Descent Algorithms

Descent Algorithms for Optimizing Unconstrained Problems

Techniques relevant for most (convex) optimization problems that do not yield themselves to closed form solutions. We will start with unconstrained minimization.

$$\min_{\mathbf{x}\in\mathcal{D}} f(\mathbf{x})$$

For analysis:

- Assume that f is convex and differentiable and that it attains a finite optimal value p^*.
- Minimization techniques produce a sequence of points $\mathbf{x}^{(k)} \in \mathcal{D}$, $k = 0, 1, \ldots$ such that $f\left(\mathbf{x}^{(k)}\right) \to p^*$ as $k \to \infty$ or, $\nabla f\left(\mathbf{x}^{(k)}\right) \to \mathbf{0}$ as $k \to \infty$.
- Iterative techniques for optimization, further require a starting point $\mathbf{x}^{(0)} \in \mathcal{D}$ and sometimes that $epi(f)$ is closed. The $epi(f)$ can be inferred to be closed either if $\mathcal{D} = \Re^n$ or $f(\mathbf{x}) \to \infty$ as $\mathbf{x} \to \partial\mathcal{D}$. The function $f(x) = \frac{1}{x}$ for $x > 0$ is an example of a function whose $epi(f)$ is not closed.

Descent Algorithms

- Descent methods for minimization have been in use since the last 70 years or more.
- General idea: Next iterate $\mathbf{x}^{(k+1)}$ is the current iterate $\mathbf{x}^{(k)}$ added with a descent or search direction $\Delta\mathbf{x}^{(k)}$ (a unit vector), which is multiplied by a scale factor $t^{(k)}$, called the step length.

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + t^{(k)}\Delta\mathbf{x}^{(k)}$$

ideally we make progress in every iteration

- The incremental step is determined while aiming that $f(\mathbf{x}^{(k+1)}) < \widetilde{f}(\mathbf{x}^{(k)})$
- We assume that we are dealing with the **extended value extension** $\widetilde{f}$ of the convex function $f : \mathcal{D} \rightarrow \Re$, with $\mathcal{D} \subseteq \Re^n$ which returns ∞ for any point outside its domain. However, if we do so, we need to make sure that the initial point indeed lies in the domain $\mathcal{D}$.

Definition

$$\widetilde{f}(\mathbf{x}) = \begin{cases} f(\mathbf{x}) & \text{if } \mathbf{x} \in \mathcal{D} \\ \infty & \text{if } \mathbf{x} \notin \mathcal{D} \end{cases} \tag{27}$$

Descent Algorithms

- A single iteration of the general descent algorithm consists of two main steps, *viz.*,
  1. determining a good descent direction $\Delta \mathbf{x}^{(k)}$, which is typically forced to have unit norm and
  2. determining the step size using some line search technique.
- If the function f is convex, from the necessary and sufficient condition for convexity restated here for reference:

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

GIVEN

- We require that $f(\mathbf{x}^{(k+1)}) < f(\mathbf{x}^{(k)})$ and since $t^{(k)} > 0$, we must have

NEED

NECESSARY CONDITION TO MEET OUR NEED BASED ON WHAT IS GIVEN

Descent Algorithms

- A single iteration of the general descent algorithm consists of two main steps, *viz.*,
  1. determining a good descent direction $\Delta \mathbf{x}^{(k)}$, which is typically forced to have unit norm and
  2. determining the step size using some line search technique.
- If the function f is convex, from the necessary and sufficient condition for convexity restated here for reference:

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

- We require that $f(\mathbf{x}^{(k+1)}) < f(\mathbf{x}^{(k)})$ and since $t^{(k)} > 0$, we must have

$$\nabla^T f(\mathbf{x}^{(k)}) \Delta \mathbf{x}^{(k)} < 0$$

  That is, the descent direction $\Delta \mathbf{x}^{(k)}$ must make (sufficiently) obtuse angle $\left(\theta \in \left(\frac{\pi}{2}, \frac{3\pi}{2}\right)\right)$ with the gradient vector

  Since the inequality above is only necessary

- A natural choice of $\Delta \mathbf{x}^{(k)}$ that satisfies the above necessary condition is

Descent Algorithms

- A single iteration of the general descent algorithm consists of two main steps, *viz.*,
  1. determining a good descent direction $\Delta\mathbf{x}^{(k)}$, which is typically forced to have unit norm and
  2. determining the step size using some line search technique.
- If the function f is convex, from the necessary and sufficient condition for convexity restated here for reference:

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

- We require that $f(\mathbf{x}^{(k+1)}) < f(\mathbf{x}^{(k)})$ and since $t^{(k)} > 0$, we must have

$$\nabla^T f(\mathbf{x}^{(k)})\Delta\mathbf{x}^{(k)} < 0$$

  That is, the descent direction $\Delta\mathbf{x}^{(k)}$ must make (sufficiently) obtuse angle ($\theta \in \left(\frac{\pi}{2}, \frac{3\pi}{2}\right)$) with the gradient vector
- A natural choice of $\Delta\mathbf{x}^{(k)}$ that satisfies the above necessary condition is $-\nabla f(\mathbf{x}^{(k)})$ (gradient descent algorithm)

Descent Algorithms (contd.)

