Mathematics and Computational Methods for Complex Systems

Coursework 1

Miguel de Llanza Varona

CandNo: 234717

Task 1: Gradient ascent and Hill-climbing with a simple function

The function $g(x, y)$ that defines the simple landscape is evaluated in the $[-2,2]$ for both x and y. In this evaluation range, $g(x, y)$ has a local maximum when $x = 0$ and $y = 2$ with height $g(0,2) = 3$, and the global maximum when $x = 2$ and $y = 2$, with height $g(2,2) = 4$.

The values for the partial derivatives of $g(x, y)$ are the following:

$$\frac{dg}{dx} = \begin{cases} -1, & if\ x \in (0, 0.5) \\ 0, & if\ x = 0 \\ 1, & if\ x \in [-2, -0.5]\ or\ \ x \in [0.5,2] \\ 3, & if\ x \in (-0.5, 0) \end{cases} \quad (1)$$

$$\frac{dg}{dy} = 1\ \ \forall y \mid y \in [2, -2] \quad (2)$$

- **Question 1**

As shown in Figure 1, given the default parameters for both algorithms and a sample of random starting points, there are considerably more points from which hill-climbing converges to the global optimum than in gradient ascent. In questions 2 and 3 I will detail why hill-climbing gets to the global maximum from more points than gradient ascent and how the number of iterations and the learning rate or mutation distance modifies the performance of each algorithm depending on the region of the landscape being evaluated.

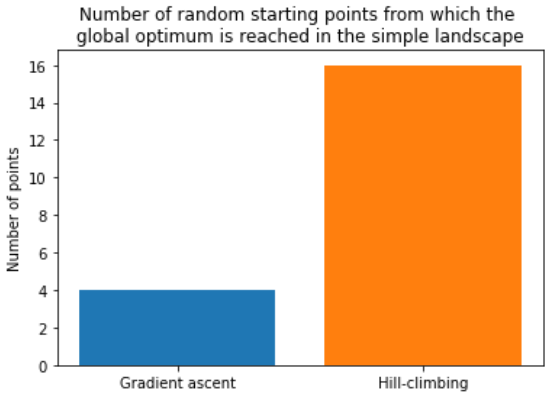

Figure 1. Testing performance of each algorithm with some random points

- **Question 2**

Figure 2 shows the performance of each algorithm for a wide range of starting points across the landscape, using the default parameters. Regarding gradient ascent, it can be clearly seen that when the x-dimension takes values in the interval $[-2,0.5)$, does not converge to the global optimum. On the other hand, the results show that hill-climbing converges to the global maximum in points from every region of the landscape except from points around $x = 0$, where a straight blue line is visible. In addition, a clear pattern emerges in gradient ascent, where the number of iterations slowly increases as the value in the y-dimension decreases. This is expected, since points further from the global optimum will require more iterations to reach it than nearer points. Regarding hill-climbing, a similar, but noisier pattern emerges around the same landscape region. It is noisier because hill-climbing is a stochastic algorithm, and this implies that there is some chance that starting points that reach the global optimum in gradient ascent might i) not reach it; ii) reach it faster; or iii) reach it slower.

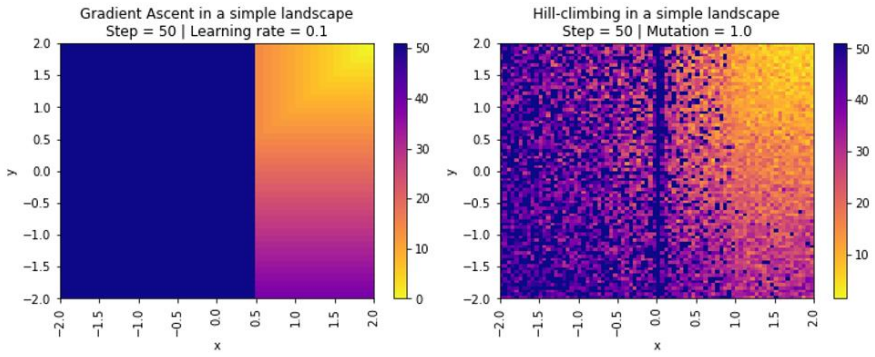

Figure 2. Performance of gradient ascent and hill-climbing with the default parameters.

