Forest Fire Propagation: A Cellular Automata Approach

Assignment 1b

Probability & Statistics SA 2023-2024

Forest fires are a devastating natural phenomenon, with their scale and frequency demanding urgent attention. Recent data reveals a stark increase in forest fires, causing significant tree cover loss globally. The year 2023 has already witnessed unprecedented fire activity across various regions, notably Canada and Hawaii, showcasing an urgent need to understand and mitigate such disasters.

This assignment introduces a simulation exercise to understand forest fire propagation in a grid-based environment. The model considers factors like wind, moisture, vegetation type, and a small chance of spontaneous ignition to reflect real-world scenarios where fires might start due to lightning strikes or human activities.

The Forest Fire Model simulates a grid where each cell represents a portion of the forest, potentially in one of three states: unburned, burning, or burned. The fire spread depends on the state of neighboring cells, aiming to provide insight into how forest fires propagate and how various factors affect this propagation.

- Adjacent cells to a burning cell may ignite if unburned, with a probability p.
- Cells may also ignite spontaneously with a small probability p_{start}.

The likelihood of a cell igniting due to neighboring cells is given by:

$$P(S_i(t+1) = \text{burning} | S_{i_{\text{neighbors}}}(t) = \text{burning}) = p$$

And the likelihood of spontaneous ignition is:

$$P(S_i(t+1) = \text{burning}) = p_{\text{start}}$$

- A burning cell transitions into a burned cell in the next time step.
- Burned and unburned cells retain their states unless altered by the above conditions.

Assignment

In this assignment, you are tasked with developing simulation for a Forest Fire Model, implementing the model, exploring further modeling approaches, and investigating additional impacting factors. Each section of the assignment will require you to draft pseudocode to outline the logic and steps of the functions to be implemented.

When writing pseudocode:

- Clearly list the input parameters and expected output for each function.
- Write step-by-step instructions in a structured and easily understandable manner.
- Ensure that the logic flows coherently, with each step following logically from the previous one.
- Comment on complex or crucial steps to provide clarity.
- Clearly denote the end of each function.

Following the pseudocode, you will implement the Forest Fire Model as per the instructions provided. Ensure your code is well-commented, compiles, and runs correctly to demonstrate the working model.

1. **(35 points) Design the pseudocode for the forest fire model:**

   (a) **(5 points) Design the `initialize_grid` function:** Develop pseudocode for the `initialize_grid` function that outputs a $m \times n$ grid, where each point represents a portion of the forest. Your map should include at least one water body (Water), a few patches of dry grass (DryGrass), areas with dense trees (DenseTrees), and a few initially burning areas (Burning). Label the rest of the areas as NormalForest. This grid will represent the initial state of the forest before the fire simulation begins.

   (b) **(5 points) Design the `plot_grid` function:** Provide pseudocode to visualize the current state of the grid using different colors for each type of area (e.g., blue for water, green for NormalForest, red for Burning areas, etc.).

   (c) **(5 points) Develop the `neighbours` function:** Create pseudocode to identify neighboring cells of a given point.

   (d) **(5 points) Construct the `propagate` function:** Draft pseudocode to manage fire propagation based on the type of area, its current state, and neighboring cell states.

   (e) **(5 points) Design the `update_grid` function:** Develop pseudocode to manage fire dynamics and grid updates using the `neighbours` and `propagate` functions.

   (f) **(5 points) Draft the `simulate` function:** Provide pseudocode to orchestrate the entire simulation over multiple iterations using the previously developed functions.

   (g) **(5 points) Implement the Forest Fire Model:** Using the pseudocode provided in the above steps as a guide, implement the Forest Fire Model. Your implementation should be able to simulate fire propagation through the grid over multiple iterations. You should provide code, and ensure it compiles and runs correctly to demonstrate the working model. Explain any deviations from the initial pseudocode and the reasons behind such changes.

2. **(30 points)** Regarding the components of the `update_grid` function:

   (a) **(10 points)** Identify and explain the random variables involved in the `update_grid` function. How are these random variables utilized in the simulation?

   (b) **(5 points)** Describe the role of random number generators in the `update_grid` function. How are random numbers generated and utilized in the simulation?

   (c) **(5 points)** Explain the stochastic processes involved in the `update_grid` function. How do these processes influence the behavior of the fire propagation?

   (d) **(10 points)** Does `update_grid` exhibit Markov Chain characteristics? How many Markov chains are involved? Explain.

3. **(35 points) Explore alternative modeling approaches:** Investigate the impact of additional factors:

   (a) **(5 points)** Discuss the potential influence of wind speed and direction on fire propagation. In particular, propose modifications to the transition probabilities to include this factor.

   (b) **(5 points)** Explore the effect of different vegetation types on fire spread. In particular, propose modifications to the transition probabilities to include this factor.

   (c) **(5 points)** Discuss how terrain elevation could affect fire spread. In particular, propose modifications to the transition probabilities to include this factor.

   (d) **(5 points)** Identify and discuss at least one other environmental or external factor that could affect forest fire propagation. In particular propose how this factor modifies the transition probabilities.

(e) **(15 points) Implement the modifications:** Implement the modifications discussed in the previous section to your Forest Fire Model. Ensure your implementation is able to simulate the fire propagation with the incorporated factors and demonstrate the working model. Explain any deviations from the initial ideas and the reasons behind such changes. Your code should compile and run correctly to demonstrate the modified model.

4. **(Bonus: 10 points) Multimedia presentation:** Prepare an animation that shows off all the features of your simulation as a short movie.

Assignment Guidelines

Submission Formats

You can submit your assignment in one of the following formats:

1. **PDF + Code:** A PDF of your written answers, analysis, and results, accompanied by a separate code file (.R or .py).
2. **RMarkdown Report:** If you are working with R, you can submit an RMarkdown file (.Rmd) along with a rendered PDF or HTML. Ensure that the RMarkdown file can be knitted without errors.
3. **Jupyter Notebook:** If you are working in a Python environment, you can submit a Jupyter notebook (.ipynb) along with a rendered PDF version.

Pseudocode Guidelines

Pseudocode is a way to plan your program without the constraints of programming syntax. Here are some guidelines for writing effective pseudocode:

- **Language Agnostic:** Your pseudocode should not be specific to any programming language.
- **Structured Approach:** Use a step-by-step approach, breaking down complex tasks into simpler sub-tasks.
- **Use English:** Write in plain English, avoiding technical jargon.
- **No Syntax:** Do not worry about exact syntax, but maintain a consistent style.
- **Indentation:** Use indentation to show loops, conditionals, and other control structures.
- **Annotations:** You can add brief comments to explain the logic or any assumptions.

Best Practices

- **Documentation:** Comment your code adequately. Your markers should be able to understand your logic just by reading the comments.
- **Function Decomposition:** Break your code into smaller, manageable functions or chunks. Each function should ideally perform one task.
- **Cite Sources:** If you consult external sources or use someone else's code snippets (with modification or directly), make sure to cite them.

Academic Integrity

Remember to maintain the highest standards of academic honesty. While discussions are encouraged, the final work should be your own.