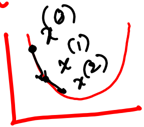
First Order Descent Methods

Instructor: Prof. Ganesh Ramakrishnan

General descent algorithm

convex fns



- Let us say we want to minimize a function $f(x)$
- The general descent algorithm involves two steps:
  - Determining a good descent direction $\Delta x^{(k)}$, typically forced to have unit norm
  - Determining the step size using some line search technique
- We want that $f(x^{(k+1)}) < f(x^{(k)})$
- If the function f is convex, we must have $\nabla^\top f(x^{(k)})(x^{(k+1)} - x^{(k)}) < 0$ (necessary condition)
- That is, the descent direction $\Delta x^{(k)}$ must make an obtuse angle with the gradient vector $\nabla f(x^{(k)})$

Dot prod between gradient at $x^{(k)}$ & step from $x^{(k)}$

Natural choice: $\Delta x^{(k)} = -\nabla f(x^k)$ --- But ignores curvature

General descent algorithm

- In descent for a convex function f, we must have:

$$f(x^{(k+1)}) \geq f(x^{(k)}) + \nabla^\top f(x^{(k)})(x^{(k+1)} - x^{(k)})$$

  Here, the LHS is the actual value and RHS is the linear approximation of $f(x^{(k+1)})$

- Since step size $t^{(k)} > 0$, $\nabla^\top f(x^{(k)}) \Delta x^{(k)} < 0$

- Algorithm:
  1. Set a starting point $x^{(0)}$
  2. repeat
     1. Determine $\Delta x^{(k)}$
     2. Choose a step size $t^{(k)} > 0$ using line search
     3. Obtain $x^{(k+1)} = x^{(k)} + t^{(k)} \Delta x^{(k)}$
     4. Set $k \leftarrow k + 1$

     until stopping criterion (such as $\left\|\nabla f(x^{(k+1)})\right\| < \epsilon$) is satisfied

What other condition could we use in this place?
$|f(x^{(k+1)}) - f(x^{(k)})| < \epsilon$

Steepest descent

$\Delta x = \arg\min_v \nabla^T f(x) v$

- The idea of steepest descent is to determine a descent direction such that for a unit step in that direction, the prediction of decrease in the objective is maximized
- However, consider $\Delta x = \text{argmin}_v \underbrace{\begin{bmatrix} -5 & 10 & 15 \end{bmatrix}}_{\nabla^T f(x)} v$

$$\implies \Delta x = \begin{bmatrix} \infty \\ -\infty \\ -\infty \end{bmatrix}$$

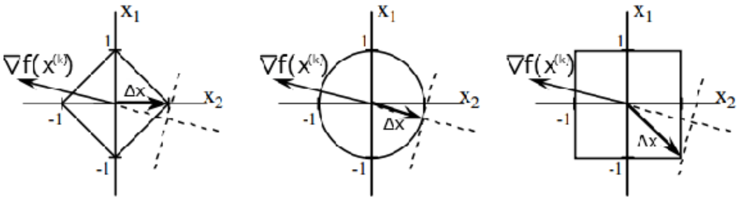
  which is unacceptable
- Thus, there is a necessity to restrict the norm of v
- The choice of the descent direction can be stated as:

$$\underbrace{\Delta x = \underset{v}{\text{argmin}}\, \nabla^\top f(x) v}$$

s.t. $\|v\| = 1$

Various choices of the norm result in different solutions for Δx

- For 2-norm, $\Delta x = -\frac{\nabla f(x^{(k)})}{\left\|\nabla f(x^{(k)})\right\|_2}$ *(gradient descent)*
- For 1-norm, $\Delta x = -\operatorname{sign}\left(\frac{\partial f(x^{(k)})}{\partial x_i^{(k)}}\right) e_i$, where e_i is the ith standard basis vector *(coordinate descent)*
- For ∞-norm, $\Delta x = -\operatorname{sign}(\nabla f(x^{(k)}))$