Before examining the effects of the learning rate in gradient ascent, let's see how affects varying the number of iterations. Figure 3 (first row) illustrates that changing the number of iterations does not affect the performance of gradient ascent once in those points from which is already possible to get to the global optimum. On the other hand, a low learning rate along with a low number of iterations can impede the convergence to the global optimum of gradient ascent in some starting points. Thus, once convergence to the global maximum is possible for a particular starting point, no matter how the number of iterations is increased (while keeping the same learning rate), the performance of gradient ascent for that starting point will not vary. Regarding hill-climbing, as can be seen in Figure 3 (second row) the number of iterations has a greater impact on its performance. As the number of iterations increases, more points along the landscape are able to reach the global maximum.

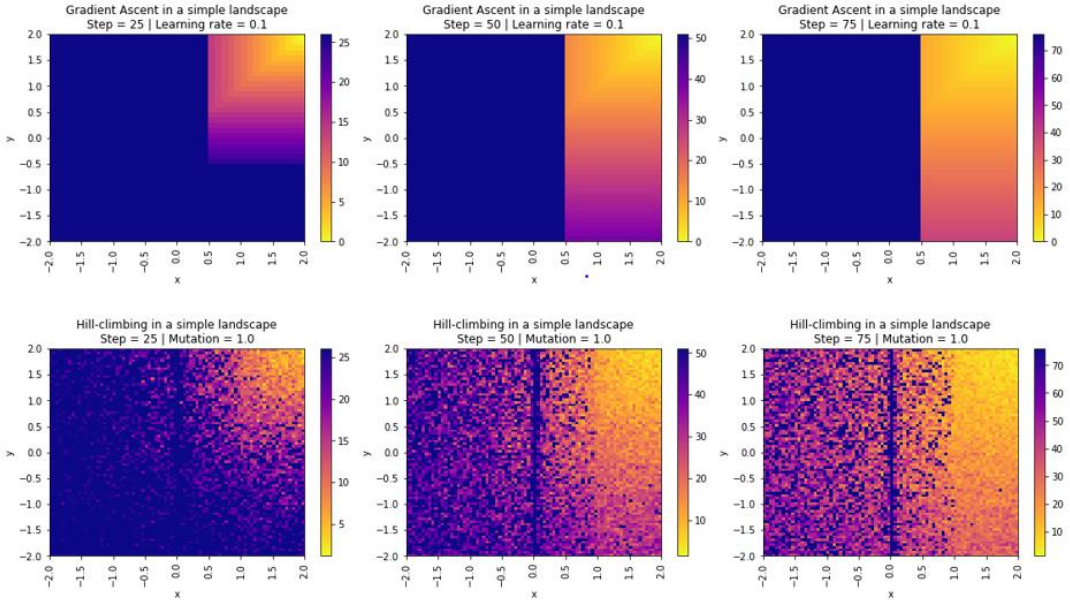

Figure 3. Number of iterations to reach the global maximum in gradient ascent and hill-climbing.

Let's see what happens when both parameters are changed in gradient ascent. In Figure 4 (left), the learning rate is set to 0.3 and the number of iterations vary from 25 to 100. As can be seen, the results are exactly the same in the four plots. On the contrary, keeping the number of iterations the same and changing the learning rate has an impact in its performance (right plots in Figure 4). Thus, changing the number of iterations (if the minimum is higher enough) does not affect to the performance of gradient ascent, while modifying the learning rate does.

In the case of hill-climbing, both changing the number of steps and the mutation distance affect its performance. In Figure 3 (second row) can be seen the impact of changing the number of iterations while keeping the same mutation distance (already explained above). Figure 5 shows how, similarly as in the previous case, as the mutation distance increases, the performance increases. Specifically, hill-climbing converges to the global maximum from more points along the landscape.

