Implementing Bak's Sand Pile Model and Schelling's Segregation Model as Cellular Automata

Leigh Tesfatsion

15 January 2007
Econ 308

Outline of Steps

- Implementations as two-dimensional "cellular automata" (checkerboard models)

- Each cell a "finite state machine" (a form of if-then behavioral rule)

- Identification of model parameters (treatment factors)

- How do these models attempt to construct actual empirical phenomena?

- Are both models examples of agent-based computation?

- Are these models consistent with empirical data? If not, what can be done?

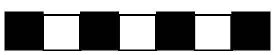
What are Cellular Automata (CAs)?

- **A regular lattice of cells in m dimensions (generally m=1 or m=2)**
- **In each cell, a finite state machine (FSM)**
- **A FSM can be in only one of a finite number of states at any given time.**
- **Transitions between states from one time step to the next for an FSM are governed by a rule (of behavior) in the form of a** *state-transition table*.
- **Given the** *current* **input and the** *current* **internal state of the FSM, the rule specifies the state to be adopted by the FSM at the** *next* **time step.**

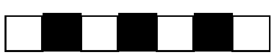
Simple One-Dimensional CA Illustration:

- **CA = Row of six FSMs, each representing rule R**
- **Rule R: Die (turn or stay white) if at least one of your neighbors is dead (white); otherwise turn or stay alive (black).**

Time Step 1:



Time Step 2:



Bak's Sand Pile Model and Self-Organized Criticality (Batten, pp. 10-12, 19-22)

See, also, the 1997 paper by Nathan Winslow (Geological Sciences, University of Michigan) linked to the course syllabus, titled "Introduction to Self-Organized Criticality and Earthquakes." The following notes summarize the discussions in Batten and Winslow.

- **When you first start building a sand pile on a tabletop, the system is weakly interactive. Sand grains drizzled from above onto the center of the sand pile have little effect on sand grains at the edges.**

- **As you keep adding sand grains to the center, a small number at a time, eventually the slope of the sand pile "self organizes" to a critical state where the sand pile cannot grow any larger and breakdowns of all different sizes are possible in response to further drizzlings of sand grains.**

- **Bak refers to this critical state as a state of** *self-organized criticality (SOC)*, **since the sand grains on the surface of the sand pile have self-organized to a point where they are just barely stable.**

What does it mean to say that "breakdowns of all different sizes" can happen at the SOC state?

- **Starting in this state, the addition of one more grain can result in an "avalanche" or "sand slide," i.e., a cascade of sand down the edges of the sand pile and (possibly) off the edge of the table.**

- **The size of this avalanche can range from one grain to catastrophic collapses involving large portions of the sand pile.**

- **The size distribution of these avalanches follows a "Power Law" over any specified period of time T. That is, the average frequency of a given size of avalanche is inversely proportional to some power of its size, so that big avalanches are rare and small avalanches are frequent.**

So what's the formal definition of a "Power Law"?

(See "Notes on Batten Chapter 1," Glossary, p. 13.)

Two variables N and C are said to satisfy a *POWER LAW* relationship if there exist constants K and s such that

$$N = KC^{-s} \quad . \tag{1}$$

Letting n=log(N), k=log(K) and c=log(C), it can be shown that equation (1) implies the linear relationship

$$n \quad = \quad k \; - \; sc \quad . \tag{2}$$

EXAMPLE:

Over 24 hours you might observe one avalanche involving 1000 sand grains, 10 avalanches involving 100 sand grains, and 100 avalanches involving 10 sand grains.

This is consistent with a power law of the form

$$N = 1000 \cdot C^{-1} \quad , \tag{3}$$

where N = number of avalanches and C = number of sand grains involved in the avalanche.

YOU CHECK!!

Winslow (1997) gives an algorithmic description of Bak's sand pile model, summarized below, but no actual code. The following pages translate Winslow's description into pseudo-code that could be fleshed out into an actual working program.

- **A sand pile on a tabletop can be modelled as a two-dimensional "cellular automaton" (checkerboard grid) in which each cell (checkerboard square) keeps numerical track of the "average gradient" G of the sand pile in that cell as successive sand grains are added to the sand pile.**

- **Each cell is assigned a common user-specified** *critical value CV*, **which can be any number greater than or equal to 3.**

- **Starting from some initial distribution of G values across the entire automaton (e.g., all G values set to 0), a cell is initially chosen at random and its G value is increased by one.**

- **If the resulting G value exceeds the critical value for this cell, then this value of G is decreased by 4 and the values of G in the north, south, east, and west neighboring cells of this cell are each increased by 1.**

- **If this redistribution of G values results in a G value in a *neighboring* cell that exceeds its critical value, then another redistribution occurs.**

- **Otherwise, another cell is chosen at random, its G value is increased by 1, and the process repeats.**

- **Winslow shows (Figures 2 and 3) that a log-log plot of the avalanche size C versus the frequency of occurrence N(C) of avalanches of size C obeys a power law distribution, where C is the number of cells whose G value is changed as a result of the avalanche.**

