CSE/ECE 848 Introduction to Evolutionary Computation

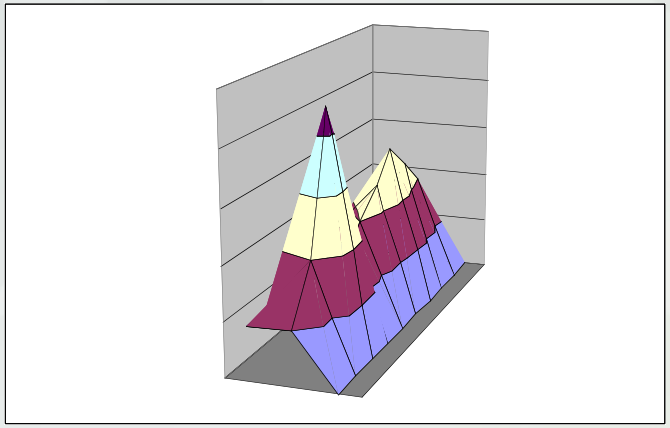
Module 1 - Lecture 4 - Part 2

Back to Search

Wolfgang Banzhaf, CSE
John R. Koza Chair in Genetic Programming

The search space

- Our understanding of search is often aided by the concept of a "search space"
- Each solution is represented in this space as a point whose coordinates are the features
- The "up" direction is the goodness of a solution

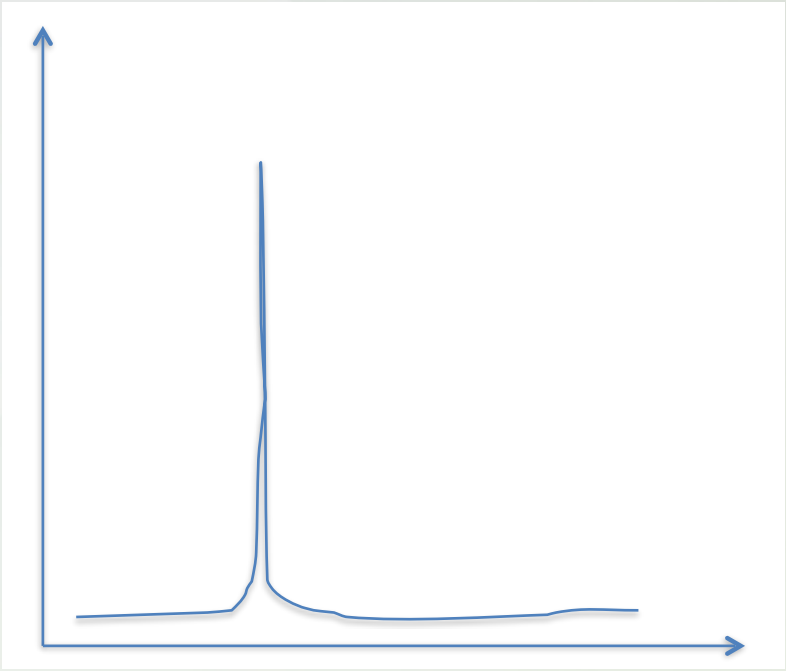


Transforming solutions, search operators

- The evaluation of the entire search space of all candidate solutions is usually impractical
- Each system must define how it will search through a limited portion of such a large space
- Search operators define how an ML system chooses to test, and in what order
- Search operator define and limit the area of representation space that actually will be searched
- A good ML system would use search operators which take a path through solution space that tends to encounter good solutions and bypass bad ones.

Needle in a Haystack Problem

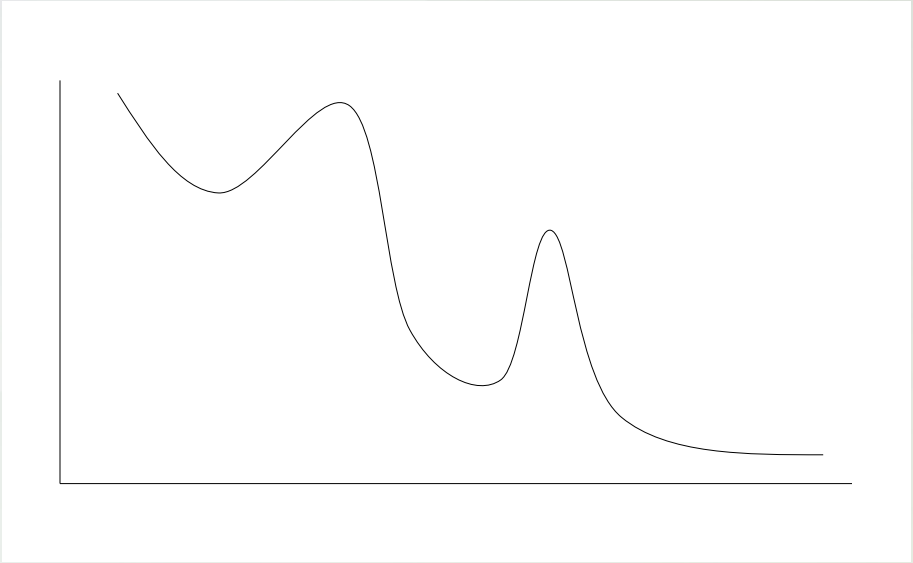
- The “no win” scenario for ML



Topology

- Frequently, ML search is viewed as a movement on a topological surface

- Problem is to avoid local minima (or maxima)

