plane. Its

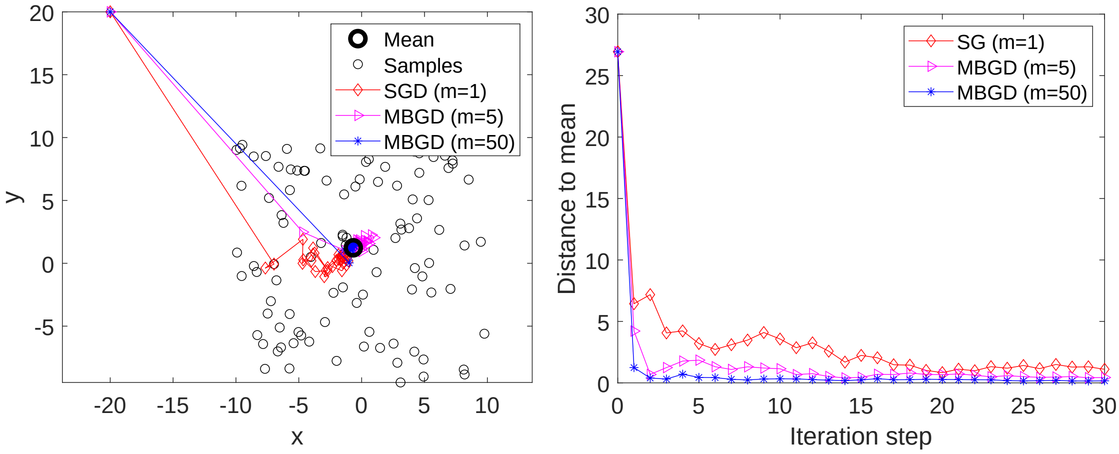

Figure 6.3: An example to demonstrate the convergence process of different gradient descent algorithms. In this example, the random variable $X \in \mathbb{R}^2$ represents a two-dimensional position in the plane. Its distribution is uniform in the square area centered at the origin with the side length as 20. The true mean is $\mathbb{E}[X] = 0$. The mean estimation is based on 100 iid samples.

distribution is uniform in the square area centered at the origin with the side length as 20. The true mean is $\mathbb{E}[X] = 0$. The mean estimation is based on 100 iid samples. As can be seen, although the initial guess of the mean is far away from the true value, the SGD estimate can approach the neighborhood of the true value fast. When the estimate is close to the true value, it exhibits certain randomness but still approaches the true value gradually.

6.4.3 A deterministic formulation of SGD

The formulation of SGD we introduced above involves random variables. One may often encounter a deterministic formulation of SGD without involving any random variables.

Consider a finite set of real numbers $\{x_i\}_{i=1}^n$, where x_i does not have to be a sample of any random variable. The optimization problem to be solved is to minimize the average:

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

where $f(w, x_i)$ is a parameterized function. Here, w is the parameter to be optimized. The gradient descent algorithm for solving this problem is

$$w_{k+1} = w_k - \alpha_k \nabla_w J(w_k) = w_k - \alpha_k \frac{1}{n}\sum_{i=1}^n \nabla_w f(w_k, x_i). \tag{6.16}$$

Suppose the set is large and we can only fetch a single number every time. In this case, we hope to calculate the optimal solution in an incremental manner. Then, we can use the following iterative algorithm:

$$w_{k+1} = w_k - \alpha_k \nabla_w f(w_k, x_k). \tag{6.17}$$

It must be noted that x_k here is the number fetched at time step k instead of the kth element in the set of $\{x_i\}_{i=1}^n$.

The algorithm in (6.17) is very similar to SGD, but the problem formulation is subtly different because it does not involve any random variables or expected values. Then, many questions arise. For example, is this algorithm SGD? How should we use the finite set of numbers $\{x_i\}_{i=1}^n$? Should we sort these numbers in a certain order and then use them one by one? Or should we randomly sample a number from the set?

