

Convex Optimization Assignment 1 Solutions

General Instructions

The following document contains the solutions to the questions for Assignment 1. Please note that the solutions provided may not be the only possible way to solve the questions. They indicate only one of the many (possibly) valid solutions. The solutions provided are relatively crisp and do not include all the steps that you must have. Your solution should be logical and contain all supporting arguments. Feel free to contact any of the TAs via email in case of any discrepancy you find in the solutions provided.

Question 1

There's only 3 possible cases in the Bisection Search Method, for a unimodal function with a minima.

Case 1:

Considering $f'(x) = 0$ In this case, $x^{(k)} = x^*$

Case 2:

Considering $f'(x) > 0$. If the slope is greater than zero, the point $x^{(k)}$ must lie to the right of the minima x^* . The search space is reduced to $[l^{(k)}, x^{(k)}]$. Now, $x^{(k+1)} = \frac{l^{(k)} + x^{(k)}}{2}$

Case 3:

Considering $f'(x) < 0$. If the slope is lesser than zero, the point $x^{(k)}$ must lie to the left of the minima x^* . The search space is reduced to $[x^{(k)}, r^{(k)}]$. Now, $x^{(k+1)} = \frac{r^{(k)} + x^{(k)}}{2}$

Thus, at the k^{th} iteration, either $x^{(k)} = x^*$, or the search space reduces by a factor of 2 for the next iteration. In fact, after k steps, the search space has been reduced by a factor of 2^k . Thus, the width of the search space after k iterations, if the original width search space was $(r^{(0)} - l^{(0)}) = c$, which is finite according to the initial condition of the algorithm, is given by:

$$w^{(k)} = \frac{c}{2^k}$$
$$\lim_{k \rightarrow \infty} w^{(k)} = \lim_{k \rightarrow \infty} \left(\frac{c}{2^k} \right) = 0$$

Therefore, $x^{(k)}$ eventually converges to x^* .

Question 2

Let the initial interval's size be $w^{(0)}$

In Golden Section Search, the size of the interval after the first iteration i.e. $w^{(1)}$ is given by,

$$\begin{aligned}w^{(1)} &= w^{(0)} - \rho w^{(0)} \\ &= (1 - \rho)w^{(0)}\end{aligned}$$

Similarly, after the second iteration, the size of the interval i.e. $w^{(2)}$ is given by,

$$\begin{aligned}w^{(2)} &= (1 - \rho)w^{(0)} \\ &= (1 - \rho)(1 - \rho)w^{(0)} \\ &= (1 - \rho)^2 w^{(0)}\end{aligned}$$

So, after the k^{th} iteration, the size of the interval i.e. $w^{(k)}$ is given by,

$$w^{(k)} = (1 - \rho)^k w^{(0)}$$

So, the size of the interval during the k^{th} iteration decreases by,

$$\begin{aligned}\Delta_k &= w^{(k-1)} - w^{(k)} \\ &= (1 - \rho)^{k-1} w^{(0)} - (1 - \rho)^k w^{(0)} \\ &= (1 - \rho)^{k-1} w^{(0)} [1 - (1 - \rho)] \\ &= \rho (1 - \rho)^{k-1} w^{(0)}\end{aligned}$$

Thus, in the k^{th} iteration the size of the initial interval i.e. w_0 decreases by a factor of $\rho(1 - \rho)^{k-1} \approx (0.382)(0.618)^{k-1}$.

Question 3

Given a convex objective function, we need to how many number of iterations we'll need to find a result which is ε -close to the minima of the function.

Using Bisection Search

Assume you're given the interval $[a, b]$ to find the given minima for the function.

- Bisection Search Algorithm, basically after every iteration decreases the width of our search space in half.
- To be ε -close to the minima, we'll need the width of our interval to be less than ε .
- So after n iterations, the width would come down to

$$W_n = \frac{|b-a|}{2^n}$$

- So we get the equation -

$$\frac{|b-a|}{2^n} \leq \varepsilon$$

- Solving for n , we get -

$$n \leq \log_2 \left(\frac{b-a}{\varepsilon} \right)$$

Thus, the algorithm runs in $\mathcal{O}(\log(\frac{1}{\varepsilon}))$ iterations.

Using Golden Section Search

Assume you're given the interval $[a, b]$ to find the given minima for the function.

Golden Section Search, after every iteration, the width decreases by a factor of $1/\varphi$ where φ is the golden ratio. To be ε -close to the minima, we'll need the width of our interval to be less than ε . Thus, after n iterations, the width would become:

$$W_n = \frac{|b-a|}{\varphi^n}$$

So we get the equation

$$\frac{|b-a|}{\varphi^n} \leq \varepsilon$$

Solving for n , we get

$$n \leq \log_{\varphi} \left(\frac{b-a}{\varepsilon} \right)$$

Thus, the algorithm runs in $\mathcal{O}(\log(\frac{1}{\varepsilon}))$ iterations.

Let us compare which method requires less number of iterations. Taking the worst case scenario, we can compare the number of iterations required in both the methods.

$$\frac{n_{\text{Bisection}}}{n_{\text{Golden}}} = \frac{\log_2 \left(\frac{b-a}{\varepsilon} \right)}{\log_{\varphi} \left(\frac{b-a}{\varepsilon} \right)} \approx 0.69$$

Even though the number of iterations are of the same order, the Bisection Method requires less number of iterations than the Golden Section Search because of the constant factor.

