Convex Optimization 10-725, Lecture 19: Introduction to non-convex optimization: Simulated Annealing, Evolutionary algorithms

Yuanzhi Li

Assistant Professor, Carnegie Mellon University
Consulting Researcher, Microsoft Research

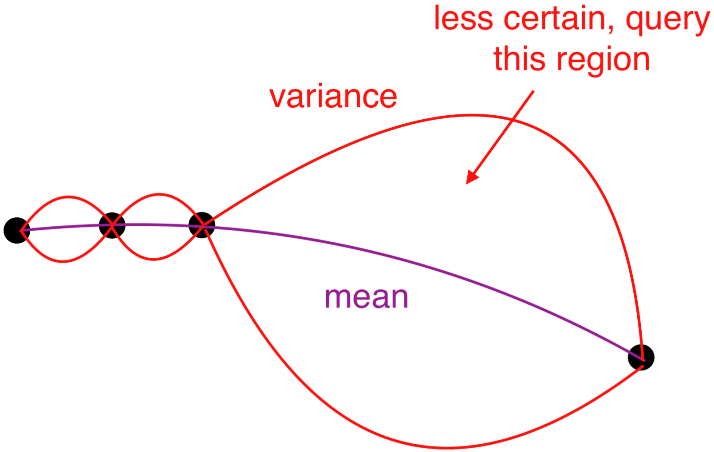
Today

Last lecture

- We learnt the Bayesian optimization.

Last lecture

- “band of possible functions” + “optimistic exploration”.



-

This lecture

- We are going to see some other completely different non-convex optimization algorithms.
- They are in spirit fundamentally different from gradient based optimization or bayesian optimization.

Simulated Annealing

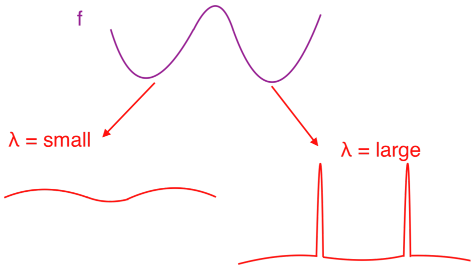
- We introduce the algorithm simulated annealing for non-convex optimization.
- This is a type of evolutionary optimization algorithm.
- The spirit of evolutionary algorithms is very very important in the optimization literature.
- We are going to see how they work.

Simulated Annealing: Motivation

- For a function $f : \mathbb{R}^d \to \mathbb{R}$ over a constraint set $\mathcal{D}$, we can view it as a distribution over $\mathcal{D}$, the density function is given as: (for some $\lambda \geq 0$)

$$p(x) \propto e^{-\lambda f(x)}$$

- Key observation: When λ is close to $= 0$, $p(x)$ is close to a uniform distribution over set $\mathcal{D}$.
- When λ is close to $= +\infty$, then $p(x)$ is close to a distribution uniformly over the global minimizers of f



-

Simulated Annealing: Motivation

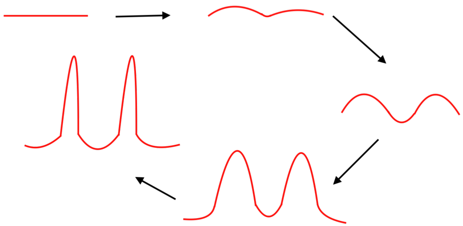
- When λ is close to $+\infty$, $p(x)$ is close to a uniform distribution over the global minimizers of f.
- If we can sample a point x' from distribution

$$p(x) \propto e^{-\lambda f(x)}$$

  for sufficiently large λ, then with high probability $f(x')$ would be close the global minimal value of f.
- Question: How do we sample from $p(x) \propto e^{-\lambda f(x)}$?

