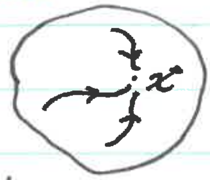
Chapter 3: Stochastic approximations

or stochastic iterative algorithms

3.1. Introduction

- x^* = solution to some problem
- Examples:
  - Solve $f(x) = a$
  - $\min f(x)$ — optimization
  - $A v_i = \lambda_i v_i$ — Eigenvalues/vectors
  - RL: find policy optimizing reward
- Goal: Calculate/find/estimate x^*
- Approach 1: Deterministic recursion iteration ~ iteration

$$x_{n+1} = \underbrace{T(x_n)}_{\text{map, transfer fct}} = x_n + F(x_n)$$



  - $x_n \xrightarrow{n\to\infty} x^*$ from x_1
  - x^* attracting fixed pt of T

- Approach 2: Stochastic recursion iteration

$$X_{n+1} = X_n + a_n Y_n$$

Y_n function of X_n related to X_n

$$\text{or} \quad = X_n + a_n \underbrace{F(X_n, C_n)}_{Y_n}$$

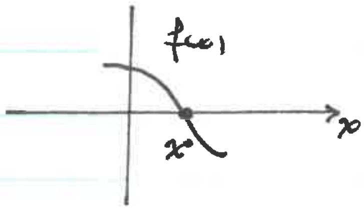
$P(Y_n | X_n)$

$$\text{or} \quad = X_n + a_n \underbrace{H(X_n)}_{\text{deterministic map}} + b_n \underbrace{M_n}_{\text{noise}}$$

  - Find $a_n, b_n \downarrow 0$ as $n \to \infty$ such that $\underbrace{X_n}_{RV} \xrightarrow{n\to\infty} \underbrace{x^*}_{\text{value/constant}}$ in probability
  - $\lim_{n\to\infty} P(|X_n - x^*| > \varepsilon) = 0$
  - Stochastic recursion = $\underbrace{\text{non-homogeneous}}_{\text{why?}}$ $\underbrace{\text{Markov chain}}_{\text{why?}}$

- Applications:
  - Stochastic gradient descent
  - Stochastic annealing
  - Reinforcement learning
  - etc.

3.2. Finding zeros



- $f : \mathbb{R}^d \to \mathbb{R}^d$
- Solve $f(x) = a$
    - Take $a = 0$ w/o loss generality
    - Can have one, no, or multiple solutions
    - Solution: $x^* \ni f(x^*) = 0$

3.2.1 Deterministic recursion

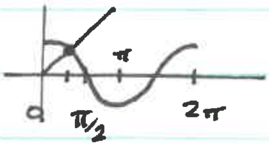
$$\begin{cases} x_{n+1} = T(x_n) & \text{iteration / map / transfer function} \\ x_1 = x & \text{initial value} \end{cases}$$

- Fixed-point condition: $T(x^*) = x^*$ norm
- Contraction condition: $\| \underbrace{DT}_{\text{Jacobian}}(x^*) \| < 1$
- In $\mathbb{R}$: $|T'(x^*)| < 1$
- Convergence: $x_n \to x^*$ as $n \to \infty$

Example: Newton-Raphson map: $T(x) = x - \dfrac{f(x)}{f'(x)}$

$$T(x^*) = x^* - \frac{f(x^*)^{=0}}{f'(x^*)} = x^*$$

Example: $\cos(x) = x$



$\Rightarrow f(x) = \cos(x) - x = 0$

$T_1(x) = \cos(x)$

$T_2(x) = x - \dfrac{f(x)}{f'(x)} = x + \dfrac{\cos(x) - x}{\sin(x) + 1}$ N-R See demo

- Iteration: $x_1 \to x_2 \to \cdots \to x_n$ deterministic from $x_1 = x$

- Stop when $|x_n - x^*| < \varepsilon$
- In practice, stop when $|x_n - x_{n-1}| < \varepsilon$ no improvement
- If many solutions: try different initial point

