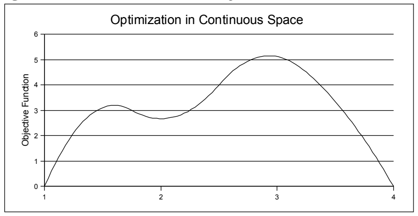
Searching in Continuous Space

Louis Oliphant
oliphant@cs.wisc.edu
cs540 section 2

Environment Types

- Up until now we have been working in discrete environments
  - given a state, there is a clear, finite set of neighbors
- Now we will look at continuous environments
  - No clear, finite set of neighbors
- Still trying to maximize an objective function f(x)

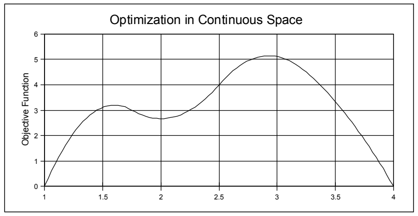


Different Approaches

- Discritize the state space
  - Not much to say about this
- Use the derivative of f(x)
  - solve for f '(x)=0
  - gradient descent
  - Newton's method

Discritizing State Space

- Binning (one method of)
  - For each continuous variable in state space divide its domain into a fixed number of equally spaced bins
  - Use standard discrete search methods



Using the Derivative

- Remember when f '(x)=0 then f(x) is:
  - at a maximum
  - at a minimum
  - at a point of inflection
- Solve f '(x)=0
- So Really we need to find the roots of f '(x)

$$f(x) = -x^4 + (32/3)x^3 - 38x^2 + 48x - 10$$

- Check each x in f(x) for a maximum
  - or check f ''(x) for each x
    - f ''(x)==0 then point of inflection
    - f ''(x)==positive then minimum
    - f ''(x)==negative then maximum

Using the Derivative

- What if you can't solve f '(x)=0 but you can still figure out the derivative?
  - Use Gradient Descent
  - Use Newton's Method

Gradient Descent (Ascent)

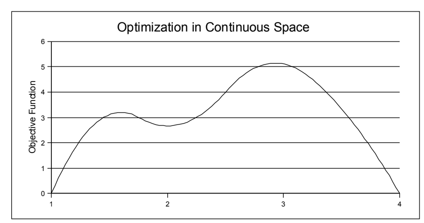
- pick an initial value for x
- pick a step size
- if f '(x) is:
  - positive – move right one step size
  - negative – move left one step size
- Repeat until sign of f '(x) changes or f '(x)=0.
- if f '(x)=0 then return x else return $(x_i+x_{i-1})/2$.

$$f(x) = -x^4 + (32/3)x^3 - 38x^2 + 48x - 10$$

- try it with x=2, step size=0.25

Picking a Step Size

- Step size to small
  - take forever to reach maximum
- Step size to large
  - “step over” the maximum

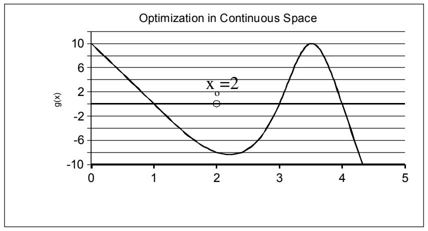


Picking a Step Size

- Use a varying step size
  - Start by using a large step-size and find the range where a maximum exists
  - repeat using a smaller step-size inside that range
  - continue this until you have enough precision
- Alternate method
  - Start by using a small step-size
  - As long as the sign of f '(x) doesn't change double the step size
  - When sign changes drop back to initial small step size and continue search

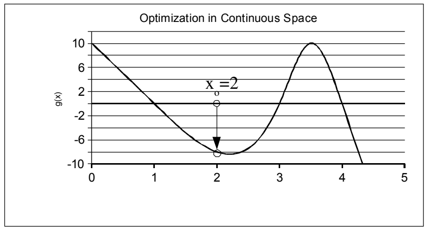
Newton's Method of finding Roots

- We are trying to find the roots of some function g(x).
  - Remember this is the derivative of the Real function that we want to maximize
- Assume g(x) is linear and that you know the value of g(x) and g'(x) for your initial guess, x_0



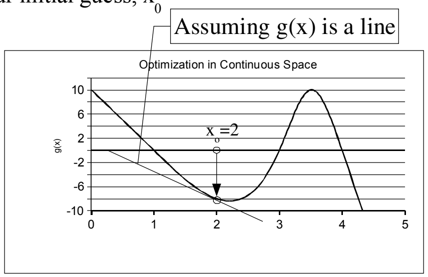
Newton's Method of finding Roots

- We are trying to find the roots of some function g(x).
  - Remember this is the derivative of the Real function that we want to maximize
- Assume g(x) is linear and that you know the value of g(x) and g'(x) for your initial guess, x_0