Simulated Annealing: Motivation

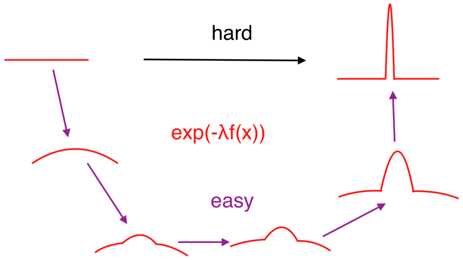
- For a good constraint set $\mathcal{D}$ (for example, the ℓ_2 ball), sampling from a uniform distribution over set $\mathcal{D}$ is easy.
- Sampling from $p(x) \propto e^{-0 \times f(x)}$ is easy.
- Simulated Annealing: Starting from distribution $p(x) \propto e^{-0 \times f(x)}$, we gradually increase the value of λ to $+\infty$ and sample from $p_\lambda(x) \propto e^{-\lambda f(x)}$ at every iteration.
- Here, λ is called the temperature.



-

Simulated Annealing: Motivation

- Simulated Annealing: Starting from distribution $p(x) \propto e^{-0 \times f(x)}$, we gradually increase the value of λ to $+\infty$ and sample from $p_\lambda(x) \propto e^{-\lambda f(x)}$ at every iteration.
- Intuition: When λ_1 is close to λ_2, sampling from $p_{\lambda_2}(x)$ using uniform samples from $p_{\lambda_1}(x)$ as a warm start is easier.



-

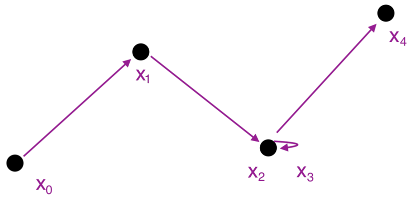
The Metropolis–Hastings algorithm

- Now we answer the main question: How do we sample from a distribution $P(x) \propto e^{-\lambda_2 f(x)}$ given samples from distribution $Q(x) \propto e^{-\lambda_1 f(x)}$ as a warm start?
- We introduce the Metropolis–Hastings algorithm.

The Metropolis–Hastings algorithm

- The Metropolis–Hastings algorithm to sample from a distribution $P(x)$ given samples from distribution $Q(x)$ as a warm start:
- Initially, sample a point x_0 from distribution $Q(x)$.
- At every iteration t, sample $y_{t+1} \sim \mathcal{N}(x_t, \sigma^2 I)$ for some $\sigma > 0$ (to be tuned based on your specific application).
- Define $\alpha = \min\left\{\frac{P(y_{t+1})}{P(x_t)}, 1\right\}$
- Define

$$x_{t+1} = \begin{cases} y_{t+1} & \text{w.p. } \alpha \\ x_t & \text{w.p. } 1 - \alpha. \end{cases}$$



The Metropolis–Hastings algorithm

- Main theorem: When $t \to \infty$, $x_t \to$ a sample according to distribution $P(x)$.
- In other words, the stationary distribution of the Metropolis–Hastings process is $P(x)$.
- Proof: Let $\pi(x)$ be a distribution, let $p(x \mid x')$ be the condition probability (density) of arriving at x by doing one step of the Metropolis–Hastings process starting from x',
- We just need to verify that when $\pi(x) = P(x)$,

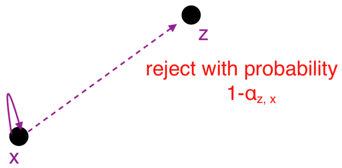
$$\pi(x) = \int_{x'} p(x \mid x')\pi(x')dx'$$

- Consider two cases: $x' = x$, $x' \neq x$.

