EE270
Large scale matrix computation, optimization and learning

Instructor : Mert Pilanci

Stanford University

Randomized Linear Algebra and Optimization
Lecture 12: Gradient Descent

Summary of randomized least squares solvers

$Sx : O(n \log n)$ $\quad$ $S = PHD$ (FJLT or SRHT) subsampled

$SA = [SA^{(1)} \cdots SA^{(d)}]$ $\quad$ $O(d \cdot n \log n)$ (near-linear time)

$A = U\Sigma V^T$

AMM for $U^T S^T S U \approx U^T U$

- Left Sketch

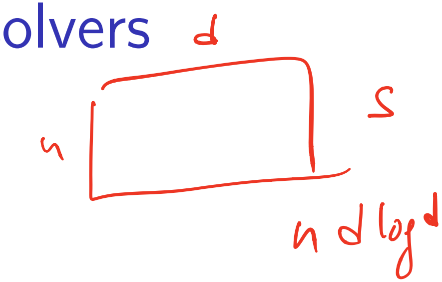
$$\min_x \|Ax - b\|_2^2$$

- $\min_x \|S(Ax - b)\|_2^2$
- Fast Johnson Lindenstrauss Transform (Randomized Hadamard Transform) / Sparse JL:
  SA and Sb can be computed in $O(ndlogn)$ time $\quad$ $O(nd \log n)$
- Gaussian sketch / ± 1
  SA and Sb can be computed in $O(ndm)$ time $= O(nd^2)$
- total complexity:

$m > d$ $\quad$ pick $m = \frac{d}{\epsilon^2}$ $\Rightarrow$ we'll have ϵ-approximate LS solution

$$\boxed{\frac{d^3}{\epsilon^2} + nd \log n}$$ compare with nd^2

Summary of randomized least squares solvers



- Right Sketch

$$\min_{Ax=b} \|x\|_2^2$$

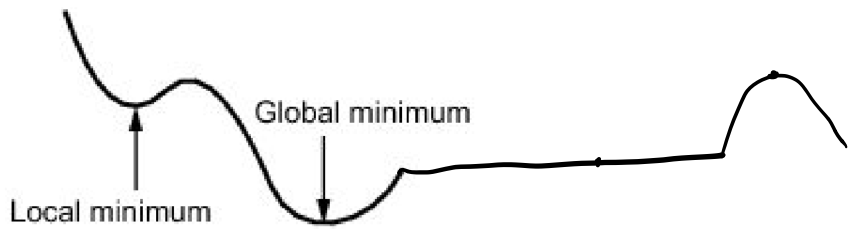
- $\min_{ASz=b} \|z\|_2^2$
- Fast Johnson Lindenstrauss Transform (Randomized Hadamard Transform)

  AS can be computed in $O(ndlogd)$ time
- Gaussian sketch

  AS can be computed in $O(ndm)$ time
- total complexity: $nd\log d + \frac{n^3}{\epsilon^2}$

$m = \frac{n}{\epsilon^2}$.

Optimization: Gradient Descent



- Consider unconstrained minimization of $f : \mathbb{R}^d \to \mathbb{R}$, differentiable function
- we want to solve

μ_t : step size at iteration t

$$\min_{x \in \mathbb{R}^d} f(x)$$

- **Gradient descent:** choose initial $x_0 \in \mathbb{R}^d$ and repeat

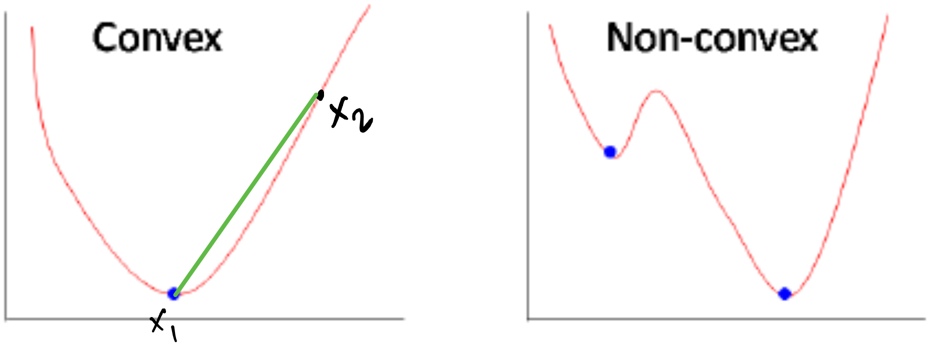
$$x_{t+1} = x_t - \mu_t \nabla f(x_t)$$

- for $t = 1, \ldots, T$

Convex vs Non-convex functions

- a function f is called **convex** if

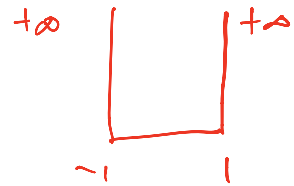
$$\forall x_1, x_2 \in \mathcal{X},\ \forall t \in [0,1]: \quad f(tx_1 + (1-t)x_2) \leq tf(x_1) + (1-t)f(x_2)$$

