

# Exploring Linear Algebra through SageMath Labs

Joint Mathematics Meetings

---

William T. Jamieson\* and Christina Starkey  
Joint Work with Adam Gilbert and Megan Sawyer  
January 11, 2018

\*w.jamieson@snhu.edu

# Introduction

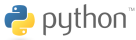
---

## Why create linear algebra labs?

- SNHU to teach out students from Daniel Webster College (9/16).
- DWC had existing four credit linear algebra course for CS.
- Fourth instructional hour necessary for ABET accreditation.
- Labs for Calculus I and II were also created.

## Why SageMath?

- Freeware.
- Inexpensive cloud based delivery (\$9) through Cocalc (formerly SageMathCloud).
- Exposure to/reinforcing Python skills.



# Student Population

- 6 computer science, 2 math students. All CS students were former DWC students.
- All had programming experience; none with Python.
- All were currently enrolled in a linear algebra class.

# The Labs

---

## Goals

1. Introduce computer applications to linear algebra.
2. Reinforce concepts covered during class time\*.
3. Give students exposure to SageMath/Python.

1. *Lights Out*
2. Condition Numbers
3. Computer Graphics
4.  $(7, 4)$ -Hamming Codes
5. The Fibonacci Sequence
6. Google PageRank

Some labs inspired by *Coding the Matrix*



1. Background
2. Mathematics
3. Goals
4. Helpful Sage Commands
5. Problems

# Lab Example

## MAT350: LAB #4 Hamming Codes

SNHU MATHEMATICS DEPARTMENT

### 1. BACKGROUND AND ASSUMPTIONS

In this lab, we will be studying (7, 4)-**Hamming Codes**, which are a particular type of **error correcting code**. When we use WiFi or our cell phones, data (a sequences of 0's and 1's) are transmitted over a **noisy channel**, that is, there is a possibility that some of the 0's and 1's are being sent or read incorrectly. An error correcting code is a method to send the data for which we can detect when an error has been made in the data transmission, and then correct it.

Let's consider the situation where Alice is sending a message to Bob. Using ASCII (American Standard Code for Information Interchange), every letter and punctuation mark has an eight-digit code associated with it. For example, if Alice wants to send the message A to Bob, she would transmit the message 01000001. We will break this eight-digit message into two pieces, each piece containing four digits.

Now we will create a (7, 4)-Hamming Code for the first half of Alice's message:  $x = 0100$ . We will name the  $i^{\text{th}}$  digit of this transmission  $d_i$ . We would like to create a system where we will know when an error in transmission has been made. We will accomplish this with **parity check digits**. Rather than sending just our four-digit message, we will also send an additional three digits  $p_1, p_2,$  and  $p_3$ .



The parity check digits are calculated by adding all of the digits that appear in the same circle as that parity check digit in the Venn diagram above. For example,  $p_1 = d_1 + d_2 + d_4$ . All calculations are done modulo 2. The computation of the parity check digits can be made through a matrix multiplication by the **code generator matrix**  $G$ :

$$G = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 \end{bmatrix}.$$

1

Hamming Codes

MAT350

For example, the (7, 4)-Hamming Code for  $x = 0100$  will be

$$Gx = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 1 \\ 1 \\ 1 \end{bmatrix}.$$

Notice that the first four digits of the code is the message, and the last three are the parity check digits. We will also need the **parity-check matrix**  $H$ :

$$H = \begin{bmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{bmatrix}.$$

Multiplying the matrix  $H$  by  $Gx$  checks that the parity check digits are correct (recall once again that the multiplication is taken modulo 2):

$$\begin{bmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} d_1 \\ d_2 \\ d_3 \\ d_4 \\ p_1 \\ p_2 \\ p_3 \end{bmatrix} = \begin{bmatrix} d_1 + d_2 + d_4 + p_1 \\ d_1 + d_3 + d_4 + p_2 \\ d_2 + d_3 + d_4 + p_3 \end{bmatrix} = \begin{bmatrix} 2p_1 \\ 2p_2 \\ 2p_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}.$$

If  $HGx \neq [0, 0, 0]$ , then we know that there has been an error in the transmission of the message. Further, if only one digit has an error in it (by far the most likely scenario, since we are only transmitting seven digits), then we can identify which digit the error occurred in. For example, let's consider the scenario when an error occurs in the transmission of the fourth digit of the message:

$$\begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 1 \\ 1 \end{bmatrix} \rightarrow \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 1 \end{bmatrix}.$$

If we multiply the parity check matrix by our new message,

$$H \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix},$$

so we know that an error has occurred. Since  $d_4$  appears in every entry of  $HGx$ , we know that the error must have occurred in the fourth digit. The following table identifies the error digit and the corresponding value

Edited by Dr. William T. Jamison

2

Updated: November 7, 2017

# Lab Example

Hamming Codes

MAT350

of  $HGx$ :

| Digit with Error | $HGx$   |
|------------------|---|
| 1                | $\begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}$ |
| 2                | $\begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix}$ |
| 3                | $\begin{bmatrix} 0 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}$ |
| 4                | $\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 0 \end{bmatrix}$ |
| 5                | $\begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 1 & 0 \end{bmatrix}$ |
| 6                | $\begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 1 & 0 \end{bmatrix}$ |
| 7                | $\begin{bmatrix} 0 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 0 \end{bmatrix}$ |

Notice that the values of  $HGx$  correspond to the columns of  $H$ .