The Metropolis–Hastings algorithm

- Let $n(x \mid x')$ be the probability density of point x associated with distribution $\mathcal{N}(x', \sigma^2 I)$.
- Define $\alpha_{x,x'} = \min\left\{\frac{P(x)}{P(x')}, 1\right\}$.
- When $x = x'$, we know that

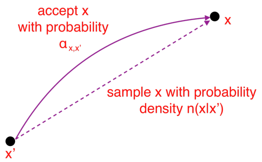
$$\Pr[x' = x \mid x] = \int_z (1 - \alpha_{z,x}) n(z \mid x) dz$$



-

The Metropolis–Hastings algorithm

- Define $\alpha_{x,x'} = \min\left\{\frac{P(x)}{P(x')}, 1\right\}$.
- When $x \neq x'$, we know that $p(x \mid x') = \alpha_{x,x'} n(x \mid x')$.



-
- This implies that when $x' \neq x$:

$$\int_{x'} p(x \mid x')\pi(x')dx' = \int_{x'} \alpha_{x,x'} n(x \mid x')\pi(x')dx'$$

- Together, we have:

$$\int_{x'} p(x \mid x')\pi(x')dx' = \int_{x'} \alpha_{x,x'} n(x \mid x')\pi(x')dx'$$

$$+\pi(x) \int_{z} (1 - \alpha_{z,x}) n(z \mid x) dz$$

The Metropolis–Hastings algorithm

- Together we have:

$$\int_{x'} p(x \mid x')\pi(x')dx' = \int_{x'} n(x \mid x')\alpha_{x,x'}\pi(x')dx'$$

$$+\pi(x)\int_z (1-\alpha_{z,x})n(z \mid x)dz$$

- Key observation:

$$P(x')\alpha_{x,x'} = \min\{P(x), P(x')\} = P(x)\alpha_{x',x}$$

- Which implies that when $\pi(x) = P(x)$,

$$\int_{x'} p(x \mid x')\pi(x')dx' = \int_{x'} n(x \mid x')\alpha_{x',x}\pi(x)dx'$$

$$+\pi(x)\int_z (1-\alpha_{z,x})n(z \mid x)dz$$

The Metropolis–Hastings algorithm

- Now we have: when $\pi(x) = P(x)$

$$\int_{x'} p(x \mid x')\pi(x')dx' = \int_{x'} n(x \mid x')\alpha_{x',x}\pi(x)dx'$$

$$+\pi(x)\int_z (1 - \alpha_{z,x})n(z \mid x)dz$$

- Key observation: $n(x \mid x') = n(x' \mid x)$.
- Therefore, when $\pi(x) = P(x)$:

$$\int_{x'} p(x \mid x')\pi(x')dx' = \int_{x'} n(x' \mid x)\alpha_{x',x}\pi(x)dx'$$

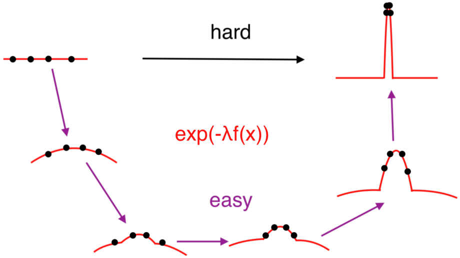
$$+\pi(x)\int_z (1 - \alpha_{z,x})n(z \mid x)dz$$

$$= \pi(x)\int_z n(z \mid x)dz = \pi(x)$$

- We show that $P(x)$ is a stationary distribution.

The Metropolis–Hastings algorithm

- Key idea: using a warm start Q that is close to P, Metropolis–Hastings algorithm converges to the distribution P faster.
- This is also heuristics (but there is a very non-trivial convergence proof when P, Q are log-concave distributions).



-

The Simulate Annealing algorithm

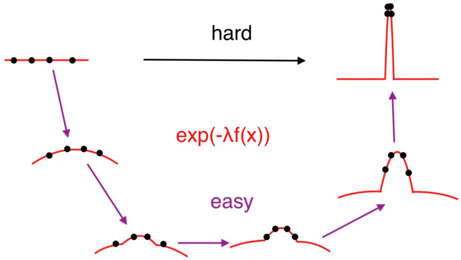
- Simulated annealing algorithm to minimize a function f in a constraint set $\mathcal{D}$ in general:
- First, sample a point x_0 uniformly at random from $\mathcal{D}$, set $\lambda_0 > 0$ to be sufficiently small.
- At every iteration t, let $\lambda_{t+1} = (1+\eta)\lambda_t$.
- Run the Metropolis–Hastings algorithm (for multiple steps) to sample x_{t+1} from distribution

$$p_{\lambda_{t+1}}(x) \propto e^{-\lambda_{t+1} f(x)}$$

starting from x_t as a "warm start".

