Numerical Optimization 10: Stochastic Methods

Qiang Zhu

University of Nevada Las Vegas

May 20, 2020

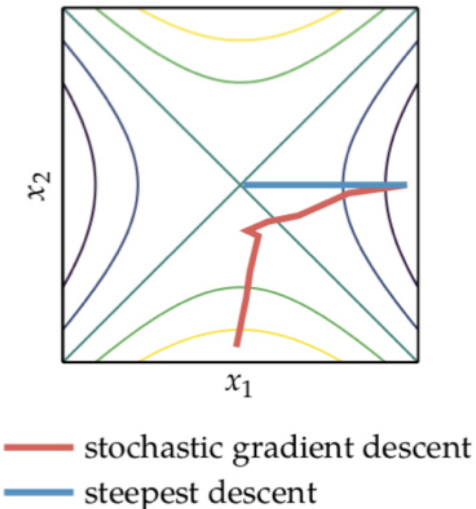
Overview

Noisy Descent

Adding stochasticity to gradient descent can be beneficial in large nonlinear optimization problems. Saddle points, where the gradient is very close to zero, can cause descent methods to select step sizes that are too small to be useful. One approach is to add Gaussian noise at each descent step

$$\boldsymbol{x}^{k+1} \leftarrow \boldsymbol{x}^k - \alpha^k \boldsymbol{g}^k + \boldsymbol{\epsilon}^k$$

where $\boldsymbol{\epsilon}(k)$ is zero-mean Gaussian noise with standard deviation σ. The amount of noise is typically reduced over time. The standard deviation of the noise is typically a decreasing sequence $\sigma(k)$ such as $1/k$.



Simulated Annealing

Simulated annealing borrows inspiration from metallurgy. Temperature is used to control the degree of stochasticity during the randomized search.

- t starts high, allowing the process to freely move , with the hope of finding a good region with the best local minimum.
- t is then slowly brought down, reducing the stochasticity and forcing the search to converge to a minimum. Simulated annealing is often used on functions with many local minima due to its ability to escape local minima.

At every iteration, a candidate transition from $\boldsymbol{x}$ to $\boldsymbol{x}$ is sampled from a transition distribution T and is accepted with probability

$$\begin{cases} 1 & \text{if } \Delta y \leq 0 \\ \min(\exp(-\Delta y/t), 1) & \text{if } \Delta y > 0 \end{cases}$$

where $\Delta y = f(\boldsymbol{x}) - f(\boldsymbol{x}\text{‘})$

Cross-Entropy Method

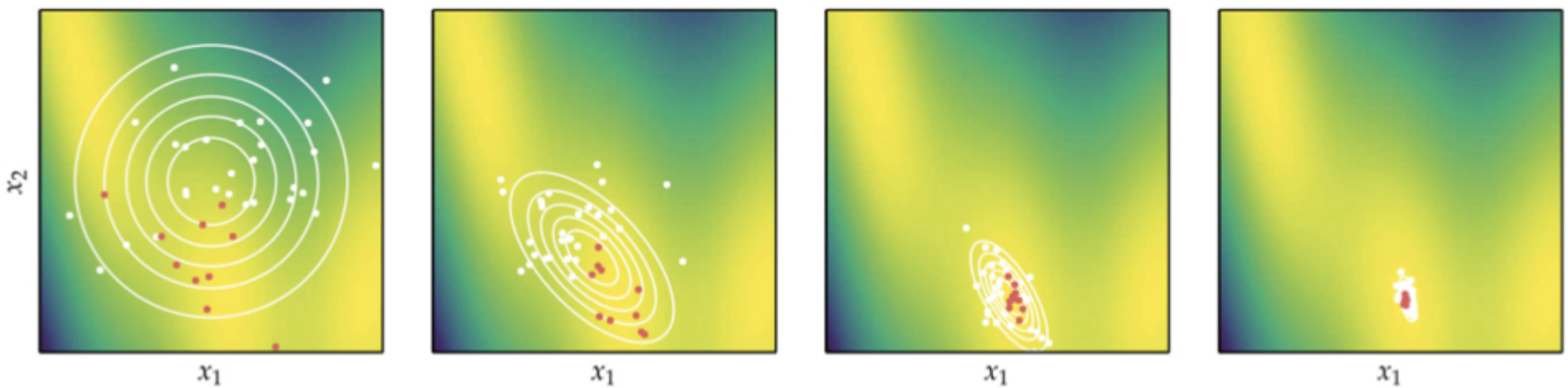
This probability distribution, often called a proposal distribution, is used to propose new samples for the next iteration. At each iteration, we sample from the proposal distribution and then update the proposal distribution to fit a collection of the best samples.
It requires choosing a family of distributions parameterized by θ, such as multivariate normal distributions with a mean vector and a covariance matrix. The algorithm also requires us to specify the number of elite samples, m_{elite} , to use when fitting the parameters for the next iteration.

$$\boldsymbol{\mu}^{k+1} = \frac{1}{m_{\text{elite}}} \sum_{i=1}^{m_{\text{elite}}} \boldsymbol{x}^i$$

$$\Sigma^{k+1} = \frac{1}{m_{\text{elite}}} \sum_{i=1}^{m_{\text{elite}}} (\boldsymbol{x}^i - \boldsymbol{\mu}^{k+1})(\boldsymbol{x}^i - \boldsymbol{\mu}^{k+1})^T$$

Cross-Entropy Method

This probability distribution, often called a proposal distribution, is used to propose new samples for the next iteration. At each iteration, we sample from the proposal distribution and then update the proposal distribution to fit a collection of the best samples.



