Learning as Hillclimbing in Weight Space

Andrew G. Barto
Department of Computer Science
University of Massachusetts, Amherst MA 01003

Running Head: Learning as Hillclimbing

Correspondence:
Andrew G. Barto
Computer Science Dept., LGRC
University of Massachusetts
Amherst, MA 01003-4610
Phone: 413 545-2109
Fax: 413 545-1249
email: barto@cs.umass.edu

Learning involves improving performance with experience. Attempts to understand learning and to devise artificial learning systems commonly employ a commonsense improvement strategy known as hillclimbing. Hillclimbing can be pictured as the process of climbing a hill by choosing each step to be one leading uphill from the current location. Following this strategy, the climber continues until there are no more steps that lead uphill. Each possible location corresponds to a possible configuration, or parameter setting, of the learning system, and the altitude of each location corresponds to a 'measure of goodness' of that configuration. The steps available in each location correspond to the simple changes that can be made to the configuration represented by that location.

Hillclimbing is widely employed because it requies so little memory and computational effort. Deciding on each step requires only local information about the landscape and does not require keeping track of the history of past steps and their consequences. It also does not require exhaustive examination of all possible configurations. Of course, a major shortcoming of such a simple procedure is obvious. Just as a mountain climber using this strategy can become stranded on the top of a foothill far below the mountain peak, the hillclimbing strategy can stop at a configuration that, while superior to its immediate neighbors, is far from being the best configuration possible. Hillclimbing stops at configurations that are *locally maximal* in the sense that they are better than, or of equal goodness to, situations in their immediate neighborhood. *Globally maximal* configurations, on the other hand, are not inferior to any possible configuration (Figure 1). Obviously there is an analogous process for climbing *down* to a *local minimum*. While the term *local descent* procedure is sometimes used for this case, the distinction is not fundamental because the two problems exactly mirror one another. Both are *local optimization* procedures for finding configurations that are locally maximal or minimal, as the case may be.

The basic hillclimbing strategy can take many forms depending on how its various elements are defined. For example, what precisely is a 'configuration' and what are 'simple changes'? What kind of knowledge about the landscape is available? One abstract formulation of hillclimbing requires the following components (see, e.g., Papadimitriou and Steiglitz, 1982): 1) a set of *feasible points*; 2) an *objective function*; and 3) a *neighborhood function*. The set of feasible points, also called the *search space*, contains a point representing each configuration that could possibly be a solution of the opti-

Figure 1: Local and Global Maxima and Minima. The bold region of the abscissa contains all possible configurations, or points, x, with $f(x)$ representing the 'goodness' of x.

mization problem. Let us denote the set of feasible points F. The objective function, denoted here by f, assigns a real number to each feasible point:

$$f : F \to \Re.$$

For any feasible point $x \in F$, $f(x)$ is the measure of the goodness of x. A global maximum is any $x \in F$ for which

$$f(x) \geq f(y) \text{ for all } y \in F.$$

The neighborhood function formalizes what we mean by the simple changes that can be made to each configuration. It is a function, denoted here by N, that maps any feasible point x to the set of all feasible points, $N(x)$, that are close to x in the sense that each can be produced from x by some simple change. A local maximum is any $x \in F$ for which

$$f(x) \geq f(y) \text{ for all } y \in N(x).$$

Given these components, the hillclimbing procedure can be expressed using a function that for any feasible point x returns either an improved feasible point or an indication that local improvement is not possible:

$$\texttt{improve}(x) = \begin{cases} \text{any } y \in N(x) & \text{with } f(y) > f(x) \text{ if such a } y \text{ exists} \\ \text{'no'} & \text{otherwise.} \end{cases}$$

Then the hillclimbing procedure is as follows:

```
procedure hillclimb
begin
    x := some initial feasible point;
    while improve(x) ≠ 'no' do
        x := improve(x);
    return x
end
```

Different versions of this hillclimbing procedure exist depending on how the function `improve` selects an improved point y from $N(x)$. If it sequentially searches $N(x)$ for an improved point, returning the first one found, it is called *first improvement* hillclimbing. In contrast, if it returns the best point in $N(x)$, it is called *steepest ascent* hillclimbing (Papadimitriou and Steiglitz, 1982).