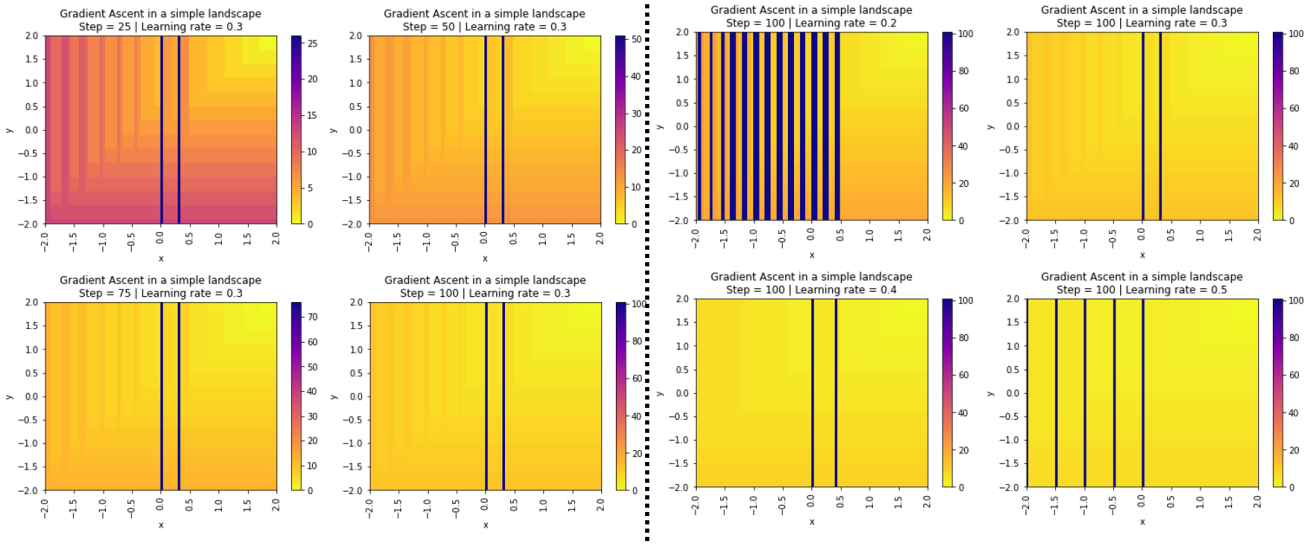

Figure 4. 4-figures left: performance of gradient ascent changing the number of iterations. 4-figures right: performance of gradient ascent changing the learning rate.

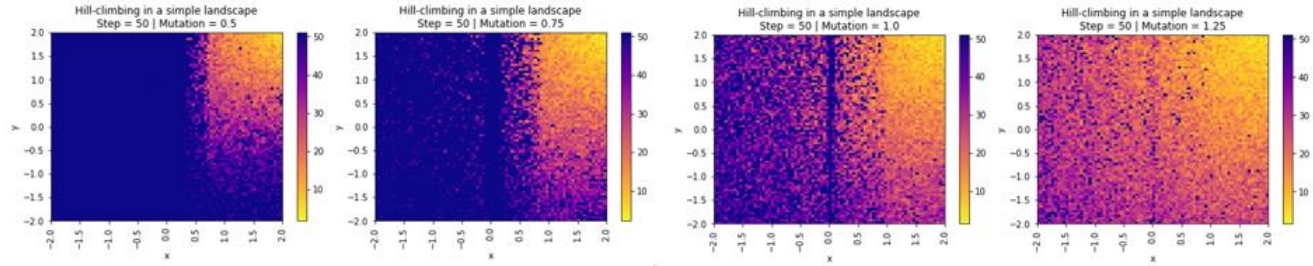

Figure 5. Performance of hill-climbing changing the mutation distance.

- **Question 3**

Let's turn to the anomalies found in each algorithm. First, as mentioned before, in gradient ascent no starting point where x belongs to the interval $[-2,0.5)$ reaches the global optimum. Why? Among this range, it is possible to define a lower bound for the learning rate such that running gradient ascent for *some* starting points in this range will converge to the global maximum. The lower bound can be obtained solving the following inequality:

$$\eta * \frac{dg}{dx} > \frac{1}{2}$$

$$\eta * 3 > \frac{1}{2}$$

$$\eta > \frac{1}{6}$$

The crucial interval to find the lower bound is when $x \in (-0.5, 0)$ because, as shown in eq. 1, for some points to avoid the local maximum, the minimum step they have to take has to be greater than $\frac{1}{2}$. For instance, $x = -0.1$, by eq.1, the gradient of g with respect to x will be 3. Thus, setting a learning rate $\eta = \frac{1}{4}$ will update the x value to $x = 0.65$. From this point, any further update of the x value will increase x (by eq. 1, the gradient of g with respect to x will always be 1) until the global optimum is reached. On the contrary, for any starting point such that $x \in [-2, 0.5)$ and for any learning rate *not* greater than $\frac{1}{6}$, gradient ascent will *never* converge to the global maximum. Since the default learning rate is $\eta = 0.1$ and $\eta < \frac{1}{6}$, the results for gradient ascent shown in Figure 2 are consistent with what has been stated.

All the reasoning developed above has been almost focused on the behavior of $g(x,y)$ around the x-dimension. Why is that? As it is shown in eq. 2, the gradient of g with respect to y is constant, which means that the behavior of g only considering the y-dimension is linear. In other words, no matter with respect to which point along the y-dimension we are studying the rate of change of g, it will always be the same. Thus, when running gradient ascent, the y-dimension does not give us information about the updated point in any region of the landscape, simply because, given some specific learning rate, we already know beforehand how its y-dimension is going to change (knowing the rate of change of a linear function is enough to faithfully predict any future value that the function can take). Again, this is consistent with the results of Figure 2; the performance of gradient ascent dramatically changes *only* along the x-axis.

Regarding hill-climbing, the main thing to highlight is the straight blue line around $x = 0$ mentioned before. This result is obtained due to the mutation distance selected. When evaluating $g(x,y)$ in $x = 0$, no matter the value of y selected, its height will only increase if y increases too, given a mutation distance of 1. Formally, $\forall x \mid x \in (-1,1)\ and\ some\ y \mid y \in (-2,2), g(x,y) \leq g(0,y)$, where the equality holds if and only if the mutated x is equal to 0; that is, if the x value does not change. This explains why there is a straight line around $x = 0$ when the mutation distance is equal to 1, no matter the number of iterations (Figure 5) In fact, running hill-climbing in points of the x-dimension very close to 0 most probably will also converge to the local maximum, *unless* the number of iterations is high enough.

To examine the performance of gradient ascent depending on the learning rate chosen, it is crucial to know the values of the partial derivative of g with respect to x in each range of x values shown in eq 1. I will only focus on the x-dimension, since, as stated above, the rate of change of $g(x,y)$ with respect to y is always constant and does not play a major role in the explanation. For x values in the range of $[-2,0)$, solving the following inequality gives the learning rate for which gradient ascent will converge to the global optimum starting from points with x in that range:

$$\eta * \frac{dg}{dx} - x > \eta$$

Let's find the minimum learning rate that satisfies the inequality above for the two sub-ranges where the value of the partial derivative differs. According to eq. 1, these two ranges in the x-dimension are $[-2,0.5]$, where the partial derivative of g with respect to x is 1; and $(-0.5,0)$, where the value of the partial derivative is 3. However, if the learning rate is lower than $\frac{1}{2}$, then *all* the starting points with $x \in [-2,-0.5]$ will eventually be updated by gradient ascent so that they the x-dimension will land in the second range; that is, in $(-0.5,0)$. Thus, finding the minimum learning rate is requires finding a learning rate less $\frac{1}{2}$ in the previous x-range. The minimum learning rate for which the inequality holds for *every* x value in the range is obtained as follows:

$$\begin{aligned} 3\eta - x &> \eta \\ 3\eta - 0.4\hat{9} &> \eta \\ 2\eta &> 0.4\hat{9} \\ \eta &> \frac{0.4\hat{9}}{2} \approx 0{,}245 \end{aligned}$$

What this inequality shows is what is the largest step required when $x \in (-0.5,0)$ so that no matter the value of x, it will always be possible to avoid either the local maximum (left plot of Figure 6) or oscillating *in aeternum* around it (right plot of Figure 6). Thus, setting the learning rate in the following range guarantees convergence of gradient ascent from *almost* all the points in the landscape:

$$0.245 < \eta < 0.5$$

Given the range of values to set an optimal η for g, for any starting point such that $x \in (0, 0.5)$, there will always be a set of points from which gradient ascent will not converge to the global maximum, which will correspond to $x = \eta$. Therefore, choosing a learning rate in that range ensures us that the only sub-optimal starting points in the landscape will be when $x = 0$ and $x = \eta$. These can be seen in the

As can be seen in Figure 2, the minimum learning rate computed above is consistent with the results obtained. In the four examples that are shown in Figure 2 (right), when η is in the range obtained above, there are only

two values along the x-dimension where gradient ascent does not converge ($x = 0$ and $x = \eta$). However, when η is not in that range, more points along the x-dimension turn out to be sub-optimal.

Regarding hill-climbing, with the default parameters, the results show that, even with a low number of iterations (25), there are some starting points far from the global maximum from which it still converges to the global maximum. Since hill-climbing is a stochastic algorithm, it is *always* possible for it to avoid the local maximum, although it will be very improbable if the mutation distance is not high enough. When the starting point is closer to the global maximum, as there are just a few points in which the function value (i.e., its height) can be greater, it is expected that the algorithm will converge faster than with distance points from the global maximum.

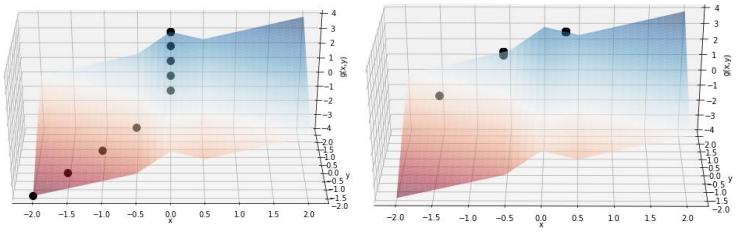

Figure 6. Two sub-optimal scenarios running gradient ascent: converging to the local maximum (left), and oscillating around it (right).

Task 2: Gradient ascent and Hill-climbing with a complex function

The global maximum for the complex function is around 12. There are several local maxima and two main local minima. Computing the gradient for points across every region of the landscape, it can be observed that it is never zero, so we will always expect some movement along what it seems to be a 'flat' surface (note that it could also be a floating point problem, but I will consider that it is not flat in order to account for the results obtained).

- **Question 1**

Using the default parameters in each algorithm and assuming that each possible starting point has the same probability of occurrence, on average, hill-climbing reaches a higher height than gradient ascent (Figure 7, left plot). Since in a complex landscape there are multiple valleys, it is easier for gradient ascent to converge to a local minimum, especially when the learning rate is not low enough. Considering the default parameters, the performance of gradient ascent in the complex landscape is very likely to get stuck either in some local maximum (as in the simple landscape) or in the 'flat' surface of the function. In the simple landscape, there is a way to ensure that from almost all points gradient ascent will converge to the global optimum, while in the complex landscape is harder to avoid any of the local maxima. As can be seen, the number of points that reach the global optimum in gradient ascent is very low compared to hill-climbing (Figure 7, right plot).

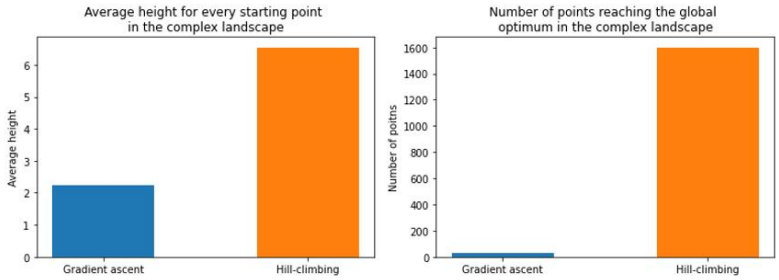

Figure 7.