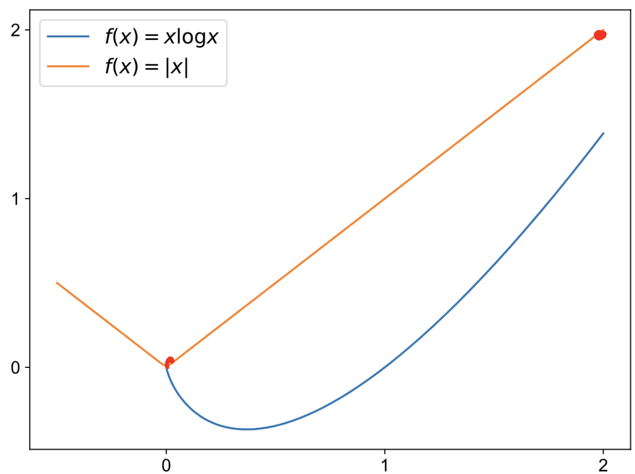


Convex vs Non-convex functions

- a function f is called **strictly convex** if

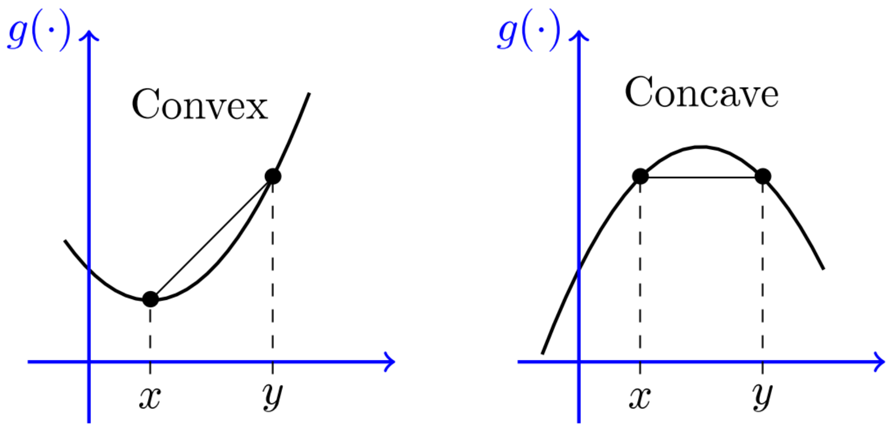
$$\forall x_1 \neq x_2 \in \mathcal{X}, \forall t \in [0,1] : \quad f(tx_1 + (1-t)x_2) < tf(x_1) + (1-t)f(x_2)$$





Concave functions

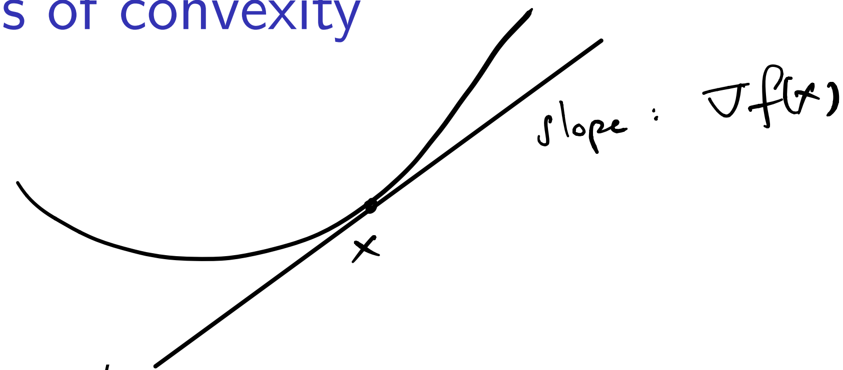
- a function f is called (strictly) **concave** if $-f$ is (strictly) convex