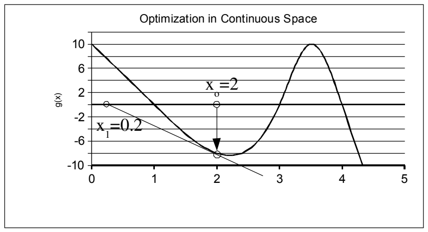
Newton's Method of finding Roots

- We are trying to find the roots of some function g(x).
  - Remember this is the derivative of the Real function that we want to maximize
- Assume g(x) is linear and that you know the value of g(x) and g'(x) for your initial guess, x_0



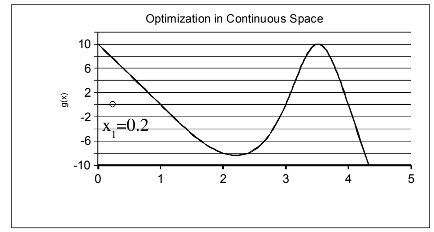
Newton's Method of finding Roots

- We are trying to find the roots of some function g(x).
  - Remember this is the derivative of the Real function that we want to maximize
- Assume g(x) is linear and that you know the value of g(x) and g'(x) for your initial guess, x_0



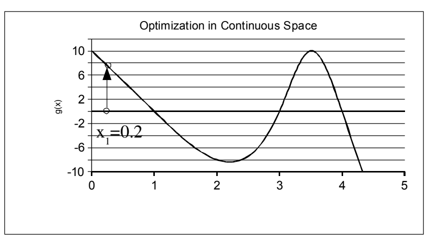
Newton's Method of finding Roots

- We are trying to find the roots of some function g(x).
  - Remember this is the derivative of the Real function that we want to maximize
- Assume g(x) is linear and that you know the value of g(x) and g'(x) for your initial guess, x_0



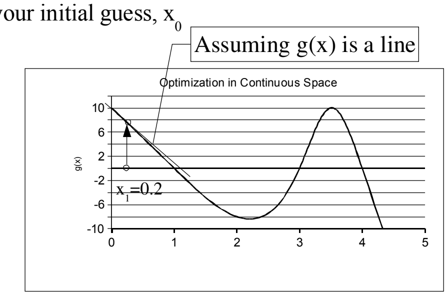
Newton's Method of finding Roots

- We are trying to find the roots of some function g(x).
  - Remember this is the derivative of the Real function that we want to maximize
- Assume g(x) is linear and that you know the value of g(x) and g'(x) for your initial guess, x_0



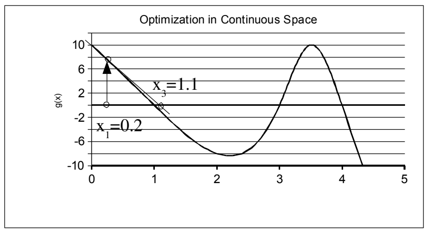
Newton's Method of finding Roots

- We are trying to find the roots of some function g(x).
  - Remember this is the derivative of the Real function that we want to maximize
- Assume g(x) is linear and that you know the value of g(x) and g'(x) for your initial guess, x_0



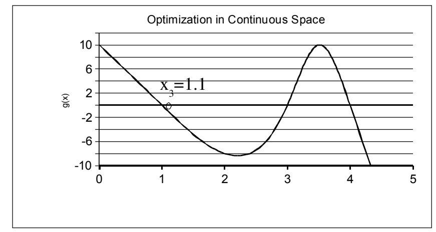
Newton's Method of finding Roots

- We are trying to find the roots of some function g(x).
  - Remember this is the derivative of the Real function that we want to maximize
- Assume g(x) is linear and that you know the value of g(x) and g'(x) for your initial guess, x_0



Newton's Method of finding Roots

- We are trying to find the roots of some function g(x).
  - Remember this is the derivative of the Real function that we want to maximize
- Assume g(x) is linear and that you know the value of g(x) and g'(x) for your initial guess, x_0



Newton's Method of finding Roots

g(x) and g'(x) are given
x_o=initial guess of root
repeat

$$x_{(i+1)} = x_i - g(x_i)/g'(x_i)$$

until little change made in x

Multi-dimensional Environments

- f(x) but now x is a vector
  $x=\{x_1,x_2,\dots,x_n\}$
- Now use the Gradient of f(x) written as ∇f
- ∇f is just a vector of the partial derivatives of f

$$\nabla f = \{\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \dots, \frac{\partial f}{\partial x_n}\}$$