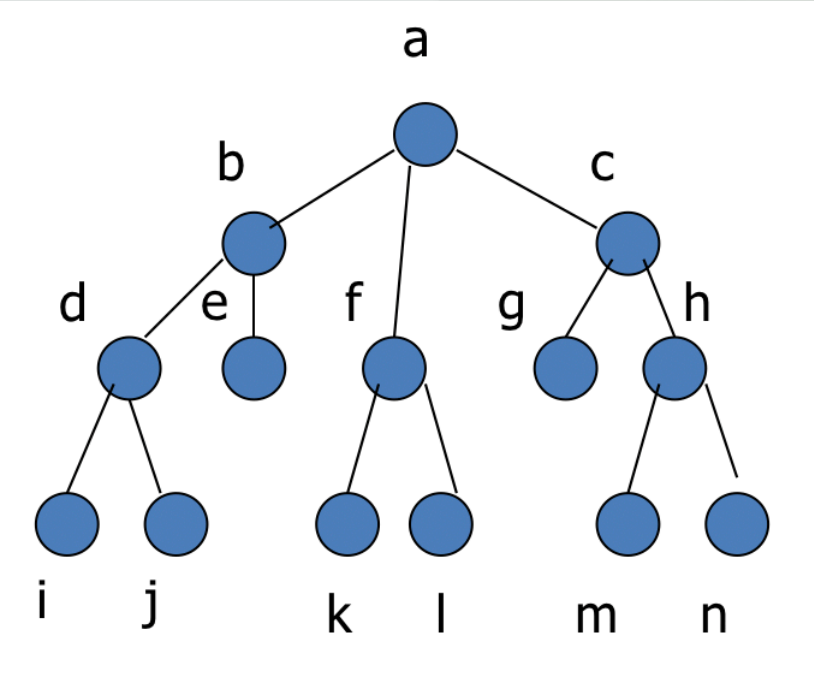


Kinds of search

- Blind search
- Stochastic search
- Informed search
- Heuristic search

Blind search

- Depth-first search
  - abdijefklcghmn
- Breadth-first search
  - abcdefghijklmn



- Breadth-first is guaranteed to find a solution if it exists. Depth-first is not (can loop, requires limits).

Blind search strategy

- Blind search
  - Searching through the solution space and picking a solution using no information about the structure of the problem or results from previous steps in the search
  - Without knowledge of the search space or the benefit of heuristics (rule of thumb) to direct the search

- Blind search, based on structure only
  - Breadth-first or depth-first tree search, iterative deepening
  - Exhaustive search
  - For nearly all interesting learning domains the search space of possible solutions is far too large for exhaustive search to be completed in reasonable time

Hill climbing, Gradient decent

- Hill climbing starts in one spot in the search space, transforms that solution, and keeps the new solution if it is better than old one
  - HC: Any better solution can be accepted
  - GD: Only the best solution is accepted
  - No record is kept of the path that has been traversed already

- Danger of getting stuck in local optimum!

Simulated Annealing

- Simulated annealing (SA) is an approach that addresses the problem of getting caught in local minima (see GD search)
- It has an analogy to cooling of a metal
- Choose a move and traverse to a new state
- If the new state is not as good as the previous one, still accept it, with a probability U based on the difference in quality Δc and a factor termed "temperature" t

$$U < e^{\frac{\Delta c}{t}}$$

- t starts rather high resulting in a rather random search, and is reduced according to a "cooling schedule", making it more and more like gradient decent
- Can achieve good performance in rough landscapes

Informed search: Tabu search

- Informed search methods make use of what has already been done in the past
- Informed search remembers aspects of the search to improve performance
- Tabu search
  - Maintains a short list of moves that cannot be repeated
  - Chooses a move from the present state that is feasible but not on the Tabu list
  - If possible, make that move and add it to the Tabu list

- Trick is to select the right size of the Tabu list, often 7. Too small a list -> cycling; too large a list -> restricts search

Heuristic Search A*

- A* is an informed search method that uses knowledge to reduce the search space of a graph
- A* examines which part of the search space to expand next, based on minimizing a function f(n) = g(n) + h(n), where n is the next state, g(n) is the cost in previous steps, and h(n) is the heuristic function that estimates the cost to the goal state
- If h() does not overestimate the distance, we get guaranteed search results (optimal and complete for local graphs)

Beam Search

- Beam search is a kind of resource-limited best-first (A*) search
  - Maintains a beam_size number of points to explore (the beam)
  - Populates the beam based on the best nodes it has seen to date
  - Does best-first search, but only expands on the nodes in the beam

- Neither optimal nor complete with finite beam_size

Random Search Methods

- We want to keep straight the difference between random search and some of the things we do here
- There is in fact a place for random search in the literature
- Monte Carlo methods were invented to deal with highly coupled systems or systems that have a high degree of uncertainty
  - Generate inputs randomly from the distribution of inputs for the problem
  - Calculate evaluation function using these inputs
  - Repeat (a lot) and aggregate the results

Monte Carlo Tree Search

- MCTS is a method to find the best moves in a game tree representing a game like chess
- It works by expanding the search tree based on random sampling
- MCTS works by multiple “playouts”
- In each playout, the game is played out to the end by selecting moves at random
- The final game result of each playout is then used to weight the nodes in the game tree so that better nodes are more likely to be chosen in future playouts