## 2. LAB SUMMARY

After reading through this project, you should be prepared to:

- Define the parity-check matrix in Sage
- Identify any errors in the transmission of a message
- Implement the (7, 4)-Hamming Code to automatically error correct messages

## 3. HELPFUL SAGE COMMANDS

In addition to commands defined in previous worksheets, the following commands will be helpful as you complete this assignment.

- `M = MatrixSpace(GF(2), n, m)`  
This command defines a function  $M()$  that creates an  $n \times m$  matrix whose entries are calculated modulo 2. For example, running the commands  $M = MatrixSpace(GF(2), 2, 2)$  and then  $M([[1, 2], [3, 4]])$  returns the matrix  $\begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix}$ .
- `list()`  
This command converts an object into a list. Be sure to convert any objects that you want to flatten into lists first.
- `flatten()`  
This command takes the elements from a list of lists, and creates a single list of those elements. For example, `flatten([[1, 2], [3, 4]])` returns  $[1, 2, 3, 4]$ .
- `str()`  
This command converts an object to a string.
- `str(0) + str(1) = '01'`  
To concatenate two strings, simply add the strings together.
- `[f(i) for i in range(n)]`  
This command returns the list  $\{f(0), f(1), f(2), \dots, f(n-1)\}$ .
- `vector()`  
This command converts a list into a vector.
- `if`  
An `if` statement runs lines of code only if a condition is true. For example,  

```
x = 2
if x < 3:
    print 1
```

Edited by Dr. William T. Jamison

3

Updated: November 7, 2017

Hamming Codes

MAT350

will return 1, but

```
x = 4
if x < 3:
```

```
    print 1
```

will return nothing. Since  $4 < 3$  is false, the `print` command was never run.

- `from sage.crypto.util import bin_to_ascii`  
This command loads the function `bin_to_ascii()`, which converts an eight-digit binary string to its corresponding letter or punctuation mark. For example, `bin_to_ascii('0101100101100101110011')` returns 'Yes'.

## 4. PROBLEMS

### Important Note

To complete Lab 4, rename the Sage Worksheet Lab4.sagews on CoCalc.com with the file naming convention:

`LastName_FirstName_Lab4.sagews`

and perform the following tasks. Utilize appropriate commenting and structure in your worksheet.

Alice has sent you a message with three characters, which is provided in the `hamming_codes_student.sagews` file.**Problem 1.** Load the `bin_to_ascii` function. Define the matrix

$$R = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

and use loops to multiply  $R$  by each seven-digit subset of the message. The result is only the first four digits of each subset (since we don't want to read the parity-check digits as part of the message). To create a string which consists of the digits from Alice's message, first `flatten()` the message, define an empty string `s = ''`, and then use loops to concatenate the digits from the message to `s` one by one. Use the `bin_to_ascii()` function to convert the message from binary to ASCII. Do you believe that there have been any errors in the transmission of the message?**Problem 2.** Define the parity-check matrix  $H$ . Calculate  $Hx$  for each seven-digit subset  $x$  of the message. Be careful that  $x$  needs to be a vector (tuple) in order to perform the multiplication. Which seven-digit subsets of the transmission contain an error?**Problem 3.** Correct any errors that you discovered in Problem 2, and convert the message to ASCII.**Problem 4.** Alice sent you a new message, which is much larger than the original one. The message is provided in the `hamming_codes_student.sagews` file. Convert the message from binary to ASCII.**Problem 5.** Alice's new message is too long for you to fix any errors by hand, so we should create some code that will fix any errors automatically. For example, if  $Hx = [1, 1, 0]$ , we know that the first digit of  $x$  needs to be fixed. We can use an `if` statement:

```
V = VectorSpace(GF(2), 7)
for i in range(len(message)):
    if H * vector(message[i]) == vector([1, 1, 0]):
        message[i] = V(message[i]) + V([1, 0, 0, 0, 0, 0, 0])
```

Edited by Dr. William T. Jamison

4

Updated: November 7, 2017

## Pass/Fail based on Lab average and practicum

| Points | Response                    | Description  |
|--------|-----------------------------|--|
| 0      | Non-submission              | No lab submission  |
| 1      | Non-acceptance              | Lab has few or no correct answers. Lab is disorganized, with little to no explanation or commenting structure.   |
| 2      | Accept with major revisions | Lab has either i) mostly or completely correct answers, but lab is disorganized, with little to no explanation or commenting structure; or ii) lab has incorrect answers or incomplete answers, but lab is well organized with appropriate explanation and commenting structure. |
| 3      | Accept with minor revisions | Lab has mostly or completely correct answers, and is well organized with appropriate explanation and commenting structure.   |
| 4      | Accept with no revisions    | Lab has completely correct answers, and is perfectly organized with appropriate explanation and commenting structure.  |

# Conclusions

---

# Student Feedback

- Students felt that coding helped them understand the course material.
- “The projects and puzzles were fun to figure out.”
- Students wanted a master list of Sage commands.
- Students wanted to see the lab solutions.

# Ideas for Future Improvement

- Discuss problem background and Sage commands together.
- Prepare students to troubleshoot technical issues with CoCalc.
- Clearly define rules for collaboration between students.
- Change the credit structure.
- Have students curate a master list of Sage commands.

Thank you for your time!