A quick answer to the above questions is that, although no random variables are involved in the above formulation, we can introduce a random variable manually and convert the *deterministic formulation* to the *stochastic formulation* of SGD as in (6.10). In particular, suppose X is a random variable defined on the set $\{x_i\}_{i=1}^n$. Suppose its probability distribution is uniform such that $p(X = x_i) = 1/n$. Then, the deterministic optimization problem becomes a stochastic one:

$$\min_w \quad J(w) = \frac{1}{n}\sum_{i=1}^n f(w, x_i) = \mathbb{E}[f(w, X)].$$

The last equality in the above equation is strict instead of approximate. Therefore, the algorithm in (6.17) is SGD and the estimate converges if x_k is *uniformly* and independently sampled from $\{x_i\}_{i=1}^n$. Note that x_k may repeatedly take the same number in $\{x_i\}_{i=1}^n$ since it is sampled randomly.

6.4.4 BGD, SGD, and Mini-batch GD

While we have introduced BGD and SGD, this section introduces mini-batch GD (MBGD). The three types of gradient descent algorithms are also compared.

Suppose we would like to minimize $J(w) = \mathbb{E}[f(w, X)]$ given a set of random samples $\{x_i\}_{i=1}^n$ of X. The BGD, SGD, MBGD algorithms solving this problem are, respectively,

$$\begin{aligned}
w_{k+1} &= w_k - \alpha_k \frac{1}{n}\sum_{i=1}^n \nabla_w f(w_k, x_i), \qquad &\text{(BGD)}\\
w_{k+1} &= w_k - \alpha_k \frac{1}{m}\sum_{j\in\mathcal{I}_k} \nabla_w f(w_k, x_j), \qquad &\text{(MBGD)}\\
w_{k+1} &= w_k - \alpha_k \nabla_w f(w_k, \tilde{x}_k). \qquad &\text{(SGD)}
\end{aligned}$$

In the BGD algorithm, all the samples are used in every iteration. When n is large, $(1/n)\sum_{i=1}^n \nabla_w f(w_k, x_i)$ is close to the true gradient $\mathbb{E}[\nabla_w f(w_k, X)]$. In the MBGD algorithm, $\mathcal{I}_k$ is a subset of $\{1, \dots, n\}$ with the size as $|\mathcal{I}_k| = m$. The set $\mathcal{I}_k$ is obtained by

m times idd samplings. In the SGD algorithm, x_k is randomly sampled from $\{x_i\}_{i=1}^n$ at time k.

MBGD can be viewed as an intermediate version between SGD and BGD. Compared to SGD, MBGD has less randomness because it uses more samples instead of just one as in SGD. Compared to BGD, MBGD does not require to use all the samples in every iteration, making it more flexible and efficient. As a result, MBGD well blends the merits of both SGD and BGD while avoiding their shortcomings. If $m = 1$, then MBGD becomes SGD. However, if $m = n$, MBGD does *not* become BGD strictly speaking, because MBGD uses randomly fetched n samples whereas BGD uses all n numbers. In particular, MBGD may use a value in $\{x_i\}_{i=1}^n$ multiple times whereas BGD uses each value once.

The convergence speed of MBGD is faster than SGD in general. That is because SGD uses $\nabla_w f(w_k, x_k)$ to approximate $(1/n)\sum_{i=1}^n \nabla_w f(w_k, x_i)$, whereas MBGD uses $(1/m)\sum_{j\in\mathcal{I}_k} \nabla_w f(w_k, x_j)$, which is more close to the true gradient because the randomness is averaged out. The convergence of the MBGD algorithm can be proved similarly to the SGD case.

A good example to demonstrate the above analysis is the mean estimation problem. In particular, given some numbers $\{x_i\}_{i=1}^n$, our aim is to calculate the mean $\bar{x} = \sum_{i=1}^n x_i/n$. This problem can be equivalently stated as the following optimization problem:

$$\min_w \quad J(w) = \frac{1}{2n}\sum_{i=1}^n \|w - x_i\|^2,$$

whose optimal solution is $w^* = \bar{x}$. The three algorithms for solving this problem are, respectively,

$$\begin{aligned}
&w_{k+1} = w_k - \alpha_k \frac{1}{n}\sum_{i=1}^n (w_k - x_i) = w_k - \alpha_k(w_k - \bar{x}), \qquad \text{(BGD)}\\
&w_{k+1} = w_k - \alpha_k \frac{1}{m}\sum_{j\in\mathcal{I}_k} (w_k - x_j) = w_k - \alpha_k\left(w_k - \bar{x}_k^{(m)}\right), \qquad \text{(MBGD)}\\
&w_{k+1} = w_k - \alpha_k(w_k - x_k), \qquad \text{(SGD)}
\end{aligned}$$

