Assume we sample uniformly $i_1, \ldots, i_K$ from $\{1, \ldots, n\}$, we can then approximate the sum

$$\sum_{k=1}^{n} \nabla \log p(y_k|X_{n-1}) \approx \frac{n}{K} \sum_{k=1}^{K} \nabla \log p(y_{i_k}|X_{n-1}).$$

Note that (n/K) factor comes here as the sum itself did not have $(1/n)$ term (as opposed to the sum example above). Therefore, the stochastic gradient Langevin dynamics (SGLD) iterate can be written as

$$X_n = X_{n-1} + \gamma \left(\nabla \log p(x) + \frac{n}{K} \sum_{k=1}^{K} \nabla \log p(y_{i_k}|X_{n-1}) \right) + \sqrt{2\gamma} V_n.$$

This is also called *data subsampling* as one can see that the gradient only uses a subset of the data. Every iteration is cheap and computable as we only need to compute K terms. This is a very popular method in Bayesian inference and is used in many applications.

5.6 MCMC FOR OPTIMISATION

MCMC methods were originally motivated by optimisation problems. These methods are a good candidate to solve challenging, nonconvex optimisation problems with multiple minima due to the intrinsic noise in the algorithms. In this section, we will briefly look at two MCMC methods that can be used for optimisation: (i) simulated annealing and (ii) Langevin MCMC.

5.6.1 BACKGROUND

It is important to note that a sampler can be used as an optimiser in the following context. Consider the target density

$$p_\star^\beta(x) \propto \exp(-\beta f(x)),$$

where $\beta > 0$ is a parameter. It is known in the literature that the density $p_\star^\beta(x)$ concentrates around the minima of f as $\beta \to \infty$ (Hwang, 1980). This connection between probability distributions and optimisation spurred the development of MCMC methods for optimisation. In what follows, we describe two methods that exploit this connection.

5.6.2 SIMULATED ANNEALING

Consider now a *sequence* of target distributions defined as

$$p_\star^{\beta_t}(x) \propto \exp(-\beta_t f(x)),$$

where $\beta_t > 0$ is a sequence of increasing parameters. This algorithm *anneals* the target distribution so that $p_\star^{\beta_t}(x)$ becomes concentrated around the minima of f. At the same time, the method uses each distribution as a proposal by an accept-reject mechanism. Drawing from previous section's MH algorithm, we can easily see the simulated annealing (SA) algorithm from Algorithm 13.

Algorithm 13 Simulated Annealing

1: X_0.
2: **for** $t = 1, 2, \ldots$ **do**
3: $X' \sim q(x|X_{t-1})$ (symmetric proposal, e.g., random walk)
4: Accept X_t with probability

$$\min\left\{1, \frac{p_\star^{\beta_t}(X')}{p_\star^{\beta_t}(X_{t-1})}\right\}.$$

5: Otherwise set $X_t = X_{t-1}$.
6: **end for**

One can see that this simulated annealing method takes a special and intuitive case for optimisation. If we look at the acceptance ratio

$$\frac{p_\star^{\beta_t}(X')}{p_\star^{\beta_t}(X_{t-1})} = \frac{\exp(-\beta_t f(X'))}{\exp(-\beta_t f(X_{t-1}))} = \exp(\beta_t(f(X_{t-1}) - f(X'))),$$

we can see that the acceptance ratio is a function of the difference in the objective function values. If $f(X') \leq f(X_{t-1})$, this proposal will take higher values, possibly bigger than 1 depends on the improvement. If, however, $f(X') \geq f(X_{t-1})$, the acceptance ratio will be small as it should be. Scheduling of $(\beta_t)_{t\geq 0}$ is a design problem that depends on the specific cost function under consideration.

Example 5.14. Consider the following challenging cost function

$$f(x) = -\left(\cos(50x) + \sin(20x)\right)^2 \exp(-5x^2), \qquad x \in [-1, 1]. \tag{5.9}$$

This is a function with multiple local minima and is nonconvex. The function has one global minima and we aim at finding it. We implement the SA algorithm with a schedule $\beta_t = \sqrt{1+t}$ where $\beta_t \to \infty$ as t grows. We use a random walk proposal with a standard deviation of $\sigma_q = 0.1$. We implement this on the log domain. We initialise $X_0 \sim \text{Unif}(-1, 1)$. The algorithm is implemented as, given X_{t-1}

- $X' \sim q(x|X_{t-1}) = \mathcal{N}(x; X_{t-1}, \sigma_q^2)$
- Sample $u \sim \text{Unif}(0, 1)$
- Accept if

$$\log(u) < \beta_t(f(X_{t-1}) - f(X'))$$

- Otherwise set $X_t = X_{t-1}$.