Find a starting point $\mathbf{x}^{(0)} \in \mathcal{D}$

repeat

1. Determine $\Delta\mathbf{x}^{(k)}$.
2. Choose a step size $t^{(k)} > 0$ using ray[a] search.
3. Obtain $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + t^{(k)}\Delta\mathbf{x}^{(k)}$.
4. Set $k = k + 1$.

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

[a] Many textbooks refer to this as line search, but we prefer to call it ray search, since the step must be positive.

Figure 7: The general descent algorithm.

There are many different empirical techniques for ray search, though it matters much less than the search for the descent direction. These techniques reduce the $n-$dimensional problem to a $1-$dimensional problem, which can be easy to solve by use of plotting and eyeballing or even exact search.

Finding the step size t

- 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 ray search
  2. Backtracking ray search

Exact ray search

Choose a step length that minimizes the function
in the chosen descent direction

$$t^{k+1} = \underset{t}{\operatorname{argmin}}\, f\left(\mathbf{x}^k + t\Delta\mathbf{x}^k\right)$$

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

- This method gives the most optimal step size in the given descent direction $\Delta\mathbf{x}^k$
- It ensures that $f(x^{k+1}) \leq f(x^k)$. Why?

Given the myopic goal of making
f(x^(k+1)) as smaller as possible than
f(x^k)

Exact ray search

From last class, convex function f restricted to a line (\phi) is also convex along that line (that is along t)

$$t^{k+1} = \underset{t}{\operatorname{argmin}}\, f\left(\mathbf{x}^k + t\Delta\mathbf{x}^k\right)$$

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

- This method gives the most optimal step size in the given descent direction $\Delta\mathbf{x}^k$
- It ensures that $f(x^{k+1}) \leq f(x^k)$. Why? Because $\phi(t^{k+1}) = f(\mathbf{x}^k + t^{k+1}\Delta\mathbf{x}^k) = \min_t \phi(t) = \min_t f\left(\mathbf{x}^k + t\Delta\mathbf{x}^k\right) \leq \phi(0) = f(x^k)$
- **Homework1**: Consider the function

$$f(\mathbf{x}) = x_1^2 - 4x_1 + 2x_1x_2 + 2x_2^2 + 2x_2 + 14$$

This function has a minimum at $\mathbf{x} = (5, -3)$. Suppose you are at a point $(4, -4)^T$ after few iterations, and $\Delta\mathbf{x} = -\nabla f(\mathbf{x})$ at every $\mathbf{x}$, then using the **exact line search algorithm**, find the point for the next iteration. In how many steps will the algorithm converge?

Ray Search for Descent: Options

1. **Exact ray search:** The exact ray search seeks a scaling factor t that satisfies

$$t = \underset{t>0}{\operatorname{argmin}} f(\mathbf{x} + t\Delta\mathbf{x}) \tag{28}$$

This might itself require us to invoke a numerical solver for \phi(t)

This may be expensive. But more importantly, is it worth it?

Can we look at the geometry of descent and come up with some intuitive criteria that ray search should meet?

1) Sufficient decrease in the function
2) Sufficient decrease in the slope after update

Ray Search for Descent: Options

1. **Exact ray search:** The exact ray search seeks a scaling factor t that satisfies

$$t = \underset{t>0}{\operatorname{argmin}} f(\mathbf{x} + t\Delta\mathbf{x}) \tag{28}$$

2. **Backtracking ray search:** The exact line search may not be feasible or could be expensive to compute for complex non-linear functions. A relatively simpler ray search iterates over values of step size starting from 1 and scaling it down by a factor of $\beta \in (0, \frac{1}{2})$ after every iteration till the following condition, called the *Armijo condition* is satisfied for some $0 < c_1 < 1$.