Differentiable functions

- A one dimensional function $f : \mathbb{R} \to \mathbb{R}$ is differentiable if the derivative
  $f'(x) := \lim_{h \to 0} \frac{f(x+h)-f(x)}{h}$ exists
- Suppose that all partial derivatives of $f : \mathbb{R}^d \to \mathbb{R}$ exists
  The gradient $\nabla f(x)$ is the vector of partial derivatives
  $[\nabla f(x)]_i = \frac{\partial}{\partial x_i} f(x)$

Alternative definitions of convexity



- Assume that $f(x) : \mathbb{R}^d \to \mathbb{R}$ is differentiable. Then f is convex, if and only if for every x, y the inequality

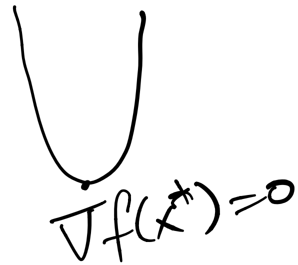
$$f(y) \geq \underbrace{f(x) + \nabla f(x)^T (y - x)}_{\text{first order Taylor approx at } x}$$

is satisfied

Twice differentiable functions

- Suppose that all second derivatives of $f : \mathbb{R}^d \to \mathbb{R}$
  $\frac{\partial}{\partial x_i} \frac{\partial}{\partial x_j} f(x)$ exists
  The Hessian $\nabla^2 f(x)$ is the matrix of partial derivatives
  $[\nabla^2 f(x)]_{ij} = \frac{\partial}{\partial x_i} \frac{\partial}{\partial x_j} f(x)$

Twice differentiable convex functions



- A twice differentiable function $f(x)$ is convex if and only if the Hessian $\nabla^2 f(x)$ is positive semi-definite for all $x \in \mathbb{R}^d$
- Suppose that f is convex and differentiable, then x^* is a global minimizer of f if and only if $\nabla f(x^*) = 0$

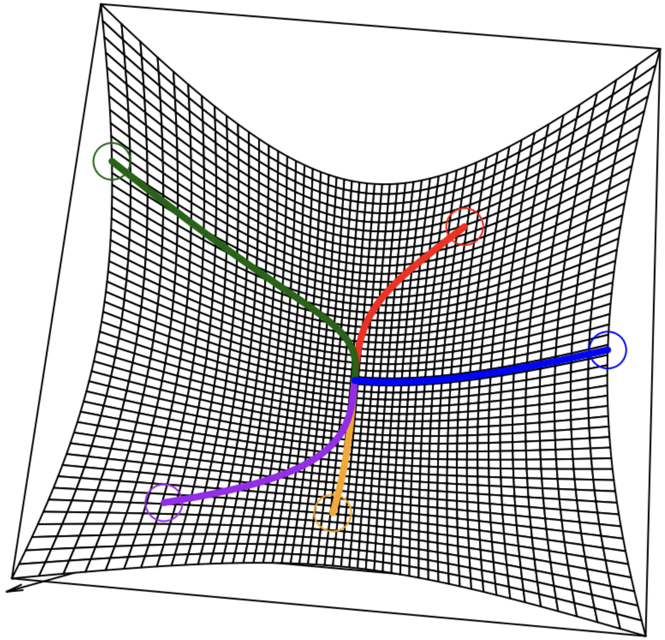
Gradient descent for differentiable functions

- $-\nabla f(x)$ is the direction of largest instantaneous decrease
- Gradient Descent (GD):

$$x_{t+1} = x_t - \mu_t \nabla f(x_t)$$

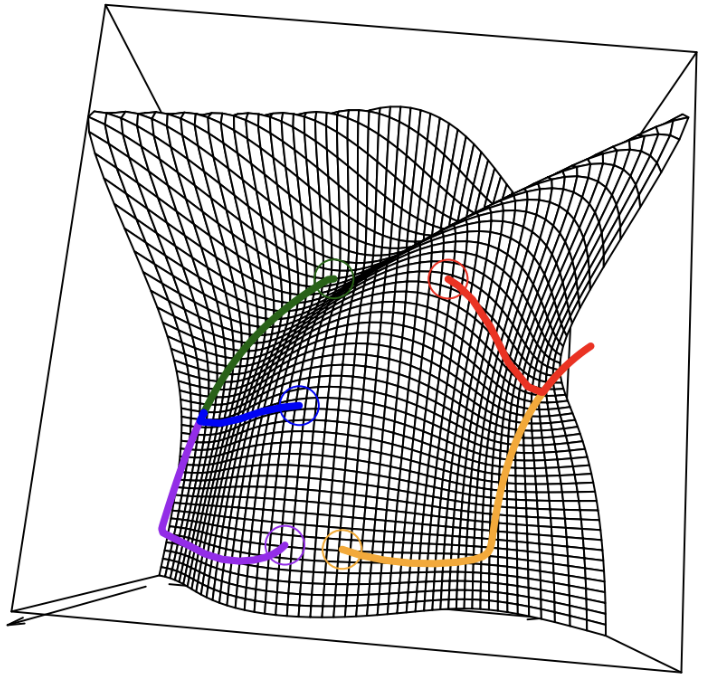
- where μ_t is the step size at iteration t.
- if μ_t is sufficiently small and $\nabla f(x_t) \neq 0$, guaranteed to decrease the value of f
- If f is convex, converges to **global minimum** under mild conditions

