Chapter 10

Optimization

Simulated annealing (SA) is one of the most widely used optimization techniques. The SA method is essentially a random search that only uses the information obtained in the previous iteration. In SA method, each point s of the search space is analogous to a state of some physical system, and the function $E(s)$ to be minimized is analogous to the internal energy of the system in that state. The goal is to bring the system, from *an arbitrary initial state*, to a state with the *minimum possible energy*.

We cover only one optimization method that follows as a corollary from Monte Carlo simulations.

10.1 Simulated annealing

There are many different methods for optimizing different functions. If the function depends on a large number of variables, *simulated annealing* can be used to find an approximation to the optimum.

The idea of simulated annealing is borrowed from what happens when steel (essentially an alloy of iron and carbon) cools down. When the cooling process is fast, only very small crystallites are formed and the material is malleable. To harden steel, it is *annealed*, i.e. heated again for some time and slowly cooled. This allows larger crystals to grow, which make the material elastic. If one anneals too much (too long and slowly), the crystals grow too large, and the material becomes brittle.

Energetically, the formation of large crystals is profitable, i.e. it brings the system closer to the global energy minimum. The small crystallites on the other hand create many local energy minima, and annealing is a way to use thermal fluctuations to "kick" the system out of these local minima and approach the global minimum.

The same ideas can be applied to get close to the global minimum of a function of many variables. As an example, consider the *travelling salesman problem* (e.g., all tours that visit a

given set of cities)[1]: A salesman needs to travel from city C_1 to N other cities $C_2, \ldots C_N$ and back to C_1. He knows the distance between any pair of cities and wants to minimize the total distance

$$D \equiv \sum_i |\mathbf{x}(C_i) - \mathbf{x}(C_{i+1})| \tag{10.1}$$

he has to travel. This optimization problem is *Np*-complete, which means it cannot exactly be solved in reasonable time if N is large.

A simulated-annealing approach (to find an approximation to the global optimum, not the optimum itself) would look like this:

1. Start with an arbitrary cyclic sequence of cities (e.g. sort alphabetically).
2. Choose a "temperature" T and use the Metropolis algorithm to exchange, or not exchange, a randomly picked pair of cities. The acceptance probability is

$$p_{\text{acc}} = e^{-\beta(D_{\text{new}} - D_{\text{old}})} , \tag{10.2}$$

   where $\beta = 1/T$. Repeat this m times.
3. According to a *schedule*, slowly "cool" the system by lowering temperature T. A commonly used schedule is linear in "time",

$$T(t) = T_0 \left(1 - \frac{t}{\Delta t}\right) , \tag{10.3}$$

   where t is a measure of the number of tries.
4. Once temperature is very small or zero, we have an approximation to the global minimum.

Note 1: Slower cooling will (in general) get you closer to the global minimum.

Note 2: Before starting a simulated-annealing scheme, you should estimate which temperature values are relevant. For example, if distances between cities for the travelling-salesman problem vary by a few thousand kilometres, your initial 'temperature' should be at least that large.

On the other hand, once your 'temperature' is 10 times smaller than a typical ΔD, the probability of accepting a change towards larger 'energy' is $\exp(-\Delta D/T) \approx 1/20\,000$, so you will see very few changes below this temperature. A good strategy is to monitor the number of changes for a faster (superficial) cooling schedule and use the results to fix the endpoints of a slower (more in-depth) schedule.

[1]The problem was first formulated as a mathematical problem in 1930 and is one of the most intensively studied problems in optimization. It is used as a benchmark for many optimization methods. Even though the problem is computationally difficult, a large number of heuristics and exact methods are known, so that some instances with tens of thousands of cities can be solved.

[*Genetic algorithms* occupy a similar ecological niche, but they are more complex and a bit tedious to apply for optimizing continuous functions. Both simulated annealing (SA) and the genetic algorithms (GA) are stochastic and derivative-free optimization technique. SA operates on one solution at a time, while the GA maintains a large population of solutions, which are optimized simultaneously. Thus, the genetic algorithm takes advantage of the experience gained in the past exploration of the solution space. Since SA operates on one solution at a time, it has very little history to use in learning from past trials. SA has the ability to escape from any local point; even it is a global optimization technique. On the other hand, there is no guarantee that the GA algorithm will succeeded in escaping from any local minima].

10.2 Appendix

10.2.1 Lab exercise

Question 29 *Simulated annealing*

Use *simulated annealing* to find an approximation to the global energy minimum for the one-dimensional Ising model.

Hint: Use your program(s) from Question 27.

Question 30 *Simulated annealing in the Market (Discussion)*

When applied to portfolio construction, the SA algorithm begins with a randomly selected portfolio. It then iteratively jumps to other portfolios in the neighborhood of its current position in search of better risk-return metrics.

A simple portfolio optimization problem is solved by implementing SA. One assumes that prior knowledge of the distribution of returns (%) in the past is available. Then these values are used to find the optimal investment strategy to build one's portfolio.