The result can be seen from Figure 5.6. We can see that the algorithm is able to find the global minima.

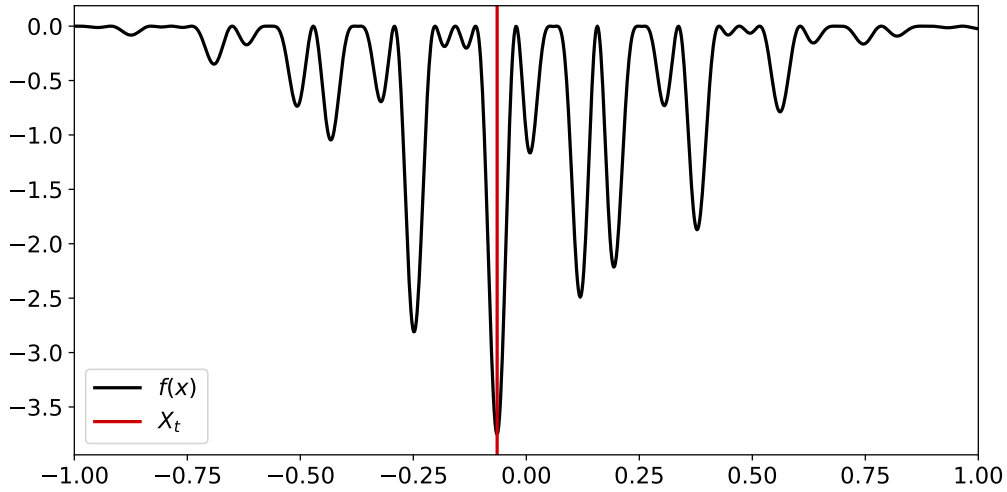

Figure 5.6: Simulated annealing for the function in Eq. 5.9. The red line shows the final estimate of SA algorithm.

5.6.3 LANGEVIN MCMC FOR OPTIMISATION

The family of MCMC methods can be used for optimisation as well. We will showcase one example.

Example 5.15 (ULA for Optimisation)**.** Assume that we try to solve the following problem:

$$\arg\min_{x\in\mathbb{R}} \frac{1}{2\sigma^2}(x-\mu)^2,$$

where μ and σ are known. Of course, (i) we do not really need the scaling factor $\frac{1}{2\sigma^2}$ and (ii) we can simply solve this problem exactly (no surprise, the minimiser is μ). However, as in other examples, this provides us a good setting to showcase the idea of optimisation with MCMC.

We can convert the optimisation problem into a sampling problem by defining the target density as

$$p_\star(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}(x-\mu)^2\right).$$

One can argue that we will not know the normalising constant – which is exactly true. In general, to solve the problem $\min_{x\in\mathbb{R}^d} f(x)$, one constructs a target density $p_\star(x) \propto e^{-f(x)}$. Returning to our example, we do not use directly ULA for this as it would sample from the posterior but would not necessarily give us samples close to minima. For this we resort to a modified version

$$X_{n+1} = X_n + \gamma\nabla \log p_\star(X_n) + \sqrt{\frac{2\gamma}{\beta}}V_n,$$

where β is a parameter that is called the inverse temperature. We can see following the same logic in Example 5.10 that, we have a target distribution

$$p_\star^\beta(x) = \mathcal{N}(x;\mu,\frac{\sigma^4}{\beta(2\sigma^2-\gamma)}).$$

One can see that as $\beta \to \infty$, we have $p_\star^\beta(x) \to \delta_\mu(x)$, i.e., the target distribution is a Dirac delta at μ. This is an example of a more general result where sampling from $p_\star^\beta(x) \propto \exp(-\beta f(x))$ (as it is what the sampler is doing) leads to distributions that concentrate on the minima of $f(x)$ as $\beta \to \infty$.

In our case, for large β, the distribution would be concentrated around μ, that is maximum. Therefore, samples from this distribution would be very close to μ. The error can be verified and quantified in a number of challenging and nonconvex settings (Zhang et al., 2019).