- Rem: $x_{n+1} = T(x_n) = x_n + F(x_n)$, $F(x^*) = 0$

3.2.2 Stochastic recursion

Ref: Robbins - Monro 1951

- Solve $f(x) = \alpha$
- $f(x)$ not known / given exactly
- Estimate : Y

$$E[Y|x] = \sum_y y\, P(y|x) = f(x)$$

$x \to \boxed{f} \to f(x)$

$x \to \boxed{} \to Y$

random variable

  - Unbiased estimate of $f(x)$
  - Unbiased function call

- Example: Additive noise model $\quad Y = f(x) + \xi \qquad E[\xi] = 0$

$$E[Y|x] = f(x) + E[\xi] = f(x)$$

- Recursion : $\quad X_{n+1} = X_n - a_n (Y_n - \alpha) \qquad X_1 = x_1$

initial value

  - Y_n : fct call for X_n = estimate of $f(X_n)$
  - a_n : annealing sequence

$$a_n \searrow 0 \quad \text{as} \quad n \to \infty$$

$$\sum_{n=1}^{\infty} a_n = \infty \qquad \sum_{n=1}^{\infty} a_n^2 < \infty$$

  - Convergence : $X_n \xrightarrow{n\to\infty} x^*$ in probability

$$\lim_{n\to\infty} P(|X_n - x^*| > \varepsilon) = 0$$

  - In practice : $a_n = \frac{1}{n}$

- Iteration : $\quad X_1 \to X_2 \to \cdots \to X_n$
  - Markov chain (non-homogeneous)
  - Probably almost correct (PAC) : Stop at X_{n_0} $\ni$

$$P(|X_n - x^*| \geq \varepsilon) < \delta \qquad \forall n \geq n_0$$

  - In practice : Stop when $|X_n - X_{n-1}| < \varepsilon$

no improvement in solution

· Example: $f(x) = \cos(x) + x \quad \rightarrow \quad Y = \cos(x) + x + \underbrace{\mathcal{U}[-1, 1]}_{\text{additive noise}}$

See demo

· Example: Mean estimator

$$S_n = \frac{1}{n} \sum_{i=1}^{n} X_i \qquad X_i \sim p$$

$$S_{n+1} = \frac{n}{n+1} S_n + \frac{X_{n+1}}{n+1}$$

$$= S_n - \frac{1}{n+1}\left(S_n - X_{n+1}\right)$$

$$= S_n - a_n Y_n$$

$$E[Y_n \mid S_n = s] = E[S_n - X_n \mid S_n = s]$$
$$= s - E[X_{n+1}]$$
$$= s - \mu = f(s)$$

$f(s) = 0 \Rightarrow s^* = \mu \Rightarrow S_n \rightarrow \mu$ in probability as $n \rightarrow \infty$ (LLN)

· Rem: Linear + additive noise models

$$X_{n+1} = X_n - a_n (\underbrace{X_n + Z_n}_{Y_n})$$

noise

noisy version of $f(x) = x$

$$= \underbrace{(1 - a_n)}_{\nearrow 1} X_n - \underbrace{a_n}_{\searrow 0} Z_n$$

reinforce update
exploitation

random update / change
exploration

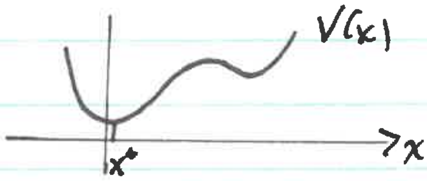
· Rem:

Before	Now
Y = noisy obs. of $f(x)$	Put noise in $f(x)$
inherent noise	explicit noise
noise is bad	noise is good ← exploration

3.3. Optimization

- Potential / cost / loss : $V : \mathcal{X} \to \mathbb{R}$
- State / solution space : $\mathcal{X} = \mathbb{R}^d$ or discrete space ✓ See Sec. 3.4
- Minimization problem :

$$\min_{x \in D} V(x)$$



  - $D \subseteq \mathcal{X}$ constraint set
  - Assume unique solution (for now) : $x^* = \operatorname{argmin}_{x \in D} V(x)$