where $\bar{x}_k^{(m)} = \sum_{j\in\mathcal{I}_k} x_j/m$. Furthermore, if $\alpha_k = 1/k$, the above equation can be solved as

$$w_{k+1} = \frac{1}{k}\sum_{j=1}^{k} \bar{x} = \bar{x}, \qquad \text{(BGD)}$$

$$w_{k+1} = \frac{1}{k}\sum_{j=1}^{k} \bar{x}_j^{(m)}, \qquad \text{(MBGD)}$$

$$w_{k+1} = \frac{1}{k}\sum_{j=1}^{k} x_j. \qquad \text{(SGD)}$$

The derivation of the above equations is similar to (6.3) and omitted here. It can be seen that the estimate of BGD at each step is exactly the optimal solution $w^* = \bar{x}$. The MBGD approach the mean faster than SGD because $\bar{x}_k^{(m)}$ is already an average.

A simulation example is given in Figure 6.3 to demonstrate the convergence of MBGD. Let $\alpha_k = 1/k$. As can be seen in Figure 6.3, all MBGD algorithms with different mini-batch sizes can converge to the mean. The case with $m = 50$ converges the fastest, while SGD with $m = 1$ is the slowest. This is consistent with the above analysis. Nevertheless, the convergence rate of SGD is still fast especially when w_k is far away from w^*.

6.4.5 Convergence of SGD

We leave the rigorous proof of the convergence of SGD to the last part of this section.

Theorem 6.4 (Convergence of SGD)**.** *In the SGD algorithm in* (6.13)*, if*

1) $0 < c_1 \leq \nabla_w^2 f(w, X) \leq c_2$*;*

2) $\sum_{k=1}^{\infty} a_k = \infty$ *and* $\sum_{k=1}^{\infty} a_k^2 < \infty$*;*

3) $\{x_k\}_{k=1}^{\infty}$ *is iid;*

then w_k *converges to the root of* $\nabla_w \mathbb{E}_X[f(w, X)] = 0$ *with probability 1.*

The idea of the proof is to show that the SGD algorithm is a special RM algorithm. We can also prove in other ways, for example, based on a more fundamental tool of quasimartingales in rigorously probability theory.

Proof of Theorem 6.4

We next show that the SGD algorithm is a special Robbins-Monro algorithm. Then, the convergence of SGD naturally follows from the Robbins-Monro theorem.

The problem to be solved by SGD is to minimize $J(w) = \mathbb{E}_X[f(w, X)]$. This problem can be converted to a root-finding problem: that is to find the root of

$\nabla_w J(w) = \mathbb{E}_X[\nabla_w f(w, X)] = 0$. Let

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

Then, the aim of SGD is to find the root of $g(w) = 0$. This is exactly the problem solved by the RM algorithm. What we can measure is $\tilde{g}(w, x) = \nabla_w f(w, x)$, where x is a sample of X. Note that $\tilde{g}$ can be rewritten as

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

Then, the RM algorithm for solving $g(w) = 0$ is

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

which is exactly the SGD algorithm. As a result, the SGD algorithm is a special RM algorithm. The convergence of SGD also naturally follows from Theorem 6.1 if the three conditions in Theorem 6.1 are satisfied.

1) Since $\nabla_w g(w) = \nabla_w \mathbb{E}_X[\nabla_w f(w, X)] = \mathbb{E}_X[\nabla_w^2 f(w, X)]$, it follows from $c_1 \leq \nabla_w^2 f(w, X) \leq c_2$ that $c_1 \leq \nabla_w g(w) \leq c_2$. Thus the first condition in Theorem 6.1 is satisfied.

2) The second condition in Theorem 6.1 is the same as the second one in this theorem.