Therefore, the Bisection Method seems to be the better method in terms of rate of convergence.

Question 4

a) Given the set \mathbb{A} as:

$$\mathbb{A} = \left\{ \begin{bmatrix} 5 \\ 9 \\ 3 \end{bmatrix}, \begin{bmatrix} 2 \\ 1 \\ 4 \end{bmatrix} \right\}$$

Notice, \mathbb{A} contains two 3 dimensional vectors. Thus, its span will be a 2-dimensional plane in 3-dimensional space. The plane's equation can be found too. A similar procedure can be repeated for

$$\mathbb{B} = \left\{ \begin{bmatrix} 7 \\ 1 \\ 7 \end{bmatrix}, \begin{bmatrix} 3 \\ 8 \\ -1 \end{bmatrix} \right\}$$

However, the equations may not be immediately identical because a plane is given by $\hat{\mathbf{n}} \cdot (\mathbf{r} - \mathbf{r}_0) = 0$. Thus, there is variation in the equation with \mathbf{r}_0 . Therefore, we opt for a different approach. To compare the subspace spanned, we need some form of row space/column space. That is, if we can somehow figure out the space spanned by the rows/columns by the matrices formed by these vectors, we are done.

Thus, firstly, express the vectors in the sets as matrices:

$$\mathbf{A}_1 = \begin{bmatrix} 5 & 9 & 3 \\ 2 & 1 & 4 \end{bmatrix}$$
$$\mathbf{B}_1 = \begin{bmatrix} 7 & 1 & 7 \\ 3 & 8 & -1 \end{bmatrix}$$

Perform elementary row operations to try to match at least 1 row of \mathbf{A}_1 to that of \mathbf{B}_1 . Remember, these operations do not change the row or column space. (Proved later)

$$\mathcal{R}_2 \leftarrow \mathcal{R}_1 - \mathcal{R}_2$$
$$\Rightarrow \mathbf{A}'_1 = \begin{bmatrix} 5 & 9 & 3 \\ 3 & 8 & -1 \end{bmatrix}$$

Beyond this, to check whether a row operation can be performed to reach \mathbf{B}_1 , simply seek the solutions for the linear equation $\alpha \mathbf{a}'_1 + \beta \mathbf{a}'_2 = \mathbf{b}_1$. In this case, such a solution does not exist.

b) Again, we are given the sets \mathbb{A} and \mathbb{B} as:

$$\mathbb{A} = \left\{ \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}, \begin{bmatrix} 6 \\ 5 \\ 4 \end{bmatrix} \right\}$$

$$\mathbb{B} = \left\{ \begin{bmatrix} 8 \\ 9 \\ 10 \end{bmatrix}, \begin{bmatrix} 5 \\ 3 \\ 1 \end{bmatrix} \right\}$$

Express the vectors in the sets as matrices:

$$\mathbf{A}_2 = \begin{bmatrix} 1 & 2 & 3 \\ 6 & 5 & 4 \end{bmatrix}$$

$$\mathbf{B}_2 = \begin{bmatrix} 8 & 9 & 10 \\ 5 & 3 & 1 \end{bmatrix}$$

Perform elementary row operations to try to match at least 1 row of \mathbf{A}_2 to that of \mathbf{B}_2 .

$$\mathcal{R}_2 \leftarrow \mathcal{R}_2 - \mathcal{R}_1$$

$$\Rightarrow \mathbf{A}'_2 = \begin{bmatrix} 1 & 2 & 3 \\ 5 & 3 & 1 \end{bmatrix}$$

$$\mathcal{R}_1 \leftarrow 3\mathcal{R}_1 + \mathcal{R}_2$$

$$\Rightarrow \mathbf{A}'_2 = \begin{bmatrix} 8 & 9 & 10 \\ 5 & 3 & 1 \end{bmatrix} = \mathbf{B}'_2$$

Thus, both sets span the same subspace.

Proving that elementary Row Operations Do Not Change Row Space. Let $\{\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_m\}$ be the row vectors of a matrix \mathbf{A} and let $\{\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_m\}$ be the row vectors of a matrix \mathbf{B} . Let \mathbf{B} be such that it has been obtained from elementary row operations on \mathbf{A} . Thus,

$$\mathbf{b}_i = \mathbf{a}_i + c\mathbf{a}_j$$

Repeat the same procedure for \mathbf{a}_i in terms of \mathbf{b}_i and \mathbf{b}_j . Thus:

$$\text{span}(\{\mathbf{a}_1, \mathbf{a}_2 \dots \mathbf{a}_m\}) \subseteq \text{span}(\{\mathbf{b}_1, \mathbf{b}_2 \dots \mathbf{b}_m\})$$

$$\text{span}(\{\mathbf{b}_1, \mathbf{b}_2 \dots \mathbf{b}_m\}) \subseteq \text{span}(\{\mathbf{a}_1, \mathbf{a}_2 \dots \mathbf{a}_m\})$$

In other words, their spans are equal. Thus, both have the same row space.