Implementing Per Bak's Sand Pile Model as a Two-Dimensional Cellular Automaton

Leigh Tesfatsion

21 January 2009
Econ 308

Presentation Outline

- Brief review: What is a Cellular Automaton?
- Sand piles and "self-organized criticality"
- Algorithmic description of Per Bak's sand pile model as a two-dimensional cellular automaton (checkerboard model)
- Pseudo-code description of Per Bak's sand pile model (Winslow, 1997)
- Is the resulting CA model consistent with empirical data? If not, what might be done to "fix up" the CA model?

As described in previous notes...

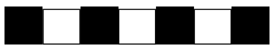
A Cellular Automaton (CA) is:

- **An array 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.**

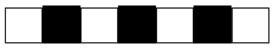
Simple One-Dimensional CA Illustration:

- 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:



Per Bak's Sand Pile Model and Self-Organized Criticality

Basic References:

1. Batten (2000, pp. 10-12, 19-22) (Chapters/entire book posted at the On-Line Econ 308 Syllabus)
2. Nathan Winslow (1997), **"Introduction to Self-Organized Criticality and Earthquakes"**, discussion paper, Department of Geological Sciences, University of Michigan.
   https://www2.econ.iastate.edu/classes/econ308/tesfatsion/SandpileCA.Winslow97.htm

- When you first start building a sand pile on a tabletop, the system is weakly interactive. A sand grain drizzled from above onto a randomly chosen location on the tabletop has little effect on sand grains at other locations.

- However, as you keep dribbling sand grains from above onto randomly chosen tabletop locations, eventually the sand pile at one or more locations reaches a "critical state" where the pile cannot grow any higher without a breakdown of the pile occurring.

- These breakdowns can be of various sizes depending on the exact configuration of the sand pile at the time the breakdown occurs.

- Bak refers to these critical states as states of *self-organized criticality (SOC)*, i.e., states in which the system has self-organized to a point where it is just barely stable.

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

- The dribbling of one more grain of sand onto a location in an SOC state 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.

- Bak conjectured that the size distribution of these avalanches obeys a "Power Law" over any specified period of time T.

- That is, he conjectured that the average frequency of a given size of avalanche is inversely proportional to some power of its size, implying 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 positively-valued variables N and C are said to satisfy a *POWER LAW* relationship if there exist positive constants K and s such that

$$N \quad = \quad KC^{-s} \quad = \quad K\left(\frac{1}{C^s}\right) . \tag{1}$$

Letting n=ln(N), k=ln(K) and c=ln(C), where ln denotes the "natural" (base e) logarithm, it can be shown that equation (4) implies the linear relationship

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

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

$$N \quad = \quad 1000 \cdot \left(\frac{1}{C}\right) \quad , \tag{3}$$

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

YOU CHECK!!

An Algorithmic Description of Bak's Sand Pile Model

Nathan Winslow (1997) gives an algorithmic description of Bak's sand pile model, summarized below, but no actual code.

The following pages translate Winslow's model description first into a verbally expressed "flow diagram" giving the logical flow of activities within the model and then 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).

- 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), SMax grains of sand are dropped on SMax randomly selected cells, which activates the cells.

- Each time a cell B is activated, its G value is increased by one.

- If the new G value for cell B exceeds the critical value for cell B, then the G value for cell B is decreased by 4 and the G values of the four neighbors of cell B (north, east, south, and west) are each increased by 1.

- If the new G value for any of these *neighboring* cells – say cell B* – now exceeds the critical value for cell B*, then the new G value for cell B* is decreased by 4 and the G values of the four neighbors of cell B* are each increased by 1. And so forth and so on.

- Winslow (1997) claims that a plot of the avalanche size C versus the average frequency of occurrence N(C) of avalanches of size C for his sand pile model obeys a power law distribution,

$$N(C) \quad = \quad K\left(\frac{1}{C}\right) \quad . \tag{4}$$

  where C is the number of cells whose G value is changed as a result of the avalanche. See his figures 2 and 3.

Pseudo-Code for Winslow's Sand Pile Model on an 8 × 8 Tabletop

A. The Main Program

```
int main () {
  int SMax; //Number of dribbled grains of sand
  Construct a SandDribbler agent;
  Construct 64 SandPileCell agents on 8 × 8 checkerboard;
  Construct and fill SandPileCell array B[8,8];
  SandDribbler.StartSim(SMax,B);
}
```

B. Pseudo-Code for the SandDribbler Agent

```
class SandDribbler {
     StartSim(SMax,B);   // Start the simulation
}



void StartSim(SMax,B) {
  int U;
  int V;
  //Randomly dribble SMax grains of sand
  //on the sand pile.
  For (int S = 0; S < SMax; S++) {
      U = Rand{1,...,8}; // Randomly select location (U,V)
      V = Rand{1,...,8};
      B[U,V].Activate(); //Activate agent at (U,V)
  }
}
```

C. Pseudo-Code for the SandPileCell Agent

```
class SandPileCell {
    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 rule for the agent
    Update();        // Update rule for the agent
}

void Activate() {
    A = 1; // Activate myself
    Update(); // Implement my update rule
}

void Update() {
    G = G+1;
    If (G > CV) {     // Does my G exceed my CV?
        G = G-4;      // If yes, roll 4 of my grains "down hill"
    // Activate my north, east, south, and west neighbors
        B[X,Y+1].Activate();
        B[X+1,Y].Activate();
        B[X,Y-1].Activate();
        B[X-1,Y].Activate();
    }
    A = 0;  // De-activate myself
}
```

This pseudo-code is deficient on several counts.

- First, it is logically incomplete.

  (a) For example, the pseudo-code makes no allowance for SandPileCell agents located at the “edge” of the table who do not have four neighbors.

  (b) Also, if two SandPileCell agents are activated “at the same time,” and the activation results in two sand grains “rolling down” on a third SandPileCell agent C “at the same time,” the program has to handle the implementation of these multiple effects on C.

  (c) Currently the coding of the Update() function for the SandPileCell agents only handles one rolled-down sand grain at a time, but nothing in the code explains how multiple possible sand-grain effects on C “at the same time” are ordered into a sequence of single-grain effects.

- Second, currently the SMax grains of sand are dribbled “all at once.” presumably one would want to control the time between the dribbling of the SMax grains of sand. For example, one might want to wait until all avalanche activity (if any) resulting from the dribbling of grain S has come to a stop before grain S+1 is dribbled, so that the avalanches do not interfere with each other.

- Third, it is not clear that it captures the empirical aspects of sand piles and sand pile avalanches in an intuitively compelling manner.

  (a) 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.

  (b) 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.