Hidden Markov Model in Hill Climbing Algorithm with Randomized Search Algorithm

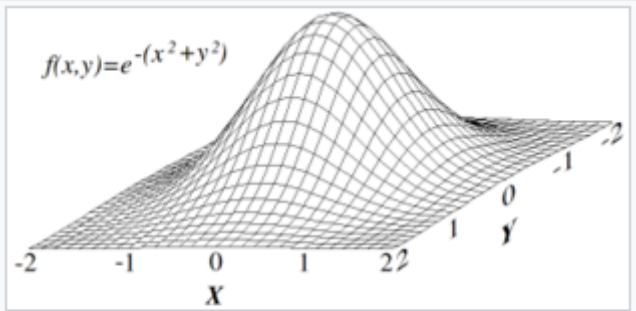
[1] Dr. Suneel Pappala,

[1] Associate Professor,
[1] Information Technology,
[1] Lords Institute of Engineering & Technology, Hyderabad, India.

Abstract - Numerical Analysis, Hill Climbing is a mathematical optimization technique that belongs to the local search family. It is an iterative algorithm that starts with an arbitrary solution to a problem and then tries to find a better solution by making an incremental change to the solution. If the change produces a better solution, another incremental change is made to the new solution, and so on until no further improvements can be found. Hill-climbing algorithm is a local search algorithm that continuously moves upward (increasing) until the best solution is reached. This algorithm terminates when the peak is reached. The state space diagram provides a graphical representation of the states and the optimization function. When the objective function is the y-axis, we try to find the local maximum and the global maximum. Hill climbing is useful for effective operation of robotics. It enhances the coordination of different systems and components in robots.

Index Terms - Hill climbing, Local Maximum, Global Maximum, Ridges, Robotics, Job planning.

Introduction Hill Climbing:
In numerical analysis, hill climbing is a mathematical optimization technique that belongs to the local search family. It is an iterative algorithm that starts with an arbitrary solution to a problem then tries to find a better solution by making an incremental change to the solution. If the change leads to a better solution, another incremental change is made to the new solution, and so on until no further improvements can be found.



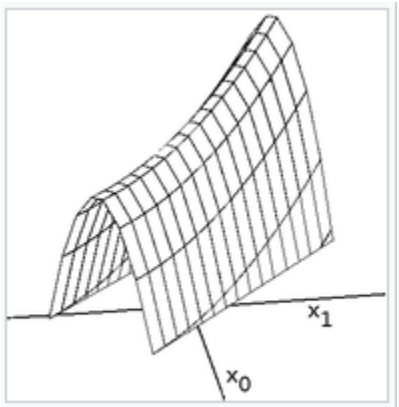
A surface with only one maximum. Hill-climbing techniques are well-suited for optimizing over such surfaces, and will converge to the global maximum.

Mathematical Description:
Hill climbing attempts to maximize (or minimize) an objective function f(x), where x is a vector of continuous and/or discrete values. At each iteration, hill climbing adjusts a single element in x and determines whether the change improves the value of f(x) (this is different from gradient descent methods, where all values in x are adjusted at each iteration according to the gradient of the hill) In hill climbing, any change that improves the value of f(x) is accepted, and the process continues until no more change can be found that improves the value of f(x). Then x is said to be "locally optimal." In discrete vector spaces, each possible value for x can be visualized as a vertex in a graph. In hill climbing, the graph is traced from vertex to vertex, always locally increasing (or decreasing) the value of f(x) locally until a local maximum (or local minimum) x_m is reached.

Ridges and Alleys

Ridges are a challenging problem for hill climbers that optimize in continuous spaces. Because hill climbers only adjust one element in the vector at a time, each step will move in an axis-aligned direction. If the target function creates a narrow ridge that ascends in a non-axis-aligned direction (or if the goal is to minimize, a narrow alley that descends in a non-axis-aligned direction), then the hill climber can only ascend the ridge (or descend the alley) by zig-zagging. If the sides of the ridge (or alley) are very steep, then the hill climber may be forced to take very tiny steps as it zig-zags toward a better position. Thus, it may take an unreasonable length of time for it to ascend the ridge (or descend the alley).

By contrast, gradient descent methods can move in any direction that the ridge or alley may ascend or descend. Hence, gradient descent or the conjugate gradient method is generally preferred over hill climbing when the target function is differentiable. Hill climbers, however, have the advantage of not requiring the target function to be differentiable, so hill climbers may be preferred when the target function is complex.