- Examples : · MLE
  · Neural net training

- Rem : $\mathcal{X} = \mathbb{R}^d$
  - Gradient : $\nabla V(x) = \begin{pmatrix} \frac{\partial}{\partial x_1} V(x_1, \dots x_n) \\ \vdots \\ \frac{\partial}{\partial x_n} V(x_1 \dots x_n) \end{pmatrix} = \begin{pmatrix} \partial_1 V \\ \vdots \\ \partial_n V \end{pmatrix}$
  - Hessian matrix : $\operatorname{Hess} V(x) = \dfrac{\partial^2 V(x)}{\partial x_i \, \partial x_j}$
  - Critical point : $\nabla V(x^*) = 0$ n equations of n variables
    - Min (local or global) : $\operatorname{Hess} V(x)$ positive definite
    - Max (" " ") : $\operatorname{Hess} V(x)$ negative def
    - Saddle point : positive and negative eigenvalues
  - In $\mathbb{R}^1$: $V'(x^*) = 0$
    - Min : $V''(x^*) > 0$
    - Max : $V''(x^*) < 0$
  - $\nabla V(x)$ = direction of greatest ~~descent~~ ascent at x

    $V'_{\vec{d}}(x) = \nabla V(x) \cdot \vec{d}$

    $\| V'_{\vec{d}}(x) \| = \| \nabla V(x) \| \, \| \vec{d} \| \cos \theta$

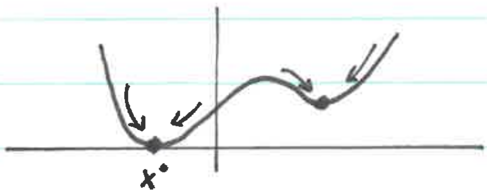
    max at $\theta = 0$ i.e. $\vec{d} \parallel \nabla V$

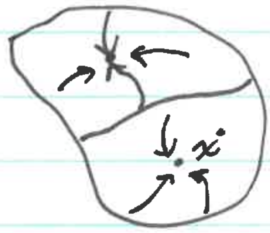
3.3.1 Gradient descent

- Iteration: $x_{n+1} = x_n - \gamma \nabla V(x_n)$

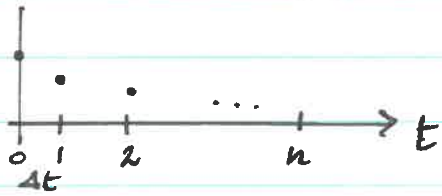
  $x_1 = x$ initial value / seed

  - Fixed point: $\nabla V(x^*) = 0$
  - Convergence: $x_n \to x^*$ if x_1 in basin of attraction of x^*
  - $V(x)$ can have multiple global/local min
  - Convergence to global min not guaranteed





- Continuous-time limit:



$$x_{n+1} = x_n - \gamma \nabla V(x_n) \qquad x_0 \text{ initial value}$$

$$x(t+\Delta t) = x(t) - \tilde{\gamma} \nabla V(x(t))\, \Delta t \qquad \gamma = \tilde{\gamma}\Delta t$$

$$\frac{x(t+\Delta t) - x(t)}{\Delta t} = -\tilde{\gamma} \nabla V(x(t))$$

$$\Rightarrow \quad \dot{x}(t) = \frac{d}{dt} x(t) = -\tilde{\gamma} \nabla V(x(t)) \qquad \text{Gradient ODE}$$

$x(t) \longrightarrow x^*$ as $t \to \infty$ if $x(0)$ in basin of attraction of x^*

- Example: $V(x) = \frac{x^2}{2}$ $\quad x^* = 0$

$$\dot{x}(t) = -\gamma V'(x(t)) = -\gamma x(t) \quad \to \quad x(t) = x(0)\, e^{-\gamma t}$$

Converges exponentially to $x^* = 0$ from any $x(0)$

· Other gradient dynamics

1- Newton - Raphson :

$$V'(x^*) = f(x^*) = 0$$

$$x_{n+1} = T(x_n) \qquad T(x) = x - \frac{f(x)}{f'(x)}$$

$$= x_n - \frac{V'(x_n)}{V''(x_n)}$$

$$= x_n - \gamma_n V'(x_n) \qquad \gamma_n = V''(x_n)^{-1}$$

adjusted learning rate

2- Gradient with momentum :

$$p_{i+1} = -\varepsilon \nabla V(x_i) + (1-\varepsilon\gamma) p_i$$

$$x_{i+1} = x_i + \varepsilon p_{i+1}$$

· Heavy ball descent with "oscillations"