Negative term assuming descent direction

$$f(\mathbf{x} + t\Delta\mathbf{x}) \leq f(\mathbf{x}) + c_1 t \nabla^T f(\mathbf{x}) \Delta\mathbf{x} \tag{29}$$

Based on first order convexity condition, it can be inferred that when $c_1 = 1$,

inequality in (29) cannot hold (and gets flipped)

Ray Search for Descent: Options

1. **Exact ray search:** The exact ray search seeks a scaling factor t that satisfies

$$t = \underset{t>0}{\operatorname{argmin}} f(\mathbf{x} + t\Delta\mathbf{x}) \tag{28}$$

2. **Backtracking ray search:** The exact line search may not be feasible or could be expensive to compute for complex non-linear functions. A relatively simpler ray search iterates over values of step size starting from 1 and scaling it down by a factor of $\beta \in (0, \frac{1}{2})$ after every iteration till the following condition, called the *Armijo condition* is satisfied for some $0 < c_1 < 1$.

$$f(\mathbf{x} + t\Delta\mathbf{x}) \leq f(\mathbf{x}) + c_1 t \nabla^T f(\mathbf{x}) \Delta\mathbf{x} \tag{29}$$

Based on first order convexity condition, it can be inferred that when $c_1 = 1$, the right hand side of (29) gives a lower bound on the value of $f(\mathbf{x} + t\Delta\mathbf{x})$ and hence

(29) cannot hold

Ray Search for Descent: Options

1. **Exact ray search:** The exact ray search seeks a scaling factor t that satisfies

$$t = \underset{t>0}{\operatorname{argmin}} f(\mathbf{x} + t\Delta\mathbf{x}) \tag{28}$$

2. **Backtracking ray search:** The exact line search may not be feasible or could be expensive to compute for complex non-linear functions. A relatively simpler ray search iterates over values of step size starting from 1 and scaling it down by a factor of $\beta \in (0, \frac{1}{2})$ after every iteration till the following condition, called the *Armijo condition* is satisfied for some $0 < c_1 < 1$.

$$f(\mathbf{x} + t\Delta\mathbf{x}) \leq f(\mathbf{x}) + c_1 t \nabla^T f(\mathbf{x}) \Delta\mathbf{x} \tag{29}$$

Based on first order convexity condition, it can be inferred that when $c_1 = 1$, the right hand side of (29) gives a lower bound on the value of $f(\mathbf{x} + t\Delta\mathbf{x})$ and hence (29) can never hold. The Armijo condition simply ensures that t decreases f sufficiently.

Backtracking ray search

c1 is fixed to a value (0,1) so that sufficient decrease is ensured when the search for t is complete

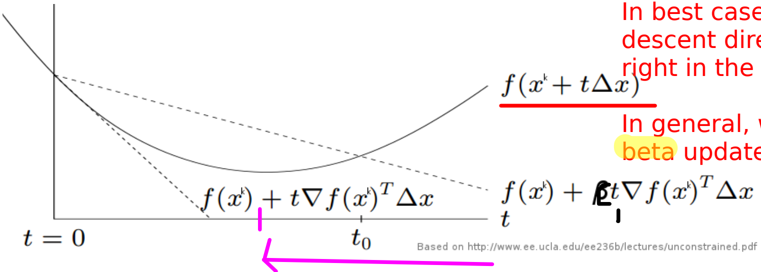
- The algorithm
  - Choose a $\beta \in (0, 1)$
  - Start with $t = 1$
  - Until $f(\mathbf{x} + t\Delta\mathbf{x}) < f(\mathbf{x}) + c_1 t \nabla^T f(\mathbf{x}) \Delta\mathbf{x}$, do
    - Update $t \leftarrow \beta t$

Questions:

1) What is a good choice of c1 in (0,1)? Further from 1 will make it feasible to satisfy Armijo condition. Further from 0 will make the decrease sufficient Often c1 = 0.5
2) Will Armijo condition be satisfied for any given c1 in (0,1)

Given that l(t) was the tightest supporting hyperplane for any c1 < 1, the rotated l(t) should intersect the graph of the function. Hence Armijo should be satisfied for some t's, a

Interpretation of backtracking line search



Based on http://www.ee.ucla.edu/ee236b/lectures/unconstrained.pdf

In best case with t=1 and the necessary descent direction, we have Armijo condition right in the first step

In general, we might have to make several beta updates before the Armijo condition is met

- $\Delta x =$ 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

Homework 2: Let $f(x) = x^2$ for $x \in \Re$. Let $x^0 = 2$, $\Delta x^k = -1$ for all k (since it is a valid descent direction of $x > 0$) and $x^k = 1 + 2^{-k}$. What is the step size t^k implicitly being used. While t^k satisifies the Armijo condition (determine a c_1) is this choice of step size ok?