A ridge

Characteristics of a Hill-Climbing-Algorithm

A hill-climbing algorithm has four main characteristics:

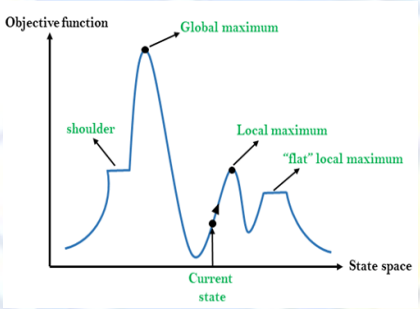
1. It uses a greedy approach, that is, it moves in a direction where the cost function is optimized. The greedy approach allows the algorithm to find local maxima or minima.
2. No backtracking: a hill-climbing algorithm works only with the current state and subsequent states (future). It does not consider the previous states.
3. Feedback mechanism: the algorithm has a feedback mechanism that helps it determine the direction of movement (uphill or downhill). The feedback mechanism is improved by the generation and testing technique.
4. Incremental change: The algorithm improves the current solution by incremental changes.

State-space diagram analysis

A state space diagram provides a graphical representation of states and the optimization function. When the objective function represents the y-axis, it attempts to determine the local maximum and the global maximum.

When the cost function represents the y-axis, the local minimum and global minimum are sought. For more information on local minimum, local maximum, global minimum, and global maximum, click here.

The following diagram shows a simple state space diagram. The objective function is shown on the y-axis, while the state space is the x-axis.



A state space diagram consists of different regions, which can be explained as follows;

- Local Maximum: A local maximum is a solution that outperforms other neighboring solutions or states, but is not the best possible solution.
- Global Maximum: This is the best possible solution that the algorithm achieves.
- Current State: This is the existing or current state.
- Flat Local Maximum: This is a flat area where the neighboring solutions reach the same value.
- Shoulder: This is a plateau whose edge extends upwards.

Problems with hill climbing

There are three areas where a hill-climbing algorithm cannot reach a global maximum or the optimal solution: local maximum, ridge, and plateau.

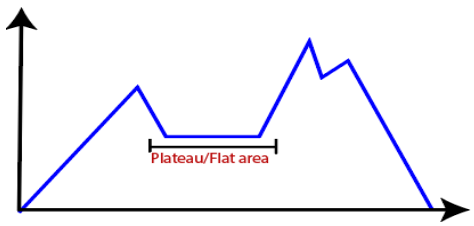
Local Maximum

At this point, the neighboring states have lower values than the current state. The greedy approach function will not move the algorithm to a worse state. This leads to the termination of the hill-climbing process, even though this is not the best possible solution. This problem can be solved with the help of Momentum. In this technique, a certain fraction (m) of the initial weight is added to the current weight. m is a value between 0 and 1. Momentum allows the hill-climbing algorithm to make large steps that take it past the local maximum.

Plateau

In this region, the values reached by the neighboring states are the same. This makes it difficult for the algorithm to choose the best direction.

This challenge can be overcome by making a large jump that takes you to a space without a plateau.



Ridge

The hill-climbing algorithm can terminate itself when it reaches a ridge. This is because the peak of the ridge is followed by a downward movement rather than an upward movement.

This obstacle can be solved by going in different directions at the same time.

Types of hill climbing algorithms

Simple hill climbing

This is a simple form of Hill Climbing in which neighboring solutions are evaluated. If the nearest neighbor state has a higher value than the current state, the algorithm moves. The neighboring state is then set as the current state.

This algorithm is time-saving and requires little computational power. However, the solutions generated by the algorithm are not optimal. In some cases, an optimal solution cannot be guaranteed.

Algorithm

- Perform an evaluation of the current state. Stop the process and indicate success if it is a target state.
- Perform a loop on the current state if the evaluation in step 1 did not yield a target state.
- Continue looping to reach a new solution.
- Evaluate the new solution. If the new state has a higher value than the current state in steps 1 and 2, mark it as the current state.
- Continue with steps 1 to 4 until a target state is reached. If this is the case, the process is terminated.

Steepest - Ascent Climbing

This algorithm is more advanced than the simple hill-climbing algorithm. It selects the next node by evaluating the neighboring nodes. The algorithm moves to the node closest to the optimal or target state.

