Course: CS30A1570 Complex Systems

Assignment 4: Information, Order and Randomness

Completion Date: 31 March 2024

Submitted by: Turzo Ahsan Sami

**Title:** An Analysis of Information, Order, and Randomness: A Study on Microstates, Macrostates, and Shannon Information Content

**Introduction:** This report presents an in-depth analysis of the concepts of Shannon information content and the concept of microstates and macrostates. These concepts are fundamental to understanding the principles of information theory and statistical mechanics. The study involves a series of tasks that explore these concepts in various contexts, ranging from a simple slot machine example to the more complex logistic map model.

**Research Question:** The primary research question of this study is: How can the concepts of microstates, macrostates, and Shannon information content be applied to various systems to understand their behavior and complexity?

**Results:** The results of the study are derived from 2 tasks each containing a series of sub-tasks, total 11 tasks. In the first task, the first sub-task involved calculating the number of microstates for different scenarios in a slot machine example. The second sub-task explored the concept of entropy in the context of a six-face dice system. The third and fourth sub-tasks calculated the Shannon information content for a six-dice system and a one-year-old baby’s vocabulary, respectively. The fifth sub-task involved creating a real-world example of a message source and calculating its Shannon information content. The sixth sub-task used a NetLogo model to explore the use of information content in distinguishing between different types of text. The seventh sub-task involved running a logistic map model for different values of R and analyzing the resulting information content. In the second task, the first sub-task involved calculatring the average Shannon Information of a 500-word vocabulary, the second sub-task involved adding macrostates in the slot machine example, the third sub-task involved modifying a coin flip netlogo model such that it rolls a six-faced die instead, and the final sub-task is modifying the logistic map model that allows users to set the threshold and analyzing the changes. The work done as part of task completion are stated below:

**Task 1.1:** In the Slot Machine example, calculate the number of microstates corresponding to: a) Exactly two of the same kind b) No lemons c) Two lemons and one orange.

In this case, the slot machine has different kinds of symbols: “lemon”, “orange”, “apple”, “cherry”, and “pear”, therefore N = 5.

1. Exactly two of the same kind: This means two slots are the same and one is different. There are 5 possibilities for the fruit that appears twice, therefore, number of choices for repeated fruit = 5. There are 3 ways to choose 2 reels with same fruit = ₃C₂. For the remaining reel, there are 4 choices left, excluding the chosen fruit, therefore, choices for remaining reel = 4. Therefore, total microstates = 5 \* ₃C₂ \* 4 = 5 \* 3 \* 4 = 60.
2. No lemons: There are 4 fruits left if we exclude lemons. It is required to find the total number of combinations for three different fruits on the reels. Therefore, total microstates = 4 \* 4 \* 4 = 64.
3. Two lemons and one orange: There are 2 specific fruits (lemon and orange) to choose from. For 2 reels to have lemons, there are 3 ways to choose 2 reels = ₃C₂. And the remaining reel must have the orange. Therefore, total microstates = ₃C₂ \* 1 = 3.

**Task 1.2:** Suppose you have six fair dice, each with six sides. The results when you roll all six dice at once is a microstate of the system. E.g., the microstate shown is {6, 6, 6, 6, 6, 6}. a) Using S(Macrostate) = ln W, what is S of the macrostate “each of the six dice shows the same number on its face”? (Note that “ln” is the notation for “natural logarithm”.) b) Same as (a) but for the macrostate “each of the six dice shows a different number on its face”?

The entropy (S) of the macrostates for the dice rolls, S(Macrostate) = ln(W), where W is the number of microstates corresponding to the microstate (Reference 2).

1. Same number on all dice: There are six possibilities for the same number on all dice (1, 2, 3, 4, 5, or 6). For each possibility, all six dice will show that same number. So, there's only one unique microstate for each case. However, for calculating entropy, we consider the total number of possible microstates (W) across all six outcomes. Microstates (W) = 6 (number of possibilities) \* 1 (microstate per possibility) = 6. Therefore, Entropy, S(Macrostate) = ln(W) = ln(6) = 1.8.
2. Different numbers on all dice: Each die can show any of the six numbers, and all six dice must be different. Since the dice results are independent, the options for each die are multiplied, therefore, Total microstates (W) = 6 (possibilities per die) ^ 6 (number of dice) = 6^6 = 46656. Therefore, Entropy S(Macrostate) = S = ln(W) = ln(46656) = 10.75. Here, maximizing the number of unique outcomes (different numbers on all dice) leads to a higher number of microstates and higher entropy. This reflects the greater uncertainty about the specific outcome when all dice show different numbers.

**Task 1.3:** Consider the six-dice system as a “message source” and a roll of the dice as a “message”. If all the dice are fair (equal probability for each value), what is the Shannon information content of this message source?

