Topic 5: Interdisciplinary Problems and Python Scripting

Lecture 5-1: Forest Fire Models

Friday April 9, 2010

Contents

1 Forest fire model

The forest fire model of Bak, Chen and Tang [Bak-1990] is a simple *probabilistic* cellular automaton with complex behavior that mimics the spreading of fires in a forest.

The forest is modeled as a square region of side L with a regular lattice of L^2 cells or sites which represent trees. A cell can have three states:

1. A living tree, which is colored green.
2. A tree on fire, which is colored yellow.
3. A dead tree, which is colored black.

1.1 Initializing the forest

At time $t = 0$, the forest is populated using two parameters:

- Each cell has a tree with probability p_t, i.e., generate a random number r in the unit interval $0 < r < 1$, and color the site green if $r < p_t$, or black otherwise.
- If the cell has a tree (if $r < p_t$ in the previous step) then set the tree on fire with probability p_f, i.e., generate a uniform deviate r' and color the tree yellow if $r' < p_f$.

1.2 Local update rule

At time t, each cell in the forest will be in one of three states, green, yellow, or black. The forest at time $t+1$ is determined by the following rules:

1. If a tree is green, check its four nearest neighbors. If any neighbor is yellow (on fire), then the tree catches fire and is colored yellow.
2. If a tree is yellow (on fire), then it dies and is colored black.
3. If a tree is dead (black), then a new tree is grown with probability p, i.e., generate a uniform deviate r and color the cell green if $r < p$.

Note that rules 1 and 2 are *deterministic*, but rule 3 is *probabilistic*: this model is therefore a *probabilistic* cellular automaton.

1.3 OpenGL Forest File Program

The program fire.cpp implements the model using OpenGL graphics.

Recall that the dynamics of the model depends on one parameter, namely the probability p for growing new trees from dead stumps. The system is more or less independent of the starting configuration. Bak et al. found the following results for this model:

- The system is characterized by a *correlation length*

$$\xi(p) \sim \frac{1}{p^{\nu}} ,$$

where $\nu \approx 1.0$ in this 2-dimensional model. (Bak et al. also studied the system in 3 dimensions.)

  – If the correlation length is larger than the linear size of the system

$$\xi(p) > L ,$$

  then the fires die out in a time $\sim L$. The growth of of new trees from dead stumps is not large enough to sustain the fires.

  – If the correlation length is smaller than the linear size of the system

$$\xi(p) < L ,$$

  then the fires are sustained: the system reaches a critical steady state in which the growth of new trees feeds the fires with sufficient fuel to keep them burning for ever.

- The distribution of fires in the critical state is a *fractal*, i.e., an object with fractional dimension. In the 2-dimensional model, they found that the fractal dimension of the fire fronts $D \simeq 1.0$, i.e., the fronts are approximately linear; but in 3 dimensions, they found $D \simeq 2.5$, i.e., intermediate between a surface ($D = 2$) and a volume ($D = 3$) distribution.

There are many variations on the simple forest fire cellular automaton model, which show interesting behavior.

This Java Applet implements a model with 7 different states and 8 probabilities.

References

[Bak-1990] P. Bak, K. Chen and C. Tang, "A forest-fire model and some thoughts on turbulence", Phys. Lett. A **147**, 297-300 (1990), http://dx.doi.org/10.1016/0375-9601(90)90451-S

[Bak-1987] P. Bak, C. Tang and K. Wiesenfeld, "Self-organized criticality: An explanation of the $1/f$ noise", Phys. Rev. Lett. **59**, 381 (1987), http://doi.org/10.1103/PhysRevLett.59.381.

[Bak-1988] P. Bak, C. Tang and K. Wiesenfeld, "Self-organized criticality", Phys. Rev. A **38**, 364 (1988), http://doi.org/10.1103/PhysRevA.38.364.

[W-CA] Wikipedia: Cellular automaton, http://en.wikipedia.org/wiki/Cellular_automaton.