Pseudo-Code for a Sand Pile Model on an 8 Tabletop

```
int main () {
  int TMAX = 1000;
  int X;
  int Y;
  int TestActive;
  AgentInit(); // Construct an array C(X,Y) of
               // 64 agents, X=1,...,8; Y=1,...,8,
               // with initial values A=0,G=0,CV=12,
               // and location indicators X and Y
//If all agents are currently inactive,
//randomly activate an agent
  For (int T = 0; T < TMAX; T++) {
     TestAValues(); // Set TestActive=Max Current A Value
     If (TestActive == 0) {
           X = Rand{1,...,8}; // Randomly select agent
           Y = Rand{1,...,8};
           C(X,Y).Activate(); //Invoke its activation method
     }
  }
  Return 0 ;
}
```

Pseudo-Code for a Sand Pile Agent on an 8× 8 Tabletop

```
class SandPileAgent {
     int A ;          // Active (A=1) or Inactive (A=0) Agent
     int G;           // G = Average gradient value of agent
     int CV;          // CV = Critical value of agent
     int X;           // X-coordinate for the agent
     int Y;           // Y-coordinate for the agent
     Activate();      // Activation method for the agent
     Rule();          // Behavioral rule for the agent
}

void Rule() {
     G = G+1;
     If (G > CV) {       // Does gradient exceed critical value?
          G = G-4;       // If yes, 4 particles roll "down hill"
     // Activate north, east, south, west neighbors
          C(X+1,Y).Activate();
          C(X-1,Y).Activate();
          C(X,Y+1).Activate();
          C(X,Y-1).Activate();
     }
     A = 0;   // De-activate myself
}
void Activate() {
     A = 1; // Activate myself
     Rule(); // Implement my behavioral rule
}
```

This pseudo-code is deficient on several counts.

First, it is logically incomplete. For example, the pseudo-code makes no allowance for agents located at the "edge" of the table who do not have four neighbors.

Second, it is not clear that it captures the empirical aspects of sand piles and sand pile avalanches in an intuitively compelling manner. Consider what you would see, for example, if you graphically visualized the agents in the sand pile model as currently represented by this pseudo-code. Would you see something that "looks like" a real sand pile subject to the type of sand dribble mechanism originally envisioned by Per Bak?

FOR THOSE WITH PROGRAMMING SKILLS AND INTEREST:

- **What about trying your hand at writing complete code for implementing Bak's sand pile model that better captures the empirical attributes of actual sand piles!**
- **Do you think this would be easy to do?**
- **How might you go about it?**

Actually, Winslow (1997) discusses the difficulties that experimenters have had in trying to get *actual* sand piles to behave in the idealized way captured in Per Bak's theory and implemented through simple computer models.

Winslow (1997) cites interesting attempts by Nagel (1992) and Bretz (1992) to conduct experiments with real sand piles.

These researchers were UNABLE to obtain SOC results with ACTUAL sand piles unless the experimental conditions were rather delicately tuned, leading Winslow to question whether actual sand piles can legitimately be said to have *self*-organizing critical states even when critical slope values are found.

Thomas Schelling's City Segregation Model
(Batten, Chapter 1, pp. 12-19)

Schelling's famous city segregation model illustrates how a highly integrated city can rapidly shift to being highly segregated in response to a local disturbance even if people have only a mild preference for living among people similar to themselves.

- In Schelling's city segregation model there are two classes of agents. The agents live in a two-dimensional square "checkerboard" city consisting of sixty-four squares, to be interpreted as a symmetrical grid of house locations.

- Each agent cares about the class of his immediate neighbors, i.e., the occupants of the abutting squares of the chessboard. Each agent has a maximum of eight possible neighbors, the exact number depending on the agent's position on the chessboard (straight edge, corner, or interior).

- Each agent has a "happiness rule" determining whether he is happy or not at his current house location. If unhappy, he either seeks an open square where his happiness rule can be satisfied or he exits the city altogether.

Example of a Happiness Rule:

An agent with only one neighbor will try to move if the neighbor is of a different class than his own; an agent with two neighbors will try to move unless at least one neighbor is of the same class as his own; an agent with from three to five neighbors will try to move unless two neighbors are of the same class as his own; and an agent with from six to eight neighbors will try to move unless at least three neighbors are of the same class as his own.

The exact degree of segregation that emerges in the city depends strongly on the specification of the agents' happiness rules. Batten notes that, under some rule specifications, Schelling's city can transit from a highly integrated state to a highly segregated state in response to a small local disturbance.

For example, if some agent decides to exit the city altogether, his neighbors might become discontented with their current locations and try to move. These efforts can lead to a chain reaction in which increasing numbers of agents become discontented and attempt to move. Thus, subject to some small initial disturbance (e.g., the exit of a few agents from the city), a city initially in a highly integrated state can "tip" to a highly segregated state.