The Simulate Annealing algorithm

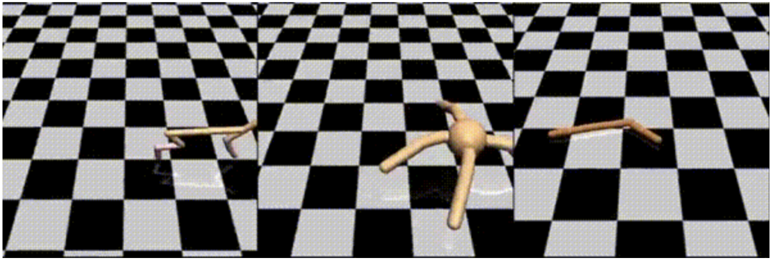
- The spirit of simulated annealing algorithm to minimize a function f:
- Evolve the distribution p from uniform at random to supports only on the minimizers of f.



-
- When we run Metropolis–Hastings algorithm for sufficiently many steps at every iteration (steps $\to +\infty$), simulated annealing converges to the global minimizer of f.
- However, the number of steps might be very large, there is no efficient rate guarantee unless f is convex.

Other evolutionary algorithm

- There is an extremely simple, and powerful alternative evolutionary algorithm used in reinforcement learning, called evolution strategies (ES).
- Example: MuJoCo controls



Other evolutionary algorithm

- Evolutionary strategies to maximize a function $f : \mathbb{R}^d \to \mathbb{R}$
- At every iteration t, randomly sample N i.i.d. vectors $\epsilon_1, \cdots, \epsilon_N \sim \mathcal{N}(0, I_{d\times d})$.
- Compute function value using a standard deviation σ_t: $f_i = f(x_t + \sigma_t \epsilon_i)$ for every $i \in [N]$.
- Update using learning rate η:

$$x_{t+1} = x_t + \eta \frac{1}{N\sigma_t} \sum_{i=1}^{N} f_i \epsilon_i$$

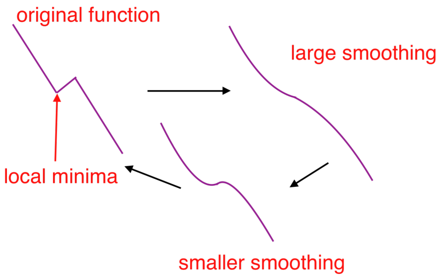
- The "lucky ones" ϵ_i with higher function values f_i gets weighted higher.

ES algorithm

- Actually, the ES algorithm is trying to do stochastic gradient ascent on the new objective
$$F_t = f * g_t$$
- Where g_t is the density function of $\mathcal{N}(0, \sigma_t^2 I)$, $*$ is the convolution operation:
$$[f * g](x) = \int_y f(y) g(y - x) dy$$
- By Stoke's formula,
$$\nabla F_t = f * \nabla g_t$$

ES algorithm

- Actually, the ES algorithm is trying to do stochastic gradient ascent on the new objective
$$F_t = f * g_t$$
- Where g_t is the density function of $\mathcal{N}(0, \sigma_t^2 I)$
- ES can even escape local minima when σ_t is large.



-

Evolutionary algorithms

- There are a lot of other evolutionary algorithms.
- The spirit: Initially, starting from a prior distribution over the set of parameters.
- At every iteration, evolve this distribution a little bit towards higher quality solutions
-