Algorithm

- Performs an evaluation of the current state. Stops the process and indicates success if it is a target state.
- Performs a loop on the current state if the evaluation in step 1 did not yield a target state.
- Continue looping to reach a new solution.
- A state (X) is set so that the successors of the current state have higher values than it.
- Execute the new operator and generate a new solution.
- Evaluate this solution to determine if it is a target state. If it is, terminate the program. Otherwise, compare it to the state (X).
- If the new state has a higher value than state (X), set it as X. The current state should be set to target if the state (X) has a higher value than the current state.

Stochastic hill climbing

In this algorithm, the neighboring nodes are selected randomly. The selected node is evaluated to determine the degree of improvement. The algorithm moves to this neighboring node if it has a higher value than the current state.

Stochastic hill climbing

In this algorithm, the neighboring nodes are selected randomly. The selected node is evaluated to determine the degree of improvement. The algorithm moves to that neighboring node if it has a higher value than the current state.

Steepest Ascent Climbing:

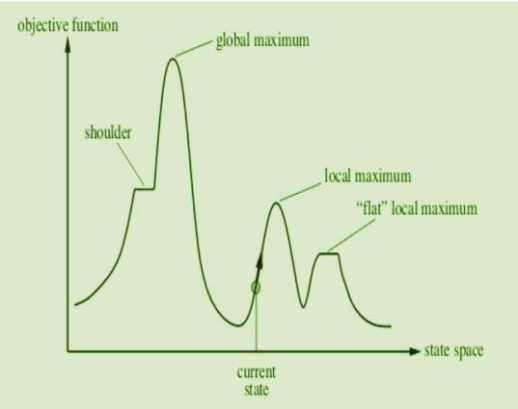
Algorithm for Steepest Ascent Hill Climbing :

- Evaluate the initial state. If it is a target state, the algorithm stops and returns success. Otherwise, the initial state becomes the current state.
- Repeat these steps until a solution is found or the current state does not change anymore
- Choose a state that has not yet been applied to the current state.
- Initialize a new 'best state' that matches the current state and apply it to create a new state.
- Follow these steps to evaluate the new state
- If the current state is a target state, then stop and return success.
- If it is better than the best state, then make it the best state, otherwise continue the loop with another new state.
- Make the best state the current state and go to step 2 of the second point.
- Finish the function.

State Space diagram for Hill Climbing

- X-axis: denotes the state space, i.e., the states or configurations that our algorithm can reach.
- Y-axis: indicates the values of the objective function corresponding to a given state.

The best solution is a state space where the objective function has a maximum value (global maximum).



Different regions in the state space diagram:

- Local maximum: This is a state that is better than its neighboring state, but there is a state that is better than it (global maximum). This state is better because the value of the objective function here is higher than its neighbors.
- Global maximum: this is the best possible state in the state space diagram. The reason is that the objective function has the highest value at this stage.
- Plateau/apartment local maximum: This is a apartment region of the state space where neighboring states have the same value.

- Ridge: It is an area that is higher than its neighbors but has a slope itself. It is a special form of a local maximum.
- Current State: The region of the state space diagram in which we are currently located during the search.
- Shoulder: It is a plateau that has a rising edge.

Applications of hill climbing algorithm

Marketing

A hill-climbing algorithm can help a marketing manager develop the best marketing plans. This algorithm is often used in solving traveling salesman problems. It can help optimize the distance traveled and improve the travel time of sales team members. The algorithm helps to determine the local minima efficiently.

Robotics

Hill climbing is useful for the effective operation of robots. It improves the coordination of various systems and components in robots.

Job planning

The Hill Climbing algorithm has also been applied to job scheduling. This is a process in which system resources are allocated to different tasks within a computer system. Job scheduling is achieved by migrating jobs from one node to a neighboring node. A hill-climbing procedure helps to determine the correct migration route.

Bibliography:

I am Dr Suneel Pappala, Associate Professor, Information Technology, Lords Institute of Engineering & Technology, Hyderabad, Telangana, INDIA. I have 15 years of Experience in Computer Science & Engineering and Information Technology as a Associate Professor.