For the six-dice system, where the dice are fair and independent, the total number of possible outcomes is the total microstates (W) = 6 (possibilities per die) ^ 6 (number of dice) = 6^6. Since the dice are fair, each outcome has an equal probability of p = 1/46656. Therefore, the Shannon information content (Reference 1) of this message source is:

= 15.51

**Task 1.4:** Suppose a one-year-old baby says five different words, each with equal probability. What is the Shannon information content of this one-year-old “message source”?

For the one-year-old baby, there are 5 different words, each with equal probability of p = 1/5. Therefore, the Shannon information content of this message source is:

= 2.32

**Task 1.5:** Come up with a real-world example of a “message source” for which you could calculate Shannon information content. What is the Shannon information content of your example message source?

Let’s imagine an application that translates sounds a six-month-old baby makes into words. A six-month-old baby has 4 emotions: happy, hungry, sleepy, crying, and the baby makes different sound for each emotion. Each emotion occurs with equal probability. Therefore, there are 4 possible messages, each with equal probability of p = 1/4. Therefore, the Shannon information content of this message source is:

= 2

**Task 1.6:** Experiment with TextInformationContent.nlogo. Use the model to see if information content is a good method for distinguishing English text, or any similar task. Describe your task and your results.

For this task, I want to compare the entropy values of verified news articles and news articles generated by Generative AI tools to see if there is a significant difference. The hypothesis is that Generative AI news articles might use more complex and less common words. This will lead to higher entropy values. For this experiment, I have selected two real news articles about upcoming transfer rumors in the European football market. Two more news articles were generated with Generative AI (Attachment 1,2,3,4). The results are:

1. For Real news 1: The Information Content, H = 6.97
2. For Real news 2: The Information Content, H = 7.39
3. For Generative AI news 1: The Information Content, H = 7.47
4. For Generative AI news 2: The Information Content, H = 7.28

The entropy values for both real and AI-generated news articles are very similar. There is no clear distinction between the two categories based solely on entropy. This is a small sample size (only 4 articles). A larger and more diverse set of real and AI-generated news articles might show a clearer trend. Based on this limited experiment, Shannon information content (entropy) doesn't appear to be a reliable method for definitively distinguishing real news from AI-generated news articles. Conducting a larger-scale experiment with a more diverse set of news articles will provide a clear picture. Also, more sophisticated techniques that analyze content structure, factual coherence, and writing style might be needed for better differentiation. Appendix 1 and 2 shows the model with real news. Appendix 3 and 4 shows the model with Generative AI news.

**Task 1.7:** Open LogisticMapInformationContent.nlogo. For each of R = 2.0, 3.1, 3.49, 3.52, 4.0 (five different values), do the following: Set x\_0 to 0.2 Click “go”, and let the model run for about 1000 ticks. For each of these values of R, record the probabilities of 0 and 1, and the final value of information content. Do your own calculation of Shannon information content using these values and see if it agrees with the NetLogo model’s results. Which values of R yields the highest information content, and why? Do you think this information content measure is a good measure of the complexity of the behavior of the logistic map? Why or why not?

After running the LogisticMapInformationContent netlogo model, the following data was obtained:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **R** | **x\_0** | **Probability of 0** | **Probability of 1** | **Information Content (H)** |
| 2.0 | 0.2 | 1 | 0 | 0 |
| 3.1 | 0.2 | 0 | 1 | 0.01 |
| 3.49 | 0.2 | 0.5 | 0.5 | 1 |
| 3.52 | 0.2 | 0.25 | 0.75 | 0.82 |
| 4.0 | 0.2 | 0.49 | 0.51 | 1 |

The Shannon information content (or entropy) of a source is calculated from the probabilities of its possible outputs. The formula is: , where n is the number of possible outcomes and  is the probability of each outcome.

* For R = 2.0, the probabilities are p\_0 = 1 and p\_1 = 0. The Shannon information content:
* For R = 3.1, the probabilities are p\_0 = 0 and p\_1 = 1. The Shannon information content:
* For R = 3.49, the probabilities are p\_0 = 0.5 and p\_1 = 0.5. The Shannon information content:
* For R = 3.52, the probabilities are p\_0 = 0.25 and p\_1 = 0.75. The Shannon information:
* For R = 4.0, the probabilities are p\_0 = 0.49 and p\_1 = 0.51. The Shannon information content:

Therefore, it is evident that the calculated Shannon Information Content agrees with the values obtained from the netlogo model.

The highest information content is yielded by R = 3.49 and R = 4.0. This is because for these values of R, the logistic map exhibits chaotic behavior, leading to a more uniform distribution of states and generates a higher entropy. As the logistic map transitions from periodic to chaotic behavior (with increasing R), the information content increases. The increase of information content shows the difficulty of predicting future values.