- gradient always points to higher ground

Multi-dimensional Environments

$$f(x, y) = x^2 - 4y^2 - 2xy + 5x - 2y + 3$$

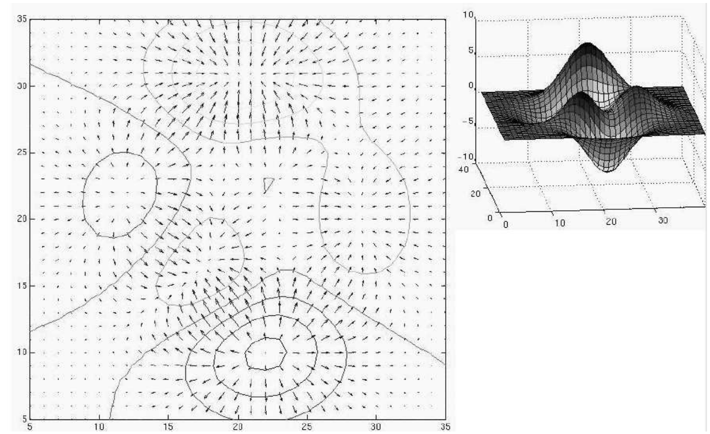
$$\nabla f = \{\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\}$$

$$\nabla f = \{2x - 2y + 5, -8y - 2x - 2\}$$

$$f(2,3) = 2^2 - 4*3^2 - 2*2*3 + 5*2 - 2*3 + 3 = -37$$

$$\nabla f(2,3) = \{2*2 - 2*3 + 5, -8*3 - 2*2 - 2\} = \{3, -30\}$$

Multi-dimensional Environments



Multi-dimensional Environments

- Follow the direction of the gradient
- Take small step sizes until the gradient is zero
- You can also vary the step size just like before

Conclusions

- Binning
- Solve for f '(x)=0
- Gradient descent
- Newton's method
- Understand what the Gradient of a multi-dimensional function is:
  - always points to higher ground

Project Ideas – Ant Colony Optimization

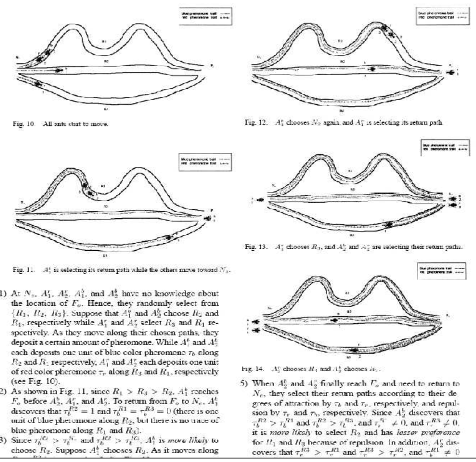


Fig. 10. All ants start to move.

Fig. 12. A_1^b chooses N_e again, and A_1^r is selecting its return path.

Fig. 11. A_1^b is selecting its return path while the others move toward N_2.

Fig. 13. A_1^r chooses R_3, and A_2^b and A_2^r are selecting their return paths.

Fig. 14. A_2^r chooses R_1 and A_2^b chooses R_2.

1) At N_e, A_1^r, A_2^r, A_1^b, and A_2^b have no knowledge about the location of F_o. Hence, they randomly select from $\{R_1, R_2, R_3\}$. Suppose that A_1^b and A_2^b choose R_2 and R_1, respectively while A_1^r and A_2^r select R_3 and R_1 respectively. As they move along their chosen paths, they deposit a certain amount of pheromone. While A_1^b and A_2^b each deposits one unit of blue color pheromone τ_b along R_2 and R_1 respectively, A_1^r and A_2^r each deposits one unit of red color pheromone τ_r along R_3 and R_1, respectively (see Fig. 10).
2) As shown in Fig. 11, since $R_1 > R_3 > R_2$, A_1^b reaches F_o before A_2^b, A_1^r, and A_2^r. To return from F_o to N_e, A_1^b discovers that $\tau_b^{R2} = 1$ and $\tau_b^{R1} = \tau_b^{R3} = 0$ (there is one unit of blue pheromone along R_2, but there is no trace of blue pheromone along R_1 and R_3).
3) Since $\tau_b^{R2} > \tau_b^{R1}$ and $\tau_b^{R2} > \tau_b^{R3}$, A_1^b is *more likely* to choose R_2. Suppose A_1^b chooses R_2. As it moves along

5) When A_2^b and A_2^r finally reach F_o and need to return to N_e, they select their return paths according to their degrees of attraction by τ_b and τ_r, respectively, and repulsion by τ_r and τ_b, respectively. Since A_2^b discovers that $\tau_b^{R2} > \tau_b^{R1}$ and $\tau_b^{R2} > \tau_b^{R3}$, and $\tau_r^{R1} \neq 0$, and $\tau_r^{R3} \neq 0$, it is *more likely* to select R_2 and has *lesser preference* for R_1 and R_3 because of repulsion. In addition, A_2^r discovers that $\tau_r^{R3} > \tau_r^{R1}$ and $\tau_r^{R3} > \tau_r^{R2}$, and $\tau_b^{R1} \neq 0$

Project Ideas – Root Finding Methods