References:

- Aarts, E., and Korst, J. (1989), Simulated Annealing and Boltzmann Machines - a Stochastic Approach to Combinatorial Optimization and Neural Computers, Wiley, New York. 11
- Ahuja, R. K., Ergun, O., Orlin, J. B., and Punnen, A. P. (2002), "A Survey of Very Large-Scale Neighborhood Search Techniques," Discrete Applied Mathematics, 123, 75- 102.
- B¨ack, T., Fogel, D., and Michalewicz, Z. (1997), Handbook of Evolutionary Computation, IOP Publishing and Oxford University Press, New York, NY.
- Baumert, S., Ghate, A., Kiatsupaibul, S., Shen, Y., Smith, R. L., and Zabinsky, Z. B., Discrete Hit-and-Run for Generating Multivariate Distributions over Arbitrary Finite Subsets of a Lattice, forthcoming in Operations Research, 2009.
- B´elisle, C. J. P. (1992), "Convergence Theorems for a Class of Simulated Annealing Algorithms on R d ," J. Applied Probability, 29, 885-895.
- B´elisle, C. J. P., Romeijn, H. E., and Smith, R. L. (1993), "Hit-and-Run Algorithms for Generating Multivariate Distributions," Mathematics of Operations Research, 18, 255-266.
- Blum, C., and Roli, A., (2003) "Metaheuristics in Combinatorial Optimization: Overview and Conceptual Comparison," ACM Computing Surveys, 35(3), 268-308.
- Boender, C.G.E, and Romeijn, H.E. (1995), "Stochastic Methods," in Handbook of Global Optimization, edited by R. Horst and P. M. Pardalos, Kluwer Academic Publishers, Netherlands, 829-869. [9] Brooks, S. H. (1958), "A Discussion of Random Methods for Seeking Maxima," Operations Research, 6, 244-251.
- Bulger, D., Baritompa, W. P., and Wood, G. R. (2003) "Implementing Pure Adaptive Search with Grover's Quantum Algorithm," Journal of Optimization Theory and Applications, 116, 517-529.
- Bulger, D. W., and Wood, G. R. (1998), "Hesitant Adaptive Search for Global Optimization," Mathematical Programming, 81, 89-102.
- Cohn, H., and Fielding, M. (1999), "Simulated annealing: searching for an optimal temperature schedule," SIAM Journal on Optimization, 9(3), 779-802.
- Davis, L. (1991), Handbook of Genetic Algorithms, Van Nostrand Reinhord, New York.
- De Jong, K. A., Spears, W. M., and Gordon, D.F. (1995), "Using Markov Chains to Analyze GAFOs," in Foundations of Genetic Algorithms 3, edited by D. Whitley, and M. Vose, Morgan Kaufmann, San Francisco, 115-137.
- Del Moral, P. (2004), Feynman-Kac Formulae: Genealogical and Interacting Particle Systems with Applications, Springer, New York.
- Del Moral, P., and Miclo, L. (1999), "On the convergence and applications of generalized simulated annealing," SIAM Journal of Control and Optimization, 37(4), 1222-1250.
- Devore, J. L. (1995), Probability and Statistics for Engineering and the Sciences, Fourth Edition, Wadsworth, Inc. Belmont, CA.
- Dixon, L. C. W., and Szeg¨o, G. P. (1975), Towards Global Optimization, North-Holland, Amsterdam.
- Dixon, L. C. W., and Szeg¨o, G. P. (1978), Towards Global Optimization 2, NorthHolland, Amsterdam.
- Dorigo, M., and St¨utzle, T. (2004), Ant colony optimization, MIT Press, Cambridge, MA.
- Dyer, M. E., and Frieze, A. M. (1991), "Computing the Volume of Convex Bodies: A Case Where Randomness Provably Helps," Proceedings of Symposia in
- Applied Mathematics, 44, 123-169.
- Dyer, M., Frieze, A., and Kannan, R. (1991), "A random polynomial time algorithm for approximating the volume of convex bodies," Journal of the ACM, 38, 1-17.
- Fielding, M. (2000), "Simulated annealing with an optimal fixed temperature," SIAM Journal on Optimization, 11(2), 289-307.
- Glover, F., and Kochenberger, G. A. (2003), Handbook of Metaheuristics, International series in operations research & management science, 57, Kluwer Academic Publishers, Boston.
- Glover, F., and Laguna, M. (1993), "Tabu Search," in Modern Heuristic Techniques for Combinatorial Problems, edited by C. R. Reeves, Halsted Press, New York, 70-150.
- Hajek, B. (1988), "Cooling schedules for optimal annealing," Math. Oper. Res., 13, 311-329.
- Horst, R., and Hoang, T. (1996), Global Optimization: Deterministic Approaches, Third Edition, Springer-Verlag, Berlin.