3) The third condition in Theorem 6.1 is $\mathbb{E}[\eta_k|\mathcal{H}_k] = 0$ and $\mathbb{E}[\eta_k^2|\mathcal{H}_k] < \infty$. To see that, since $\{x_k\}$ is idd, we have $\mathbb{E}_{x_k}[\nabla_w f(w, x_k)] = \mathbb{E}_X[\nabla_w f(w, X)]$ for all k. Therefore,

$$\mathbb{E}[\eta_k|\mathcal{H}_k] = \mathbb{E}[\nabla_w f(w_k, x_k) - \mathbb{E}_X[\nabla_w f(w_k, X)]|\mathcal{H}_k].$$

Since $\mathcal{H}_k = \{w_k, w_{k-1}, \dots\}$, the first term is $\mathbb{E}[\nabla_w f(w_k, x_k)|\mathcal{H}_k] = \mathbb{E}_{x_k}[\nabla_w f(w_k, x_k)]$. The second term is $\mathbb{E}[\mathbb{E}_X[\nabla_w f(w_k, X)]|\mathcal{H}_k] = \mathbb{E}_X[\nabla_w f(w_k, X)]$ because $\mathbb{E}_X[\nabla_w f(w_k, X)]$ is a function of w_k. Therefore,

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

Similarly, it can be proved that $\mathbb{E}[\eta_k^2|\mathcal{H}_k] < \infty$ if $|\nabla_w f(w, x)| < \infty$ for all w given any x.

While the three conditions in Theorem 6.1 are satisfied, the convergence of the SGD immediately follows.

6.5 Summary

Instead of introducing new RL algorithms, this chapter introduced the preliminaries to stochastic approximation such as the RM algorithm and the SGD algorithm. Compared to many other root-finding algorithms, the RM algorithm does not require knowing the expression of the objective function or its derivative. It is shown that the SGD algorithm is a special RM algorithm. Moreover, an important problem frequently discussed throughout this chapter is mean estimation. The mean estimation algorithm (6.4) is the first stochastic iterative algorithm we ever introduced in this book. We showed that it is a special SGD algorithm. We will see in the next chapter that the temporal-difference learning algorithms have similar expressions. Finally, the name stochastic approximation was first used by Robbins-Monro's paper in 1951 [6]. A rigorous treatment of stochastic approximation can be found in [7].

6.6 Q&A

- Q: What is stochastic approximation?

  A: Stochastic approximation refers to a broad class of stochastic iterative algorithms solving root finding or optimization problems.

- Q: Why do we need to study stochastic approximation?

  A: That is because the temporal-difference RL algorithms that will be introduced in the next chapter can be viewed as stochastic approximation algorithms. With the knowledge introduced in this chapter, we can be better prepared for the algorithms next chapter. At least, it would not be abrupt for us to see these algorithms for the first time.

- Q: Why do we frequently discuss the mean estimation problem in this chapter?

  A: It is because state value and action value are both defined as means of random variables. In the last chapter, we studied how to approximate means using Monte Carlo methods. In this chapter, we showed that means can be calculated incrementally based on random samples.

- Q: What is the advantage of the RM algorithm compared to other root-finding numerical algorithms?

  A: Compared to many other numerical algorithms for root-finding, the RM algorithm is powerful in the sense that it does not require knowing the expression of the objective function or its derivative. As a result, it is a black-box technique, which only requires knowing the input and output of the objective function. The famous stochastic gradient descent algorithm is a special form of it.

– Q: What is the basic idea of stochastic gradient descent?

A: Stochastic gradient descent aims to solve optimization problems involving random variables. While the probability distributions of the random variables are not known, stochastic gradient descent can solve the optimization problems merely by using samples. Mathematically, it replaces the gradient expressed as an expectation in the gradient descent algorithm with a stochastic gradient.

– Q: Can stochastic gradient descent converge fast?

A: Stochastic gradient descent has an interesting convergence pattern. That is, if the estimate is far away from the optimal solution, then the convergence is fast. When the estimate is close to the solution, the randomness of the stochastic gradient becomes influential and the convergence rate downgrades.

– Q: What is mini-batch gradient descent? What are its advantages compared to SGD and BGD?

A: Mini-batch gradient descent can be viewed as an intermediate version between SGD and BGD. Compared to SGD, it has less randomness because it uses more samples instead of just one as in SGD. Compared to BGD, it does not require using all the samples, which is more flexible and efficient.