· Continuous limit :

$$\dot{p}(t) = -\underbrace{a\, p(t)}_{\text{friction force}} - \underbrace{b \nabla V(x(t))}_{\text{potential force}}$$

$$\dot{x}(t) = c\, p(t) \qquad \text{momentum}$$

· Comes from Newton's law : $F = ma = m\ddot{x}$

$$\Rightarrow \underbrace{\ddot{x}(t) = F/m}_{2^{nd}\text{ order}} \rightarrow \begin{cases} \dot{x} = v \\ \dot{v} = \ddot{x} = F/m \end{cases} \qquad \text{2 first-order eqs}$$

· Adam (adagrad) :

$$m_{i+1} = \beta_1 m_i + (1-\beta_1)\nabla V(\theta_i)$$

$$v_{i+1} = \beta_2 v_i + (1-\beta_2)\nabla V(\theta_i)^2$$

$$\hat{m}_{i+1} = \frac{m_{i+1}}{1-\beta_1^{i+1}} \qquad \hat{v}_{i+1} = \frac{v_{i+1}}{1-\beta_2^{i+1}}$$

$$\theta_{i+1} = \theta_i - \alpha \frac{\hat{m}_{i+1}}{\sqrt{\hat{v}_{i+1}} + \varepsilon}$$

2 momenta reinforced

3.3.2 Stochastic gradient descent (SGD)

$$X_{n+1} = X_n - \underbrace{a_n}_{\text{learning rate}} G_n \qquad \underbrace{X_1 = x}_{\text{initial value}}$$

- Gradient estimate:

$$E[G|x] = \sum_g g\, P(g|x) = \nabla V(x)$$

$$E[G_n|x] = E[G_n | X_n = x] = \nabla V(x)$$

- Example: Additive noise: $G_n = \nabla V(X_n) + \underbrace{\xi_n}_{\text{noise}}$ $\qquad E[\xi_n] = 0$

$$E[G_n | X_n = x] = \nabla V(x) + E[\xi_n] = \nabla V(x)$$

- Convergence conditions (sufficient):
  - $\sum_{n=1}^{\infty} a_n = \infty \qquad \sum_{n=1}^{\infty} a_n^2 < \infty$
  - $X_n \to x^*$ as $n \to \infty$ in probability
  - $\lim_{n\to\infty} P(|X_n - x^*| > \varepsilon) = 0$ $\qquad X_n$: Markov chain non-homogeneous
  - Example: $a_n = 1/n$

- Example: Kiefer-Wolfowitz 1952

$$X_{n+1} = X_n - a_n \underbrace{\left(\frac{Y_n^+ - Y_n^-}{c_n} \right)}_{\text{discrete derivative}}$$

$Y_n^+ \sim P(\cdot | X_n + c_n)$ — estimate of $V(x+\Delta x)$ $\qquad E[Y|x] = V(x)$

$Y_n^- \sim P(\cdot | X_n - c_n)$ — estimate of $V(x - \Delta x)$

$$\sum a_n = \infty \qquad \sum a_n c_n < \infty \qquad \sum a_n^2 c_n^{-2} < \infty$$

- In practice: $c_n = \Delta x$ fixed

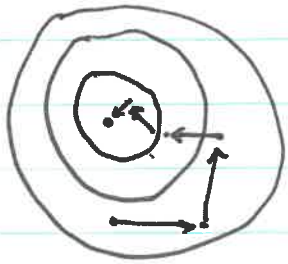
$Y_n^+ \sim P(\cdot | X_n + \Delta x)$

$Y_n^- \sim P(\cdot | X_n - \Delta x)$

$\sum a_n = \infty \qquad \sum a_n^2 < \infty$

- Noisy estimation of $V(x)$
- ∇V by discrete derivative

· Example : Random descent



$$\nabla V = \begin{pmatrix} \partial_1 V \\ \partial_2 V \\ \vdots \\ \partial_n V \end{pmatrix}$$

direction of largest ~~descent~~ ascent

· Random directional derivative : $G = \nabla_{\vec{d}} V = \nabla V \cdot \vec{d}$

$$E[G|x] = \nabla V$$ unbiased gradient

· In $\mathbb{R}^2$: $\nabla_x V = \partial_1 V$, $\nabla_y V = \partial_2 V$, $\vec{d} \sim \mathcal{U}\left\{ \begin{pmatrix}1\\0\end{pmatrix}, \begin{pmatrix}0\\1\end{pmatrix} \right\}$

$$E[G|x] = \sum_i p_i \, \partial_i V = \frac{1}{2} \nabla V(x) \propto \nabla V(x)$$