Covariance Matrix Adaptation

Covariance matrix adaptation maintains a mean vector $\boldsymbol{\mu}$, a covariance matrix $\boldsymbol{\Sigma}$, and an additional step-size scalar δ. The covariance matrix only increases or decreases in a single direction with every iteration, whereas the step-size scalar is adapted to control the overall spread of the distribution. At every iteration, m designs are sampled from the multivariate Gaussian

$$\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{\mu}, \sigma^2 \Sigma)$$

The designs are then sorted according to their objective function values such that $f(x^1) \leq f(x^2) \leq \cdots \leq f(x^m)$. A new mean vector $\boldsymbol{\mu}^{k+1}$ is formed using a weighted average of the sampled designs:

$$\boldsymbol{\mu}^{k+1} \leftarrow \sum_{i=1}^{m} w_i \boldsymbol{x}^i$$

$$\sum_{i}^{m} w_i = 1 \qquad w_1 > w_2 > \cdots > w_m > 0$$

Covariance Matrix Adaptation

The recommended weighting is obtained by

$$w'_i = \ln \frac{m+1}{2} - \ln i \text{ for } i \in \{1, \cdots, m\}$$

to obtain $\boldsymbol{w} = \boldsymbol{w}' / \sum_i w'_i$.
The step size is updated using a cumulative $\boldsymbol{p}_\sigma$ that tracks steps over time

$$\begin{aligned}
\boldsymbol{p}_\sigma^1 &= 0 \\
\boldsymbol{p}_\sigma^{k+1} &\leftarrow (1-c_\sigma)\boldsymbol{p}_\sigma + \sqrt{c_\sigma(2-c_\sigma)\mu_{\text{eff}}}(\Sigma^k)^{-1/2}\sigma_w \\
\mu_{\text{eff}} &= \frac{1}{\sum_i w_i^2} \\
\sigma_w &= \sum_{i=1}^{m_{\text{elite}}} w_i \sigma^i \text{ for } \sigma^i = \frac{\boldsymbol{x}^i - \boldsymbol{\mu}^k}{\sigma^k}
\end{aligned}$$

Covariance Matrix Adaptation

The new step size is

$$\sigma^{k+1} \leftarrow \sigma^k \exp\left(\frac{c_\sigma}{d_\sigma}\left[\frac{||\boldsymbol{p}_\sigma||}{\mathbb{E}||\mathcal{N}(0,\boldsymbol{I})||} - 1\right]\right)$$

where $\mathbb{E}$ is the expected length of a vector drawn from Gaussian distribution.

$$\mathbb{E}||\mathcal{N}(0,\boldsymbol{I})|| = \sqrt{2}\frac{\Gamma(\frac{n+1}{2})}{\Gamma(\frac{n}{2})} \approx \sqrt{n}\left(1 - \frac{1}{4n} + \frac{1}{21n^2}\right)$$

$$c_\sigma = (\mu_{\text{eff}} + 2)/(n + \mu_{\text{eff}} + 5)$$
$$d_\sigma = 1 + 2\,\text{max}(0, \sqrt{\mu_{\text{eff}} - 1)/(n+1)} - 1) + c_\sigma$$

Covariance Matrix Adaptation

The covariance matrix is updated as follows

$$\boldsymbol{p}_{\Sigma}^{1} = 0$$
$$\boldsymbol{p}_{\Sigma}^{k+1} \leftarrow (1 - c\Sigma)\boldsymbol{p}_{\boldsymbol{\Sigma}}^{\boldsymbol{k}} + h_{\sigma}\sqrt{c\Sigma(2 - c_{\Sigma})\mu_{\mathrm{eff}}}\boldsymbol{\sigma}_{w}$$

where

$$h_{\sigma} = \begin{cases} 1 & if \frac{||\boldsymbol{p}_{\Sigma}||}{(1-c_{\sigma}^{2k+1})} < (1.4 + \frac{2}{n+1})\mathbb{E}||\mathcal{N}(0, \boldsymbol{I})|| \\ 0 & \text{otherwise} \end{cases}$$

The update requires the adjusted weights $\boldsymbol{w}$:

$$w_{i}^{0} = \begin{cases} w_{i} & \text{if } w_{i} \geq 0 \\ \frac{nw_{i}}{||\Sigma^{-1/2}\boldsymbol{\delta}^{i}||^{2}} & \text{otherwise} \end{cases}$$

Covariance Matrix Adaptation

The The covariance update is then

$$\Sigma^{k+1} \leftarrow [1 + c_1 c_\sigma (1 - h_\sigma)(2 - c_\sigma) - c_1 - c_\mu]\Sigma^k + c_1 \boldsymbol{p}_\Sigma \boldsymbol{p}_\Sigma^T + c_\mu \sum_{i=1}^{\mu} w_i^0 \boldsymbol{\delta}^i (\boldsymbol{\delta}^i)^T$$