Gradient descent for convex functions



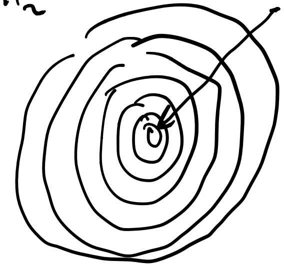
slide credit: R. Tibshirani

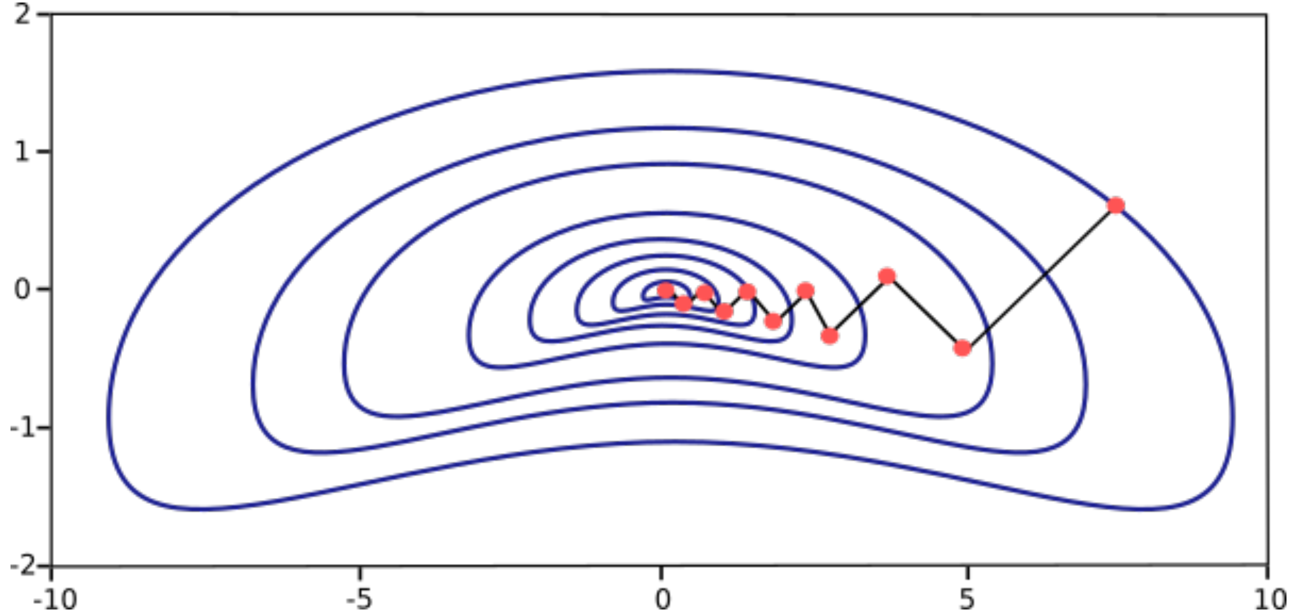
Gradient descent for non-convex functions



