Notes on ‘CS5660: Convex Optimization 2’

Pranav K Nayak

Lecture 2

02.01.2024

The course will deal mainly in *blackbox* solvers. Given an optimization problem of the form

$$\begin{aligned}&\min f(x)\\ s.t.\;\; &g(x) < 0\end{aligned}$$

Nothing is known about f. All we have is an *oracle*, that we can query at specific instances of x to get a single local instance of information about f.

These blackbox solvers tend to be iterative, meaning that they have an initial guess for the optimum, and then they iteratively improve that by incorporating information provided by calls to the oracle.

In general:

$$\begin{aligned}&x^{(0)} \xrightarrow{\text{oracle}(x^{(0)})} \nabla f(x^{(0)}) \xrightarrow{\text{update}} x^{(1)}\\ &\xrightarrow{\text{oracle}(x^{(1)})} \nabla f(x^{(1)}) \xrightarrow{\text{update}} x^{(2)}\\ &\dots\, x^{(k-1)} \xrightarrow{\text{oracle}(x^{(k-1)})} \nabla f(x^{(k)})\end{aligned}$$

Here, the oracle is a *first-order* one, meaning that querying it gives us the gradient.

Assumption: We restrict ourselves to algorithms that compute

$$x^{(k+1)} = x^{(k)} + \sum_{i=1}^{k} \lambda_i \nabla f(x^{(i)}).$$

Intuitions behind Gradient Descent

GD can be thought of in two ways:

- A Greedy Algorithm
- A Divide and Conquer Linear Approximation Approach

The Greedy Algorithm

“Choose the direction of maximum local decrease”.

Say we’re at some point x_k, and we want to move in a direction d. Then the update rule can be formalized as $x_{k+1} = x_k + sd$, where s is how far we move in d’s direction, and we set $s \geq 0$.

The directional gradient of the function f in the direction d is $\nabla f(x_k)^T d$. We want this to point in the direction of greatest decrease, or least increase, which means that choosing d can now be formulated as an optimization problem. To make our lives easier, we can restrict the possible values of d, since otherwise solving this problem would not be likely.

Say we limit d to being in the unit circle $\mathcal{B} = \{d : ||d||_n \leq 1\}$

Our optimization problem can then be stated as

$$\begin{aligned}\arg\min_{d\in\mathbb{R}^n} &\nabla f(x_k)^T d\\ &||d||_n \leq 1\end{aligned}$$

Intuitively, we can see that we're trying to minimize the dot product between $\nabla f(x_k)$ and d, for which d (normalized), should be $-\frac{\nabla f(x_k)}{||\nabla f(x_k)||_n}$

Thus, the update rule becomes

$$x_{k+1} = x_k - s\nabla f(x_k).$$

The norm is absorbed into s.

Thus, we've arrived at our rule for gradient descent.

Depending on how we restrict the search space for d, we can bias certain values for d. The unit hypersphere is an unbiased search space, but what if we were to choose, instead, the space $d : ||d||_1 \leq 1$? This would end up being a unit hypercube, centered at the origin, whose vertices lie on symmetrical points along the axes. In such a scenario, it is more likely that the vertex points would cause maxima and minima to occur during the dot product if d was to be one of the vertex points (intuitively, we would be selecting a single element from the gradient's coordinates, which, if it is the maximum or minimum, will be greater or smaller than the weighted average of the rest of the elements).

The Linear Approximation Approach

Any function can be approximated, in a local neighbourhood, by a linear form, through its Taylor approximation.