Although the procedure given above clearly implements the basic idea of hillclimbing, it applies literally only if the search space consists of a finite number of discrete points because only in this case can one exactly implement the function `improve`. However, the idea of hillclimbing also applies to problems with continuous search spaces, such as the n-dimensional Euclidean space $\Re^n$. When the objective function f is a differentiable function of a region of $\Re^n$, hillclimbing can take the form of a *gradient ascent* method. A point x in $\Re^n$ is an n-dimensional vector $(x_1, x_2, \ldots, x_n)$. The *gradient* of f at x is the vector $(\frac{\partial f}{\partial x_1}(x), \ldots, \frac{\partial f}{\partial x_n}(x))$, where $\frac{\partial f}{\partial x_i}(x)$ is the partial derivative of f with respect the i^{th} dimension evaluated at the point x. This vector, often denoted $\nabla f(x)$, points in the direction of the steepest increase of the objective function at the point x.

In gradient ascent the function `improve` works by moving x some distance in the direction of the gradient vector:

$$\texttt{improve}(x) = x + \eta \nabla f(x), \tag{1}$$

where η is often a nonnegative scalar that, together with the magnitude of $\nabla f(x)$, determines the distance along the gradient direction at which the new point is chosen. When $\nabla f(x) = 0$, x is a local maximum, and hillclimbing stops. The analogous procedure of *gradient descent* applies when the objective function is to be minimized. When a gradient ascent or descent method is applied to a learning system, such as an artificial neural network,

it is common to call η the 'learning rate parameter' because larger steps in the search space often—but not always—produce faster learning.

In fact, the choice of η is critical to the operation of the method: if it is too large, the new point may actually be worse than the old point because `improve` overshot the hill's peak in the gradient direction; if it is too small, on the other hand, hillclimbing progress may be unacceptably slow. More sophisticated versions of gradient ascent select a different η at each step, or even replace multiplication by η by a more complex operation, in order to avoid these difficulties. Many variations of basic gradient ascent have been studied (e.g., Luenberger, 1984). Obviously, to apply a gradient ascent method, it must be possible to compute the gradient of the objective function at each point generated in the search. This is possible in many optimization problems, but in others an approximation of the gradient must be used instead.

The role hillclimbing plays in a learning system depends on what the search space represents and how one defines the objective function. Since learning involves improving aspects of behavior, each point in the search space must somehow represent a possibility for how the system might behave, and the objective function must indicate what it means to achieve varying degrees of behavioral success. A common approach to specifying a search space for a learning system is to devise a mathematical description of the system's behavior that depends on a set of numbers, called *parameters*. As the values of the parameters change, the system's behavior changes.

For example, the connection weights of an artificial neural network are parameters, and learning is said to occur as the weights change according to one of many different learning algorithms, most of which implement a form of hillclimbing. In this case, the search space is the *weight space* consisting of all feasible combinations of weight values. Usually *all* combinations of weight values are feasible, but there are some network learning methods in which some combinations are excluded from the search space (e.g., networks with weight sharing (e.g., Le Cun et al., 1990). The most commonly employed objective functions for artificial neural networks provide measures of the error in a network's behavior as compared to its desired behavior. Networks designed for *supervised learning* (PERCEPTRONS, ADALINES, AND BACKPROPAGATION), for example, usually perform gradient descent on an objective function giving the average over a set of training examples of the squared error between actual and desired network outputs. REINFORCEMENT LEARN-

ING networks, on the other hand, try to maximize an objective function giving (as one example) the probability of reward.

Hillclimbing in continuous search spaces can take forms other than the gradient ascent strategy described above. In fact, one form of hillclimbing that works in continous spaces is practiced by certain bacteria as they "swim" in their continuous fluid surroundings. A bacterium such as *Bacillus subtilis*, for example, propels itself along a straight path by rotating hair-like appendages called flagella. Whenever it reverses the rotational direction of its flagella, the bacterium stops and tumbles in place before heading off in a new, randomly determined, direction. This bacterium tends to move toward higher concentrations of certain chemical substances, called attractants, by regulating the frequency with which it stops and tumbles according to changes in the amount of attractant it senses. When it is moving toward higher attractant concentrations, it reduces the frequency with which it stops and tumbles. Thus, the straight segments of its path that lead uphill tend to be longer than the downhill segments, where the hill is formed by the spatial distribution of the attractant. As a consequence, the bacterium tends to find and remain near a local maximum of the attractant concentration. This behavior, which is a special kind of bacterial chemotaxis known as *klinokinesis,* is discussed by Lackie (1986).

