Practice using SageMath

March 29, 2023

Use the following resources to help you complete this worksheet:

- The polynomial ring tutorial at `https://doc.sagemath.org/html/en/tutorial/tour_polynomial.html`.
- The Coding theory thematic tutorial at `https://doc.sagemath.org/html/en/thematic_tutorials/coding_theory.html`
- The documentation at `https://doc.sagemath.org/html/en/reference/coding/index.html`.
- Your classmates.
- Trevor.
- Google, searching "SageMath" and the thing you are trying to look up.

1. (a) Create the ring $\mathbb{Z}/m$ and name it `R` for some choices of m by calling `R = ZZ.quo(m)`. Choose some numbers and try basic arithmetic $(+, -, \times, \div)$ with them. Do you get the expected answers? What happens if you call `R(n)` with $n > m$?

2. (a) Create a polynomial ring with coefficients in $\mathbb{R}$ in the variable x, and name it `S`.

   (b) Create three polynomials, name them `f, g, h`. If you get an error message, read the bottom line and talk to a neighbor about how you might fix it.

   (c) Try basic arithmetic $(+, -, \times, \div)$. Do you get the expected answers?

   (d) Determine if `f,g,h` are reducible, and if so, what are their factors? Compute their GCDs.

3. Let the source $W = \{w_1, w_2, \ldots, w_7\}$ have word probabilities

$$\left(\frac{2}{5}, \frac{1}{5}, \frac{3}{25}, \frac{2}{25}, \frac{1}{25}, \frac{1}{25}, \frac{3}{25}\right).$$

   Use Sage to create a Huffman code with these word probabilities. (Use `{'w1': p1, ...}` to assign weights to each source word.) Is this the same code you got by hand on the last midterm?

4. (a) Come up with your own linear code. Write down the generating matrix and field here: $\mathbb{F}_p =$

   $G =$

(b) Using Sage, determine whether or not your matrix is full rank, as well as its length, dimension, and minimum distance.

Full rank?	
Length	
Dimension	
Minimum Distance	

(c) Use tab-completion (type `C.<tab>`) to experiment with other methods of the code. See what happens if you use them. (If you get an error message related to Gap, let Trevor know.)

5. Send a message.

(a) Using the linear code with generator matrix *G* above, come up with a code, giving at least 6 distinct codewords. Associate them to English letters, including at least the 6 most common English letters e, a, r, i, o, t (but more is ok too!). Write down your code here:

(b) Write a short English message using letters you have associated to codewords and code your message. (You can just do this by hand.)

(c) Corrupt your message using a `StaticErrorRateChannel` transmission.

(d) Trade messages with a partner and use Sage to decode their (corrupted!) message using `codes.decoders.LinearCodeNearestNeighborDecoder`.

6. We haven't yet discussed Reed–Solomon codes in class yet, but that is OK!

Type `codes.GeneralizedReedSolomonCode?` to learn about GRS codes. What happens if you use two question marks?? Experiment and see how the parameters effect the minimum distance.