SDD in L1-norm   SDD in L2-norm   SDD in L∞ -norm

Gradient Descent

Interpretation of gradient descent

- Consider the optimization problem

$$x^* = \arg\min_{x \in \mathbf{R}^n} f(x)$$

- The idea behind gradient descent is that you start with a $x^0 \in \mathbf{R}^n$, and $\forall k = 0, 1, 2, \ldots$,

$$x^{k+1} = x^k + t^k \Delta x^k \rightarrow \Delta x^{(k)} = -\nabla f(x^k)$$

- x^{k+1} can be treated as a solution to a quadratic approximation of f around x^k

$$x^{k+1} = \arg\min_x f_{Q_k}(x)$$

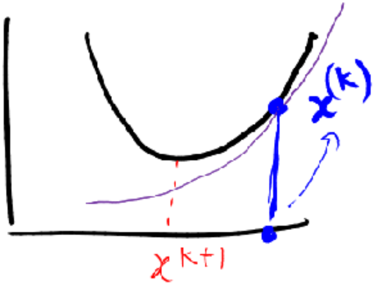
where:

$$f_{Q_k}(x) = f(x^k) + \nabla f(x^k)(x - x^k) + \frac{1}{2t}\|x - x^k\|^2$$

- At each iteration, we can consider the quadratic approximation

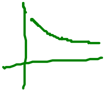
$$f_{Q_k}(x^{k+1}) = f(x^k) + \nabla f(x^k)^\top (x^{k+1} - x^k) + \frac{1}{2t}\left\| x^{k+1} - x^k \right\|^2$$

- Equating $\nabla f_{Q_k}(x^{k+1}) = 0$
  $\implies \nabla f(x^k) + \frac{1}{t}(x^{k+1} - x^k) = 0$
  $\implies x^{k+1} = x^k - t\nabla f(x^k)$



Finding the step size t

works for small curvature



if actual curvature is high

- If t is too large, we get diverging updates of x
- If t is too small, we get a very slow descent
- We need to find a t that is *just right*
- We discuss two ways of finding t:
  1. Exact line search
  2. Backtracking line search → tries to sense the curvature!

Both can be used for any descent algo

$$t^{(k)} = \underset{t}{\operatorname{argmin}} \; f(x^k + t\,\Delta x^k)$$

Exact line search

$$t^{k+1} = \underset{t}{\text{argmin}}\, f\left(x^k - t\nabla f(x^k)\right)$$

$$= \underset{t}{\text{argmin}}\, \phi(t)$$

→ possibly solve for $\phi'(t) = 0$

→ OR numerical methods

- This method gives the most optimal step size in the given descent direction $\nabla f(x^k)$
- It ensures that $f(x^{k+1}) \leq f(x^k)$ → By very design
- If f is itself quadratic, it gives an optimal solution to the minimization of f (since the quadratic approximation f_Q would become exact and no longer approximate)

Backtracking line search

Strong convexity

Choice of this gap: convexity only gives $f(x^{k+1}) \geq f(x^k) - t\|\nabla f(x^k)\|^2$

Existence is guaranteed

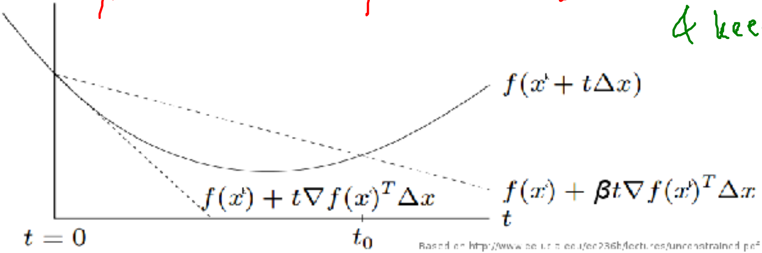
- The algorithm
  - Choose a $\beta \in (0, 1)$
  - Start with $t = 1$
  - While $f\left(x^k - t\nabla f(x^k)\right) > f(x^k) - \frac{t}{2}\left\|\nabla f(x^k)\right\|^2$, do
    - Update $t \leftarrow \beta t$

You want: $f(x^k - t\nabla f(x^k)) \leq f(x^k) - \text{gap}$ as stopping criterion

Interpretation of backtracking line search

You can associate β with t OR with Δx

& keep $t=1$



- Δx = direction of descent $= -\nabla f(x^k)$ for gradient descent
- A different way of understanding the varying step size with β: Multiplying t by β causes the interpolation to tilt as indicated in the figure

In practice Armijo conditions are used

Assumptions for proving the convergence of gradient descent

- $f : \mathbf{R}^n \to \mathbf{R}$ is convex and differentiable
- ∇f is Lipschitz continuous

- **Claim:** If $t^k \leq \frac{1}{L}$, then

$$f(x^k) - f(x^*) \leq \frac{\left\|x^0 - x^*\right\|^2}{2tk}$$

  - The gap between the optimal solution and the solution at the $\mathbf{kth}$ step is going to decrease with increasing step size t
  - $O(\frac{1}{k})$ rate or linear convergence