Klinokinesis is a hillclimbing strategy that works somewhat differently from the methods described above because it neither systematically searches the neighborhoods of feasible points nor computes a gradient at each point. The bacterium effectively estimates the slope of the attractant distribution only in the direction it is moving, whereas a gradient estimate would effectively estimate the slope in all directions from each point. In fact, klinokinesis shows that hillclimbing is possible even when it is impossible to consider any alternative situation without actually committing to it. Instead of our mountain climber being able somehow to examine the neighboring terrain in order to determine—before it takes a step—the uphill direction, this mountain climber has to actually move to a point before its altitude can be determined. Although each step under these conditions cannot always produce an improvement, the overall trend of movement stills lead uphill. An example of a strategy like klinokinesis conducted in the weight space of an artificial neural network is provided by Unnikrishnan and Venugopal's (1994) algorithm for adjusting a network's weights.

We have pointed out that hillclimbing is a local optimization procedure

because it finds local instead of global optima. It is sometimes said of a learning system using hillclimbing that it "got stuck" in a local minimum or maximum. Another—and probably more serious—shortcoming of hillclimbing is that it needs a hill to climb! The most natural objective functions for many problems are essentially flat for large regions of the search space. On such 'plateaus', hillclimbing is useless. For example, imagine a search space consisting of all possible lists of some fixed length of cooking instructions and an objective function that rates each such 'recipe' according to how good you think its product tastes. One can imagine using hillclimbing to find a recipe for some delicious treat, but in reality hillclimbing may be practically useless because only very rarely will such a random recipe produce anything edible at all. Hillclimbing works well for improving configurations having neighborhoods full of interesting alternatives, but it is not useful for finding such configurations in the first place. Minsky and Selfridge (1961) discussed this issue with regard to experimentation with randomly connected neural networks and the automatic synthesis of computer programs.

This shortcoming of hillclimbing can be addressed in several ways. First, one can try to define the search space and its neighborhood structure so that plateaus are small and/or few. This requires using prior knowledge about the problem to find a search–space representation that leads to a search space rich in attractive alternatives. In fact, designing appropriate representations is an essential aspect of any attempt to construct a learning system. A second way to address hillclimbing's plateau problem is to use other search methods to find regions of the search space in which hillclimbing can be useful. The *heuristic search* methods of artificial intelligence (AI) (e.g., Pearl, 1984) were developed to work when hillclimbing does not. According to AI terminology, hillclimbing is a *greedy* and *irrevocable* search strategy. It is greedy because it selects the best alternative in the immediate neighborhood of the current configuration without considering the possibility that such a selection may prevent future access to even better alternatives. Hillclimbing is irrevocable because it does not permit attention to return to a configuration considered earlier, even if that configuration showed more promise than the current one. Search strategies that are neither greedy nor irrevocable can be useful when simple hillclimbing is not.

One way to modify hillclimbing so that it is neither greedy nor irrevocable is to sometimes allow moves to configurations that are inferior to the current configuration, i.e., to sometimes allow moves downhill on the

objective function landscape. For example, the SIMULATED ANNEALING algorithm does this using a probabilistic rule for selecting configurations (although simulated annealing is usually presented as a descent procedure). A computational temperature, T, determines how likely an inferior configuration will be accepted as the next configuration. When $T = 0$, the algorithm reverts to a greedy, irrevocable form of hillclimbing that always rejects moving to an inferior configuration.

Finally, methods exist for changing the objective function itself as a result of the learning system's experiences to make it more informative in evaluating alternatives. These methods have reached a high state of development in the field of REINFORCEMENT LEARNING, where they are sometimes called *adaptive critic* methods.

REFERENCES

Lackie, J.M., 1986, *Cell Movement and Cell Behavior*, London: Allen & Unwin.

Le Cun, Y., Boser, B., Denker, J.S., Henderson, D., Howard, R.E., Hubbard, W., and Jackel, L.D., 1990, Handwritten digit recognition with a back-propagation network, in *Advances in Neural Information Processing Systems 2*, (D.S. Touretzky, Ed.), San Mateo CA: Morgan Kaufmann, pp.396-404.

* Luenberger, D.G., 1984, *Linear and Nonlinear Programming* (second edition), Reading, MA: Addison-Wesley.

* Pearl, J., 1984, *Heuristics: Intelligent Search Strategies for Computer Problem Solving*, Reading MA: Addison-Wesley.

* Papadimitriou, C.H., and Steiglitz, K., 1982, *Combinatorial Optimization: Algorithms and Complexity*, Englewood Cliffs, NJ: Prentice-Hall.

Unnikrishnan, K.P., and Venugopal,K.P., 1994, Alopex: A correlation-based learning algorithm for feed-forward and recurrent neural networks, *Neural Computation*, 6:469-490.

FIGURE CAPTION

Figure 1. Local and Global Maxima and Minima. The bold region of the abscissa contains all possible configurations, or points, x, with $f(x)$ representing the 'goodness' of x.