· In $\mathbb{R}^d$:
- Uniform directions
- Any uniform subsets of $m \leq d$ directions
- Unbiased over directions = no preferred direction

· Example : Drop out
- Neural network (model) with d parameters
- Loss : $L(\theta)$
- Choose subset of parameters randomly $\{\theta\}_{chosen}$ — same as dropping parameters
- Gradient estimate :

$$\nabla_\theta L \approx \nabla_{\{\theta\}_{chosen}} L$$

· Unbiased : $E\left[\nabla_{\{\theta\}_{chosen}} L \right] = \nabla_\theta L$

· Rem : Mini batch optimization

$$L(\theta) = \frac{1}{n} \sum_{i=1}^{n} C_i(\theta) \overset{e.g.}{=} \frac{1}{n} \sum_{i=1}^{n} | y_i - f(x_i, \theta) |^2$$

$$\nabla_{\theta_i} L(\theta) \approx \frac{1}{m} \sum_{j=1}^{m} \nabla_{\theta_i} C_j(\theta)$$

estimate loss on random subset of data

$m \leq n$

3.3.3 Langevin dynamics

· Gradient dynamics:

$$\dot{x}(t) = -\gamma \nabla V(x(t)), \qquad x(0) = x$$

initial value

· SGD: $\underbrace{\dot{X}(t) = -\gamma \nabla V(X(t)) + \sigma \underbrace{\xi(t)}_{\text{noise}}}_{\text{noisy gradient}}$

noise amplitude

· Stochastic differential equation (SDE):

$$dX(t) = -\gamma \nabla V(X(t))\, dt + \sigma\, dW(t)$$

$$\dot{X}(t) = \frac{X(t+dt) - X(t)}{dt} = -\gamma \nabla V(X(t)) + \sigma \xi(t)$$

$$X(t+dt) - X(t) = -\gamma \nabla V(X(t))dt + \sigma \underbrace{\xi(t)dt}_{dW(t)}$$

$$\Rightarrow \quad X(t+dt) = X(t) - \gamma \nabla V(X(t))dt + \sigma \Delta W(t)$$

· Gaussian white noise: $\Delta W(t) \sim \mathcal{N}(0, \Delta t)$

$$= \sqrt{\Delta t}\, Z, \qquad Z \sim \mathcal{N}(0,1)$$

$$\Rightarrow \quad X_{n+1} = X_n - \gamma \nabla V(X_n)\Delta t + \sigma \sqrt{\Delta t}\, Z$$

Euler–Maruyama Scheme

See CW3

· Stationary distribution:

$$p(x,t) \longrightarrow p^*(x) = \frac{e^{-2\gamma V(x)/\sigma^2}}{Z}$$

Gibbs density

$$= \frac{e^{-\beta V(x)}}{Z} \qquad \beta = \frac{2\gamma}{\sigma^2}$$

· Annealing: Decrease σ in time:

$$dX(t) = -\nabla V(X(t))\, dt + \sigma_t\, dW(t)$$

$$\sigma_t \searrow 0 \quad \text{as} \quad t \to \infty$$

See simulated annealing

See CW3

3.4 Simulated annealing

· Potential / cost / loss : $V: \mathcal{X} \to \mathbb{R}$

· Minimization problem : $\min\limits_{x \in D} V(x)$

- ∇V exists $\to$ use GD or SGD
- ∇V doesn't exist (e.g. $\mathcal{X}$ discrete) ?