The Shannon information content is a good measure of the complexity of the behavior of the logistic map. The Shannon information content measures the average amount of uncertainty in the outcomes. It can measure the unpredictability of the system. Shannon Information content shows the average information required to predict the next value in the sequence. A higher information content indicates a more complex system where predicting the next value is harder. However, it may not capture all aspects of complexity. For example, the presence of patterns in the sequence of states. If a system that alternates between two states in a regular pattern would have a low Shannon information content as the outcomes are very predictable.

Image 1: Logistic Map Information Content netlogo model

A screenshot of a computer

Description automatically generated

**Task 2.1:** Suppose three-year-old Jake has a vocabulary of 500 words (including “um”). When talking, he will say the word “the” one-tenth of the time, the word “um” one sixth of the time, and the rest of the time all his other words will be used equally often. What is the average Shannon information of his side of a conversation?

Shannon information is an approach to calculate the amount of information in a message (Reference 3, 4). Shannon Information ), where pi​ is the probability of each word, and the sum is over all words in the vocabulary.

In Jake’s case, he has a vocabulary of 500 words. He says “the” one-tenth of the time, “um” one-sixth of the time, and the rest of the words are used equally often. Since "the" and "um" take up a fixed amount, the remaining probability is distributed equally among the other words, so, we can calculate the probabilities as follows:

* = probability of the word "the" = 1/10 = 0.1
* = probability of the word "um" = 1/6 = 0.167
* = probability of any other word = (1 – 0.1 – 0.167) / (500 - 2) = 0.0014

Therefore, the Shannon Information of Jake’s words is:

So, the average Shannon information of Jake’s side of a conversation is approximately 7.37 bits. This means that, on average, each word Jake says conveys about 7.37 bits of information.

**Task 2.2:** The Slotmachine netlogo model has been modified to include 3 more macrostates (a) Exactly two of the same kind b) No lemons c) Two lemons and one orange). These additional microstates provide more information about the outcomes of the slot machine pulls (Attachment 5: [modified]\_SlotMachine.nlogo). which are-

1. Exactly two of the same kind: This macrostate checks if exactly two of the reels have the same shape. The “remove-duplicates” primitive is used to create a list of unique shapes from all the reels. If exactly two of the reels have the same shape, then the length of this list will be 2 and the goal-macrostate is incremented by 1, otherwise the non-macrostate is incremented by 1.
2. No lemons: This macrostate checks if none of the reels have the shape “lemon”. The “member?” primitive is used to check if “lemon” is in the list of shapes from all the reels. If “lemon” is in the list, it means there is at least one lemon, so the non-macrostate is incremented by 1. If “lemon” is not in the list, it means there are no lemons, so the goal-macrostate is incremented by 1.
3. Two lemons and one orange: This macrostate checks if exactly two of the reels have the shape “lemon” and one reel has the shape “orange”. The “remove” primitive is used to create a list of shapes from all the reels, excluding “orange”. If exactly two of the reels have the shape “lemon”, then the length of this list will be 2. Similarly, a list of shapes excluding “lemon” is created. If one reel has the shape “orange”, then the length of this list will be 1. If both these conditions are met, the goal-macrostate is incremented by 1, otherwise the non-macrostate is incremented by 1.

Appendix 5,6 and 7 illustrate the modified slot machine netlogo model.

**Task 2.3:** The CoinFlipInformationContent netlogo model is modified, so that, instead of a coin flip, a six-sided die is rolled (Attachment 6: DiceRollInformationContent.nlogo). The code starts by importing the start extension and defining global variables. These include txt for storing the results of die rolls as a string, freq-table and probability-table for storing the frequency and probability of each die roll result, word-count and Max-Word-Count for controlling the maximum number of words to consider, result for storing the result of the current die roll, and die-counts for storing the count of each die roll result. The *setup* procedure initializes the global variables, creates a turtle with the shape of a die, and sets its size and color. The *go* procedure is the main loop of the simulation. It rolls the die, updates the txt string with the roll results, builds the frequency and probability tables based on the words in txt, and sorts the list of words. The *roll-die* procedure simulates the roll of a six-sided die by generating a random number between 1 and 6. The *show-roll* procedure updates the turtle’s label and shape based on the result of the die roll, updates the roll-results string and die-counts list with the roll result, and adds the roll result to the txt string. This netlogo model simulates the roll of a six-sided die and analyzes the results of the rolls. It calculates the frequency and probability of each roll result, calculates the entropy of the results, and displays the results as a string and as a turtle with the shape of a die. The code uses the table extension to store and manipulate the frequency and probability data.

