Convex Optimization

Lecture 15 - Gradient Descent in Machine Learning

Instructor: Yuanzhang Xiao

University of Hawaii at Manoa

Fall 2017

Today's Lecture

Outline

Why Gradient Descent in Machine Learning?

in theory:

- Newton methods much faster than gradient descent

in practice, especially in machine learning involving big data:

- gradient descent (and its variations) are used

why and how to use gradient descent in machine learning tasks?

Example – Least Square

least squares:

$$\text{minimize} \quad \|Ax - b\|_2^2$$

assuming $A^T A$ is invertible, we have analytical solution:

$$x^\star = (A^T A)^{-1}(A^T b)$$

building block of Newton methods (i.e., inverse of $\nabla^2 f(x)$)

Example – Least Square

with big data, we can have $A \in \mathbb{R}^{m \times n}$ where $m \approx 10^6$ and $n \approx 10^5$

the size of $(A^T A)^{-1}$ is $10^5 \times 10^5 \rightarrow 74$GB of memory

sparsity of A does not help, because $(A^T A)^{-1}$ can be dense

Example – Least Square

in gradient descent, we do

$$\begin{aligned} x^{(k+1)} &= x^{(k)} - t^{(k)} \nabla f(x^{(k)}) \\ &= x^{(k)} - t^{(k)}(A^T A x^{(k)} - A^T b) \end{aligned}$$

need to store

- $A \in \mathbb{R}^{m \times n}$: $\approx 740MB$ memory if density of A is 10^{-3}
- $A^T b \in \mathbb{R}^n$: $\approx 7MB$ memory
- $Ax^{(k)} \in \mathbb{R}^m$: $\approx 0.7MB$ memory
- $A^T(Ax^{(k)}) \in \mathbb{R}^n$: $\approx 0.7MB$ memory

memory needed in gradient descent: $\approx 740MB$

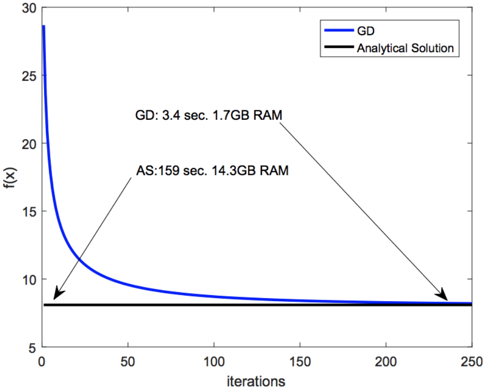
- as compared to $74GB$ in analytical solution or Newton method

Example – Least Square

least squares:

$$\text{minimize} \quad \|Ax - b\|_2^2$$

with $m = 500,000$, $n = 400,000$, and density of A being 10^{-3}



Outline

Motivation – Objective is Not Differentiable

least squares with regularization:

$$\text{minimize} \quad \|Ax - b\|_2^2 + \|x\|_1$$

do not want to introduce additional variables

- $n = 10^5 \rightarrow$ another 10^5 variables

do not want to use Newton method

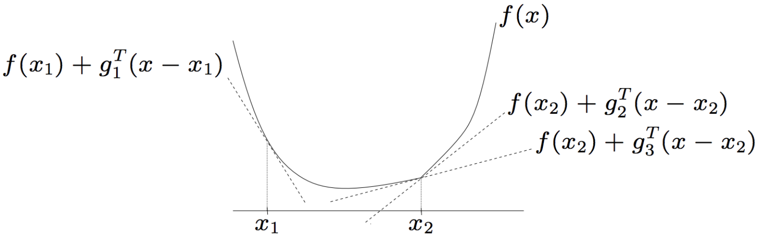
can we use variations of gradient descent?

Subgradient

g is a subgradient of f at x if:

$$f(y) \geq f(x) + g^T(y - x) \text{ for all } y$$

the line $f(x) + g^T(y - x)$ is a global underestimator of f

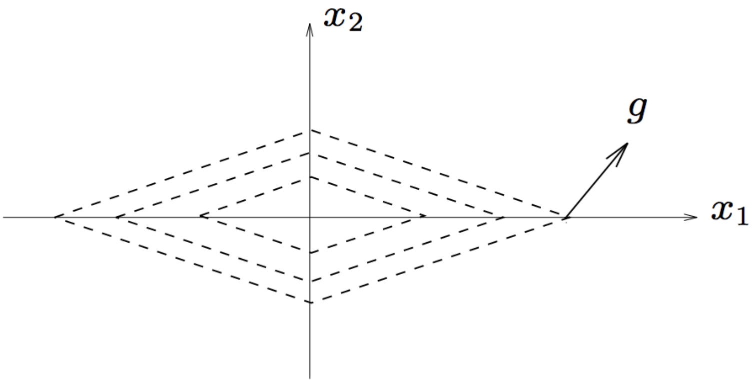


g_1 subgradient at x_1; g_2, g_3 subgradients at x_2

Descent Directions

$-g$ may not be descent direction for nondifferentiable f

example: $f(x) = |x_1| + 2|x_2|$



Subgradient Method

subgradient method:

$$x^{(k+1)} = x^{(k)} - \alpha_k g^{(k)}$$

- $g^{(k)}$ is any subgradient at $x^{(k)}$

not a descent method; need to keep track of the best point

$$f_{\text{best}}^{(k)} = \min_{i=1,\ldots,k} f(x^{(i)})$$

Subgradient Method – Step Size Rules

not a descent method – no backtracking line search

step sizes fixed ahead of run time

- constant step size: $\alpha_k = \alpha$
- square summable but not summable:

$$\sum_{k=1}^{\infty} \alpha_k^2 < \infty, \quad \sum_{k=1}^{\infty} \alpha_k = \infty$$

  - example: $\alpha_k = 1/k$
- diminishing, not summable:

$$\lim_{k\to\infty} \alpha_k = 0, \quad \sum_{k=1}^{\infty} \alpha_k = \infty$$

  - example: $\alpha_k = 1/\sqrt{k}$

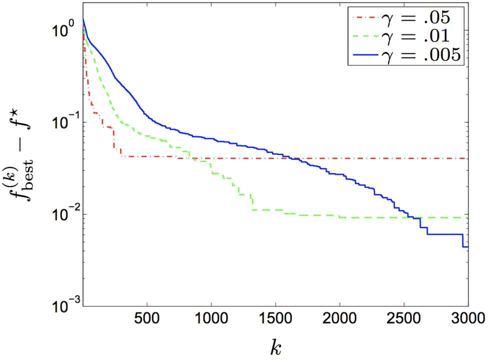
Example – Piecewise Linear Minimization

piecewise linear minimization

$$\text{minimize} \quad \max_{i=1,\ldots,m} (a_i^T x + b_i)$$

with $m = 100$, $n = 20$

constant step size:



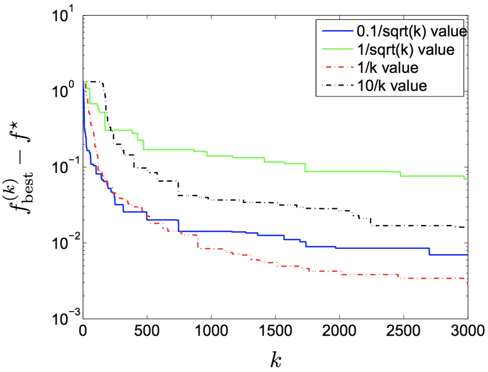
Example – Piecewise Linear Minimization

piecewise linear minimization

$$\text{minimize} \quad \max_{i=1,\dots,m} (a_i^T x + b_i)$$

with $m = 100$, $n = 20$

diminishing step size:



Outline

Stochastic Subgradient Method

least squares reformulation:

$$\text{minimize} \quad \sum_{i=1}^{m} (a_i^T x - b_i)^2$$

more generally, consider a problem:

$$\text{minimize} \quad \sum_{i=1}^{m} f_i(x)$$

stochastic subgradient method:

$$x^{(k+1)} = x^{(k)} - \alpha_k g_i^{(k)}$$

- $g_i^{(k)}$ is any subgradient of randomly chosen f_i at $x^{(k)}$

Advantages of Stochastic Subgradient Method

even less memory

- in the LS example, need to store $a_i \in \mathbb{R}^{100{,}000}$ (≈ 0.7MB)

sometimes data arrive consecutively

- update x after each data a_i arrives

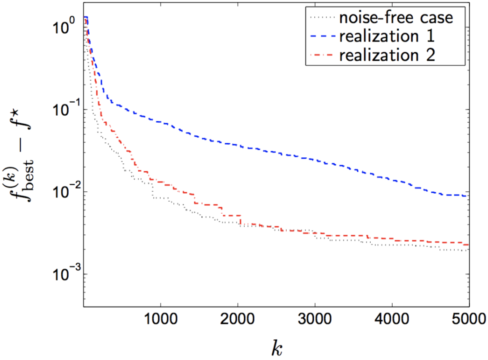
Example – Piecewise Linear Minimization

piecewise linear minimization

$$\text{minimize} \quad \max_{i=1,\ldots,m} (a_i^T x + b_i)$$

with $m = 100$, $n = 20$

convergence:



Example – Piecewise Linear Minimization

piecewise linear minimization

$$\text{minimize} \quad \max_{i=1,\ldots,m} (a_i^T x + b_i)$$

with $m = 100$, $n = 20$

empirical distribution of $f_{\text{best}}^{(k)} - f^\star$: