Christopher Morgan

CS470: AI

Sep 25, 2013

Part 1

1. **Suppose *f*(*x*) = − 3(*x* − 2)2. At what value of *x* does *f*(*x*) take its maximum value?**

Derivative = -6(x-2) \* (1)  
0 = -6(x-2)  
0 = x – 2  
**2 = x**

1. **Suppose that we use gradient ascent (from page 131 and 132 of the book) to find the maximum of the function from problem 1. Let α = 0.1 and suppose that our first guess of the maximum is *x* = 0. What is our guess after one iteration of applying the equation at the top of page 132? What is our guess after two iterations of applying the algorithm?** (Hint: be careful that you carefully plug your answer from the first iteration into the update for the second iteration.)

**X 🡨 X + α**x1 = 0 + 0.1 \* (-6(0-2))  
x1 = **1.2**

x2 = 1.2 + 0.1 \* (-6(1.2 – 2))  
x2 = **1.68**

1. Compute the gradient of *f*(*x*,*y*) = 3*x*2*y*3 − *x*2 + 5*y*2 by computing the partial derivatives \frac{\partial f}{\partial x}  and \frac{\partial f}{\partial y} . (See [this link](http://en.wikipedia.org/wiki/Partial_derivative) for hints.)

Ask **Word** why these are Centered: !?

1. Suppose that we use gradient ascent (from page 131 and 132 of the book) to find the maximum of *f*(*x*,*y*) = − (*x* − 3)2 − 5*y*2 + 2*xy*. Let α = 0.05 and suppose that our first guess of the maximum is *x* = 0,*y* = 1. What is our guess after one iteration of applying the equation at the top of page 132? What is our guess after two iterations of applying the algorithm? (Hint: Notice that this equation is different from the one in the previous problem.)

x1 = [0, 1] + 0.05[(-2(0-3) + 2(1)), -10(1) + 2(0)]

x1 = [0, 1] + 0.05[8, -10]

x1 = [.4, -1.5]

x2 = [.4, 1.95] + 0.05[-2(.4 -3) + 2(1.95), -10(1.95) + 2(.4)]

x2 = [.855, 1.015]

Part 2

The attached [Local Search MATLAB code](https://facwiki.cs.byu.edu/cs470fall2011/index.php/Local_Search_MATLAB_code) has you play with various local search algorithms: hill-climbing, random restart hill-climbing, and beam-forming. The code is organized as follows:

* hill climbing on a surface with one bump
* hill climbing on a surface with more than one bump
* random restart hill climbing on a surface with more than one bump
* local beam search on a surface with more than one bump

Run the code so that you learn how hill-climbing, random restart hill-climbing, and local beam search differ. Then, answer the following questions:

1. What is the probability that hill climbing will reach the maximum point on a surface with more than one bump? How does this probability depend on the footprint of the hill?

With the following assumptions:   
1) The hill climbing algorithm starts at a random location.   
2) Each bump has an equal chance of a starting point be selected on it.

Then the probability of a hill-climbing algorithm finding the true max is 1/(number of bumps).

This probability depends on the footprint of each hill being the same because if the footprint of any given hill becomes larger than another, the algorithm will become more likely to select a point on it as its starting point. If any given hill then is more likely to have a starting point on it selected then its maximum point will more frequently be found as the highest point. This would then mean the probability of 1/(number of bumps) for selecting the true highest point is no longer correct.

1. When would you perform random restart hill-climbing rather than hill-climbing? How would you know which to run?

I would perform random restart hill-climbing whenever the data has enough local maxima to merit the extra computation. In other words, I would use random restart hill-climbing over hill-climbing whenever I have the computational power, time, and it very likely I will run into a local maxima that is unacceptable.

The tricky part is that I would need to know something about the search space. I would need to know that there are local maxima in my search space.

1. What would happen if you changed the beam search algorithm so that it took the 10 best solutions instead of the 7 best? Why?

I would find the true max more often. Because the increase in beams would increase the chances of landed a beam on the hill with the highest peak and I would have more beams to keep around during each iteration, which would hopefully help prevent all of our beams being pulled off before the search on the right hill gets high enough.