Thus, although hill-climbing performs a lot better than gradient ascent (reaches, on average, a height three times higher), there are still a lot of points in the landscape from which it does not reach the global maximum, compared to the simple landscape.

- **Question 2**

The results obtained by the systematic analysis suggest that the performance of gradient ascent in a complex function is better if a “conservative” strategy is followed; that is, using a very low learning rate so that the steps that it takes are not very abrupt. Basically, in a complex landscape, taking big steps can derive in a sub-optimal result since it is very likely to land in a lower point with lower height. Figure 8 shows how the performance of hill-climbing is highly influenced by, as in the simple landscape, the number of iterations and the mutation distance. As one of these parameters is increased, the performance gets better and from more points along the landscape it is possible to reach a higher position. Since hill-climbing only moves to higher positions, it is expected i) not to find negative values; ii) high convergence along the landscape as either of the parameters is increased.

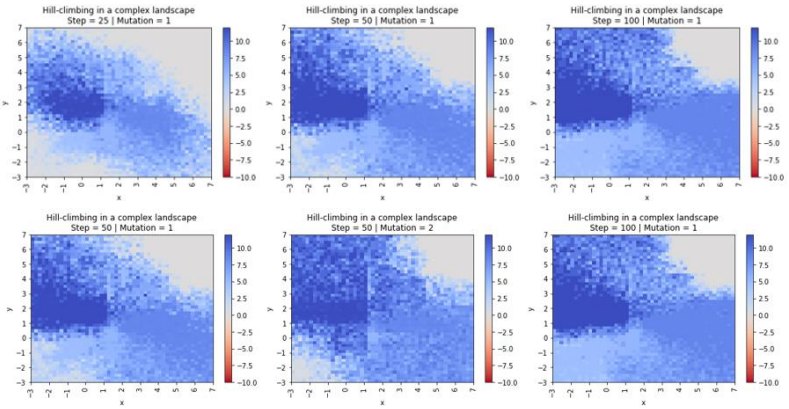

Figure 8.

Regarding gradient ascent, as mentioned above, the best strategy is to set a low learning rate. As shown in Figure 9, when the learning rate is low enough (e.g., 0.01), gradient ascent will converge to high positions for starting points around the cluster of maxima (local and global), and, contrary to the simple landscape, increasing the number of steps does make a difference. In this case, for the same learning rate, less points will end up in a height with negative value. However, if the learning rate is too high (Figure 9, second row), then more points will abandon its initial positions, and convergence to positive height values will decrease as the learning rate increases. Note that, with a sufficient high learning rate, some points might even end up in a position where the height has a considerably negative value. This is the case because the gradient of this function with respect both dimensions fluctuates a lot due to the proximity between maxima and minima, and their respective steepness. Thus, taking big steps will lead to abrupt movements that will decrease the performance of the algorithm.

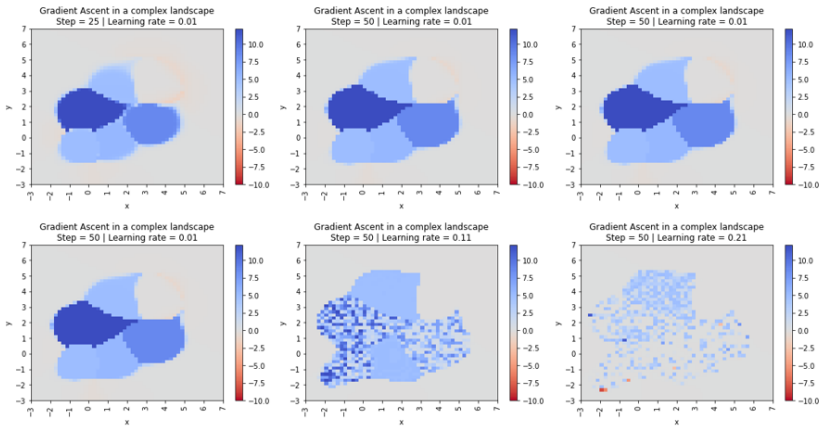

Figure 9.