Image 2: Die Rolls Netlogo Model

**A screenshot of a computer

Description automatically generated**

**Task 2.4:** The LogisticMapInformationContent netlogo model has been modified by adding threshold to the model as a slider (Attachment 6: [modified]\_ LogisticMapInformationContent.nlogo). In the update-info-content procedure, the hardcoded 0.5 was replaced with the threshold value. In a BehaviorSpace experiment the threshold value was varied for each of R = 2.0, 3.1, 3.49, 3.52, 4.0. The experiments result (Attachment 7) analysis shows the relationship between three variables: R, threshold, and one-prob. For a given R value, as the threshold increases, the probability of 1 generally decreases. For R=2, the probability of 1  is 1 or close to 1 until a threshold of 0.5, where it drops to 0. For R values of 3.1, 3.49, and 3.52, probability of 1 remains high until around a threshold of 0.5-0.6 and then begins to decrease, but not as drastically as R=2. At R=4, probability of 1 starts at a lower value and decreases more gradually across increasing thresholds. The observations suggest that the probability of 1 value is sensitive to changes in both R and threshold. The observation results are displayed in Appendix 8, 9, 10, 11 and 12.

Image 3: Modified Logistic Map Information Content

A screenshot of a computer

Description automatically generated

**Discussion:** The results of the tasks demonstrated the applicability of the concepts of microstates, macrostates, and Shannon information content in various contexts. The study found that the Shannon information content is a useful measure of the complexity of a system, as it quantifies the average amount of uncertainty in the outcomes. However, it may not capture all aspects of complexity, such as the presence of patterns in the sequence of states.

**Conclusion:** In conclusion, the concepts of microstates, macrostates, and Shannon information content provide valuable insights into the behavior and complexity of various systems. While these concepts have broad applicability, their effectiveness may vary depending on the specific characteristics of the system under consideration. Further research could explore other measures of complexity and how they compare to the Shannon information content.

**References**

1. <https://en.wikipedia.org/wiki/Information_content>
2. <https://courses.lumenlearning.com/suny-physics/chapter/15-7-statistical-interpretation-of-entropy-and-the-second-law-of-thermodynamics-the-underlying-explanation/>
3. <https://machinelearningmastery.com/what-is-information-entropy/#:~:text=We%20can%20calculate%20the%20amount,log(%20p(x)%20)>
4. <https://www.britannica.com/science/information-theory/Entropy>

**Attachments**

1. https://github.com/ahsan-sami-turzo/complex-system-code/blob/main/CS\_Assignment\_4/real\_news\_1.txt
2. https://github.com/ahsan-sami-turzo/complex-system-code/blob/main/CS\_Assignment\_4/real\_news\_2.txt
3. https://github.com/ahsan-sami-turzo/complex-system-code/blob/main/CS\_Assignment\_4/fake\_news\_1.txt
4. https://github.com/ahsan-sami-turzo/complex-system-code/blob/main/CS\_Assignment\_4/fake\_news\_2.txt
5. <https://github.com/ahsan-sami-turzo/complex-system-code/blob/main/CS_Assignment_4/%5Bmodified%5D_SlotMachine.nlogo>
6. <https://github.com/ahsan-sami-turzo/complex-system-code/blob/main/CS_Assignment_4/DiceRollInformationContent.nlogo>
7. <https://github.com/ahsan-sami-turzo/complex-system-code/blob/main/CS_Assignment_4/%5Bmodified%5D_LogisticMapInformationContent.nlogo>
8. <https://github.com/ahsan-sami-turzo/complex-system-code/blob/main/CS_Assignment_4/%5Bmodified%5D_LogisticMapInformationContent%20experiment-spreadsheet.csv>

**Appendices**

Image 1: the model with real news

**A screenshot of a computer

Description automatically generated**

Image 2: the model with real news

**A screenshot of a computer

Description automatically generated**

Image 3: the model with Generative AI news

**A screenshot of a computer

Description automatically generated**

Image 4: the model with Generative AI news

**A screenshot of a computer screen

Description automatically generated**

Image 5: Modified Slot machine model with “Exactly two of the same kind" macrostate

**A screenshot of a computer

Description automatically generated**

Image 6: Modified Slot machine model with “No lemons” machostate

**A screenshot of a computer

Description automatically generated**

Image 7: Modified Slot machine model with Two lemons and one orange” macrostate

**A screenshot of a computer

Description automatically generated**

Appendix 8: Modified Logistic Map Information Content -Threshold versus Probability of 1 when R = 2

Appendix 9: Modified Logistic Map Information Content -Threshold versus Probability of 1 when R = 3.1

Appendix 10: Modified Logistic Map Information Content -Threshold versus Probability of 1 when R = 3.49

Appendix 11: Modified Logistic Map Information Content -Threshold versus Probability of 1 when R = 3.52

Appendix 12: Modified Logistic Map Information Content -Threshold versus Probability of 1 when R = 4