· Rem :
- Exhaustive search : $O(|\mathcal{X}|)$
- Random search : Needs structure / guiding
- Use $V(x)$ in search : Explore : $x \to x'$
  Reinforce : Accept if $\Delta V < 0$

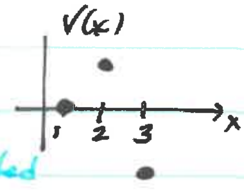
· Gibbs distribution : $P_T(x) = \dfrac{e^{-V(x)/T}}{Z_T}$

- Partition function / normalization : $Z_T = \sum\limits_x e^{-V(x)/T}$
  or $= \int dx\, e^{-V(x)/T}$
- T = temperature (usually >0)
- Inverse temperature : $\beta = T^{-1}$ $\quad T \to 0^+ \quad \beta \to \infty$

$$P_\beta(x) = \frac{e^{-\beta V(x)}}{Z_\beta} \qquad Z_\beta = \sum_x e^{-\beta V(x)}$$

Pincus 1970 · Laplace principle : $P_T(\cdot)$ concentrates on $\min V(x)$ as $T \to 0$.

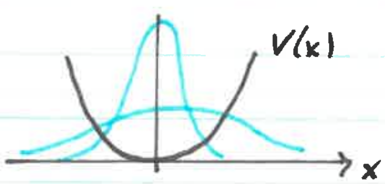
· Example : $\mathcal{X} = \{1, 2, 3\}$ $\quad V(1) = 0,\ V(2) = 1,\ V(3) = -1$



$P_T \to (0 \quad 0 \quad 1) \qquad T \to 0 \qquad (\beta \to \infty)$ peaked

$P_T \to (\tfrac{1}{3} \quad \tfrac{1}{3} \quad \tfrac{1}{3}) \qquad T \to \infty \qquad (\beta \to 0)$ uniform

· Example : $\mathcal{X} = \mathbb{R}$ $\quad V(x) = \dfrac{x^2}{2}$ $\qquad P_T(x) = \dfrac{e^{-x^2/2T}}{\sqrt{2\pi T}}$



$\mathrm{Var}(X) = T \downarrow 0$ as $T \to 0$

Ref: Kirkpatrick et al 1983

· Simulated annealing (SA) algorithm:

- Sample p_T with low $T \to$ samples around x^*
- Metropolis-Hastings with p_T
- Time-dependent $T \to T_n$
- Steps:

1- $X_1 = x$, $T_1 = T$ initial value

2- Proposal: $X \to x'$

3- Accept with prob

$$\rho = \min\left\{1, \frac{p_{T_1}(x')}{p_{T_1}(x)}\right\} = e^{-\Delta V/T_1}$$

Metropolis or MH
symmetric non-symmetric

if $u[0,1] < \rho$:
  $X_2 = x'$
else
  $X_2 = x$

4- Annealing: $T_1 \to T_2$ decrease

5- Repeat

· Rem:
- Time-dependent Markov chain
- Semi-greedy: $\Delta V \le 0$ x' accepted for sure (exploit)
  $\Delta V > 0$ x' " with prob. ρ (explore)

· Annealing schedule: $T_n \downarrow 0$ as $n \to \infty$

- Decrease too fast: $X_n \not\to x^*$ get stuck in local min
- " " slow: X_n too noisy
- Log schedule: $T_n = \dfrac{T_1}{\log n + 1}$ can be slow
- Geometric schedule: $T_n = \dfrac{T_1}{k^n}$ $k > 1$
  no convergence guarantee
  Simulated quench

$\beta_n = \beta_1(\log n + 1)$

$\beta_n = \beta_1 k^n$

See CW3

3.5 Remarks on optimization

- Potential: $V: \mathcal{X} \to \mathbb{R}$ $V(x)$
- Minimization: $\min_{x \in D} V(x)$

$\mathcal{X}$
- Continuous
  - ∇V exists → GD or SGD
  - ∇V doesn't exist
- discrete
  - low dim → exhaustive search
  - high dim → SA

- Ultimate goal: cost(optimization) ~ cost (simulation)
- Why use randomness / noise in optimization?
  - ← $V(x)$ rugged, many mins
  - $V(x)$ non convex

- Neural network training: $L(D, \theta)$
  data, parameters
  - D only a sample of "true" underlying distribution
  - Compute $\nabla_\theta L$ over subset of parameters   drop out
  - Estimate L on subset of D mini batch

  → Estimation of "true" L, $\nabla_\theta L$
  "Noisy" estimate of L, $\nabla_\theta L$

- Rem: Why not training by solving $\nabla_\theta L = 0$?
  " " " using MCMC / SA ?