slide credit: R. Tibshirani

Gradient descent iterations $J^2 \quad \|Ax-b\|_2^2$

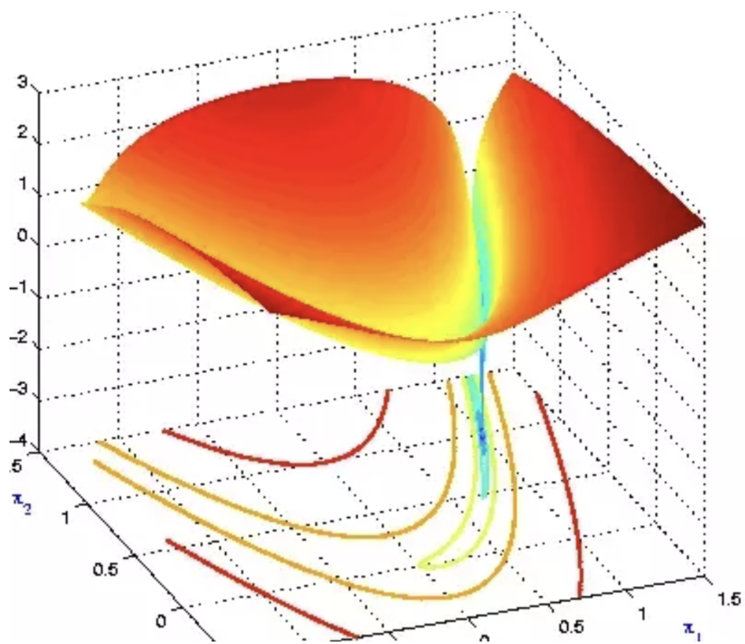




slide credit: A. Quesada

Gradient descent on highly curved functions

- Rosenbrock function (non-convex)
  $f(x_1, x_2) = (a - x_1)^2 + b(x_2 - x_1^2)^2$
  where a and b are parameters, e.g., $a = 1, b = 100$
  has a global minimum at $(x_1, x_2) = (a, a^2)$



Optimizing convex least squares cost

- Consider

$$\min_x \underbrace{\frac{1}{2}\|Ax - b\|_2^2}_{f(x)}$$

- gradient $\nabla f(x) = A^T(Ax - b)$
- Gradient Descent:

$$x_{t+1} = x_t - \mu A^T(Ax_t - b)$$

- fixed step size $\mu_t = \mu$

Optimizing convex least squares cost

$A^T b = A^T A x^*$

- Basic (in)equality method

(1) x^* minimizes $f(x)$, hence $\nabla f(x^*) = A^T(Ax^* - b) = 0$

(2) $x_{t+1} = x_t - \mu A^T(Ax_t - b)$

(3) define error $\Delta_t = x_t - x^*$

$$\underbrace{x_{t+1} - x^*}_{\Delta_{t+1}} = \underbrace{x_t - x^*}_{\Delta_t} - \mu \underbrace{A^T(Ax_t - b)}_{A^T A x_t - A^T A x^* = A^T A \cdot \Delta_t}$$

$$\boxed{\Delta_{t+1} = (I - \mu \cdot A^T A) \cdot \Delta_t}$$

$\Delta_1 = (I - \mu A^T A)\Delta_0$

$\Delta_2 = (I - \mu A^T A) \cdot \Delta_1 = (I - \mu A^T A)^2 \cdot \Delta_0$

$\Delta_T = (I - \mu \cdot A^T A)^T \cdot \Delta_0$

Optimizing convex least squares cost

- Basic (in)equality method

  (1) x^* minimizes $f(x)$, hence $\nabla f(x^*) = A^T(Ax^* - b) = 0$

  (2) $x_{t+1} = x_t - \mu A^T(Ax_t - b)$

  (3) define error $\Delta_t = x_t - x^*$

- $\Delta_{t+1} = \Delta_t - \mu A^T A \Delta_t$

Optimizing convex least squares cost

$\|Q\|_2 = \max_{\|x\|_2 \le 1} \|Q \cdot x\|_2 \Rightarrow \|\Delta_M\|_2 \le \underbrace{\|(I - \mu A^T A)^M\|_2}_{\sigma_{max}\cdot((I-\mu A^T A)^M)} \|\Delta\|$

$= \max_k |\lambda_k((I - \mu A^T A)^M)|$

- run gradient descent M iterations, i.e., $t = 1, \ldots, M$
- $\Delta_M = (I - \mu A^T A)^M \Delta_0$
- $\|\Delta_M\|_2 \le \sigma_{\max}\left((I - \mu A^T A)^M\right) \|\Delta_0\|_2$

$= \max_k |1 - \mu \lambda_k(A^T A)|^M$

$\sigma_{\max}\left(I - \mu A^T A\right)^M = \max_{i=1,..,d} \left|1 - \mu\lambda_i(A^T A)\right|^M$

where λ_i is the i-th eigenvalue in decreasing order

$\|Ax - b\|_2^2 = x^T A^T A x - 2b^T A x + b^T b$

$\nabla^2 = 2 \cdot A^T A$

$\max_i \underline{|1 - \mu\lambda_i|} < 1$

$\max_{i=1,\ldots,d} \max(1 - \mu\lambda_i, \mu\lambda_i - 1) < 1$

convergence condition

Optimizing convex least squares cost

Questions?