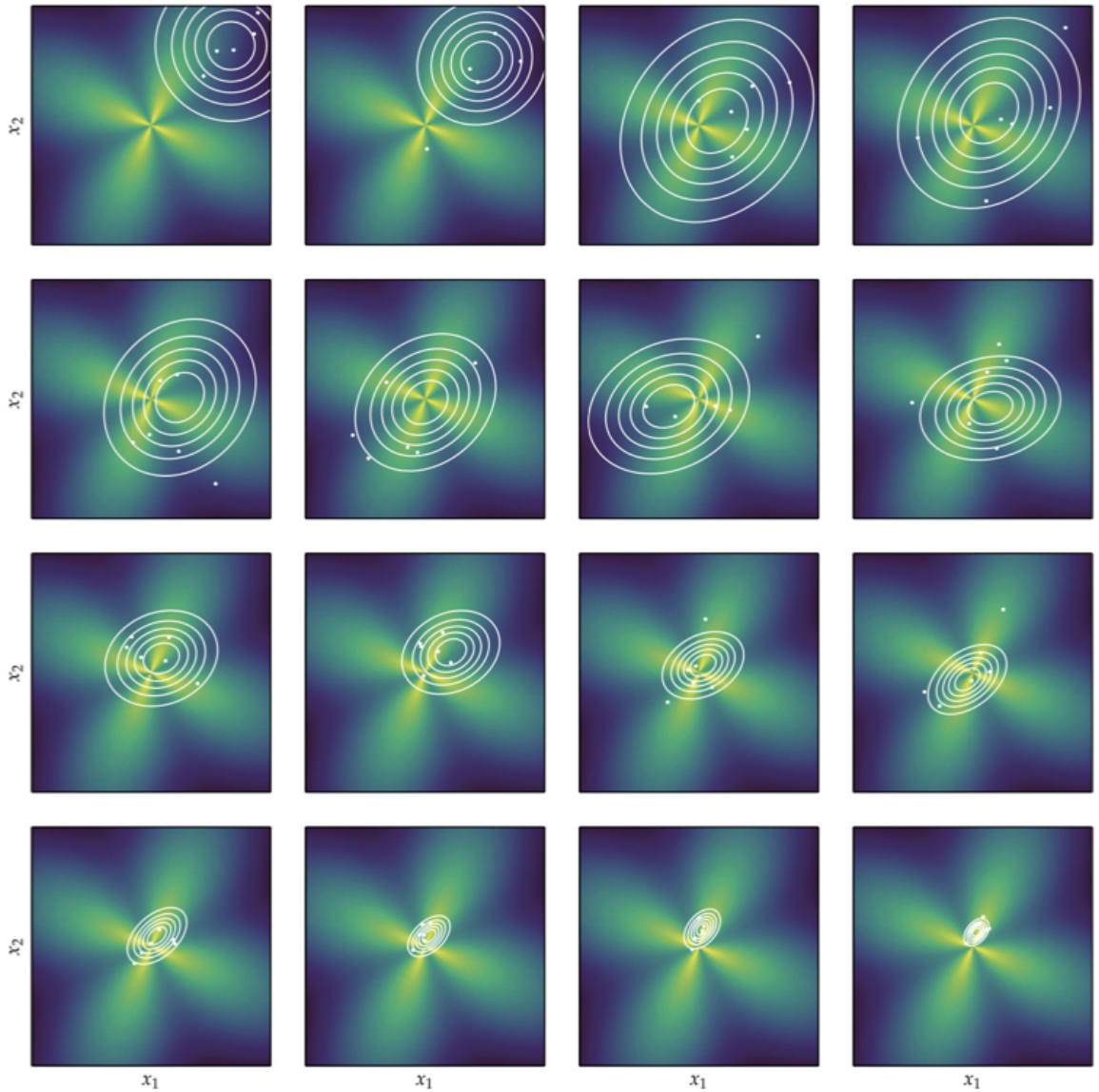
The constants have the following recommended values

$$c_\Sigma = \frac{4 + \mu_{\text{eff}}/n}{n + 4 + 2\mu_{\text{eff}}/n}$$

$$c_1 = \frac{2}{(n + 1.3)^2 + \mu_{\text{eff}}}$$

$$c_\mu = \min\left(1 - c_1, 2\frac{\mu_{\text{eff}} - 2 + 1/\mu_{\text{eff}}}{(n + 2)^2 + \mu_{\text{eff}}}\right)$$

Covariance Matrix Adaptation



22

Summary

- Stochastic methods employ random numbers during the optimization process
- Simulated annealin guses a temperature that controls random exploration and which is reduced over time to converge on a local minimum.
- The cross-entropy method and evolution strategies maintain proposal distributions from which they sample in order to inform updates.
- Covariance matrix adaptation is a robust and sample-efficient optimizer that maintains a multivariate Gaussian proposal distribution with a full covariance matrix.