Unit 7C: Counting, Dec. 6, 2002

In-Class Exercises

Exercise C1

A. Before modern civilization, human beings were still able to count numbers, e.g., using fingers. With 10 fingers, we can conveniently distinguish 10 events. If the 10 events are equally likely, how many bits of information can we represent? [give two integers to limit the range]

B. If the events are not equally likely, would the entropy increase or decrease?

CMSC210 Unit7C 13

Exercise C2

- How many ways the letters in the word ***banana*** can be arranged? Explain.

CMSC210 Unit7C 21

Exercise C3

- A baseball player's batting average is 0.25. Find the following probabilities:
  - Exactly 2 hits in 4 times at bat
  - At least one hit in 4 times at bat

CMSC210 Unit7C 26

Exercise C4

- Do you accept the presented account of "irreversibility of time"? Explain.
- We can see the course materials as a set of (sort of) logical statements. What kind of 'structure' do you have in your mind?

CMSC210 Unit7C 33

Take-Home Exercises

Note: These take-home exercises will *not* be collected. If you want to check your answers with me, either show your work or send me e-mail.

Exercise H1: *Alice in Wonderland*

The letter frequency of the novel *Alice in Wonderland* has been analyzed as follows (high to low): "*etaoihnsrdluwgcymfpbkvqxjz*". There is another document called *Morse Code Practice Book* containing 26 alphabetic characters, which appear to be completely random. But the letter frequency of this book is the same for all 26 characters (i.e., even distribution).

A. Between *Alice in Wonderland* and *Morse Code Practice Book*, which has higher entropy? Explain.

Answer: Morse Code Practice Book. The character sequence in "Alice in Wonderland" follows the English grammar is not completely random. The grammar gives some predictability.

Not so long ago, we needed to use a telephone line to connect to the Internet. The connection speed was abysmal. We used to drive to a nearest Wawa (15 miles away) to buy a cup of coffee while waiting for a transmission of a small file.

One day, we had to send the entire content of *Alice in Wonderland* and *Morse Code Practice Book* over the Internet (under this unbearable condition). In order to speed up the transmission, we thought about using file compression software. We expected that the sizes of the files be reduced (without loosing information) and that the files be sent with less time. Of course, the receiver will need to recover the actual content, but we will let them worry about it.

B. To our surprise, the software was able to compress only one of the two files. Which was successfully compressed? And, why?

Answer: Alice in Wonderland. File compression works for low entropy files, i.e., ones with redundant information. Due to the English grammar, "Alice in Wonderland" has a lot of redundancy and can be compressed, while the completely random "Morse Code Practice Book" cannot.

We also found another document, *Penmanship Practice Book I* (for lowercase alphabetic characters). The letter frequency of this book appears to be even for all 26 characters. However, you see all '*a*'s before '*b*'s, '*b*'s before '*c*', etc.

C. Among the three documents, identify the book with the least entropy and the one with the greatest entropy. Explain.

Answer: The least entropy: Penmanship Practice Book I. The greatest entropy: Morse Code Practice Book. Since "Morse Code Practice Book" is completely random, it has the highest entropy. "Penmanship Practice Book I" has a lot of repetitions (redundancy) and thus has the least entropy. In fact, you can send the complete information of "Penmanship Practice Book I", e.g., as "$a^n b^n c^n \ldots z^n$".

Exercise H2: Winning Strategy

A. What kind of information would you need to compute the probability of winning a Jack Pot (lotto)?

Answer: The number of players, the number of winning tickets, and the number of tickets you bought.

B. [Optional] You learned that your favorite deli sold a winning ticket of a Jack Pot a month ago. Discuss whether you should buy a ticket at this deli.

Note: There is no 'correct' answer. You need to position yourself as a frequentist or a Bayesian and make a corresponding judgment.

Exercise H3: Counting

A. Compute the probability of "four cards" (all four cards of the same number in a hand of five cards) in a poker game.

Answer: $13 \times (52 - 4) / C(52, 5)$

13: numbers from A to K

$(52 - 4)$: one more card other than the four

$C(52, 5)$: all the possibilities of choosing 5 cards

B. Justify the formula for $P(n, r)$, using mathematical induction on n.

Hint: In the induction step, you will need to consider two cases: $r = 1$ and $r \neq 1$.

Answer:

Base case ($n = 1$): $P(1, 1) = 1$

Induction step:
Assume $P(n, r) = n \times (n - 1) \times ... \times (n - r + 1) = n!/(n - r)!$
To show $P(n + 1, r) = (n + 1) \times n \times (n - 1) \times ... \times (n + 1 - r + 1) = (n + 1)!/(n + 1 - r)!$
To compute $P(n + 1, r)$, we need to start from choosing one from $n + 1$ elements.
If $r = 1$, we are done. $P(n + 1, 1) = (n + 1) = (n + 1)!/(n + 1 - 1)!$
If $r \neq 1$, we need to compute $P(n, r - 1)$. By applying the induction hypothesis, $P(n, r - 1) = n \times (n - 1) \times ... \times (n - r) = n!/(n - (r - 1))! = n!/(n - r + 1)!$ Then, $P(n + 1, r) = (n + 1)\, n!/(n - r - 1)! = (n + 1)!/(n - r + 1)! = (n + 1)!/(n + 1 - r)!$

Exercise H4: Final Practice Problems (Unit 7A)

A. Represent the set of valid web addresses as a regular expression. Construct a FSA that can accept such a set. A typical web address has the form:

“**`http://pio.intrasun.tcnj.edu/closinginfo.html`**”

Hint: Distinguish the fixed parts, e.g., **`http://`**, and parts that can be arbitrarily complex.
Note: You may need to make additional assumptions, e.g., excluding complicated patterns such as “**#Snow%20Closing**”.

Answer: Assuming the following conditions:

- The pattern after “http://” can be separated by possibly multiple ‘/’.
- Between ‘/’ and after the last ‘/’, there can be many characters (other than ‘/’) possibly separated by ‘.’
- Σ = the set of all characters – {‘.’, ‘/’}

http://Σ(.Σ)*(/Σ(.Σ)*)*

B. Discuss whether the set of all regular expressions (object language) can be specified by a (i.e., another) regular expression (meta-language).

Answer: It would be impossible to represent all regular expressions using a regular expression because regular expressions may involve matching parentheses, which requires a Context-Free Grammar to represent.

<End>