Theaching Coding and Information Theory using the pyCoding Python Module

Árpád Horváth
Alba Regia Technical Faculty of Óbuda University,
H-8000 Székesfehérvár, Budai út 45., Hungary
Email: horvath.arpad@amk.uni-obuda.hu

***Abstract*—I have developed a Python module to teach coding and information theory. This module can be used to create a chain of coders, decoders and a channel to investigate the source and channel coding. I have developed a Web page to use the module not only from Python interpreter, but, without any knowledge about Python programming language, by a web browser. Integrating the Pylab module with the pyCoding module makes possible to visualize the results. These modules, the web page and the videos of the web page makes easier for the students to understand the importance of the source and channel coding.**

I. Introduction

There is more courses in our faculty the information and coding theory included in. I have developed earlier a program called pyCodes with which students are able to study communication systems using an interactive Python shell. To learn the basics of the Python language and the interactive shell was acceptable for courses on that we could deal with this topics in a computer laboratory for more occasions. However there are courses with just lessons where the students learn information and coding theory. For this courses I have developed a web surface the students can experimenting on. This web site includes some video tutorials to deepen the knowledge about coding algorithms.

II. Digital communication systems

The goal of a communication system is to transmit a message from one place to the other, from one person to the other, from one computer to the other. These systems include computer networks, broadcasting networks and the telephone networks.

A schematic diagram of the communication systems can be shown on Fig. 1. The starting and ending points of the communication system the information source and the destination of the message. We need to have a transmitter and a receiver to sent the message through a channel[1]. There are two main problems of the digital communication:

- All of the channels add lesser or more noise to the transmitted signal. How can we code the message to correct or a least detect the errors resulted by the noise? The coding with this aim is called *channel coding*.
- The information sources have some known properties such as the distribution of the symbols of the message. How can we code the message to need to transmit the shortest sequence of bits? This is the question of the *source coding*.

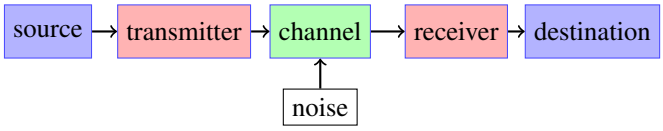

Figure 1. A schematic diagram of a communication systems

class	description
Source	Randomly distributed sources, a symbol to probability mapping.
Code	A symbol to bit sequence mapping.
Hamming	A Hamming coder/decoder.
Channel	A channel with a given number of errors or a given bit error rate.
Chain	A chain of the other elements in this table forming a communication system.

Table I. Main classes of pyCodes

The courses taught by the author include the details of the next codes:

1) Source coding: Huffman code, arithmetic code, and a dictionary coder (LZ78)
2) Channel coding:
   a) error detecting code: parity bit, CRC
   b) error correcting code: Hamming code

III. The Python language and the pyCodes module

The Python language is a general purpose object oriented language with a huge number of standard and third-party modules[2], [3]. Its interactive shell called ipython and the pylab module gives an excellent possibility to the students to experimenting with writing codes and analyzing data.

The pylab module has function names mostly the same as those in MATLAB, and these functions call other functions of program libraries written in C and FORTRAN, so however we are working in Python, the running time of the function is quite short.

The pyCodes module has not extra dependencies except of the Python 3 programming language and its standard module library. It has classes for the elements of the communication systems (Table I).

With the source below, we can create an information source, print its entropy, and draw the distribution of the length of the

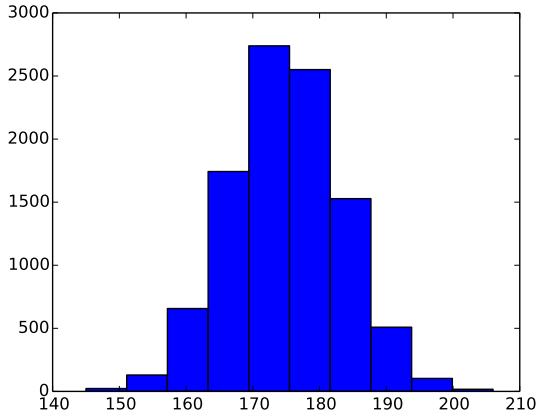

Figure 2. The length histogram of the source coded messages

bit sequences we got after source coding. We provided that the ipython were started with the `--pylab` option to make usage of the plotting functions of pylab easily.

```
from coding import Source, Code
source = Source([1/2, 1/4, 1/8, 1/8])
print(source.entropy())
code =  Code("0 10 110 111")
lengths = [
   len(code.coder(source.message()))
   for i in range(10000)
]
hist(lenghts)
```

The `message` method of the Source class creates 100-symbol-length sequences (100 is the default) with the distribution of the source. The `coder` method of the `Code` class can code the symbol sequence to a bit sequence. (Its `decoder` method would decode from the bit sequence.) With this code we generated 10000 bit sequences. The plotted histogram of the length shows, that this source coding is more efficient, than if we would code each of the 4 symbols with two bits (Fig. 2). The coded length of the codes than would be always 200 bits. With this coding the length of the bit sequences falls near 170 or 180 and there is just tiny probability to be more than 200. We got 1.75 for the entropy, that means we can not get below 175 in average with any coding.

With pyCodes it is quite easy to create a communication system. With the code below we can create one and run it.

```
chain = Chain(
    Source([1/2, 1/4, 1/8, 1/8],
           length=10)
    Code("0 10 110 111"),
    Hamming(8),
    Channel(1)
    )
result = chain.print_run()
```

The system created above consists of:

1) an information source with 4-symbol-source: A, B, C and D, with probabilities 1/2, 1/4, 1/8 and 1/8 respectively, the message length of a run was set to 10 symbols now,
2) and a source coder/decoder (A:0 B:10 C:110 D:111),
3) a Hamming coder/decoder, that the 8 length of blocks of bit sequences translates into 12 length error correction code
4) and a channel with one error in every run.

```
A:0.5 B:0.25 C:0.125 D:0.125
∇ 10 "CCAABBABDA"
Δ 10 "CCAABBABDA"
A:0 B:10 C:110 D:111
∇ 19 "11011000101001011110"
Δ 19 "11011000101001011110"
Hamming(8)
∇ 30 "001010111000111001000101011110"
Δ 30 "001010111000111001000100011110"
Channel("num=1")
```

Figure 3. A result of the print_run method

If we call the `print_run` method of the chain than there will a run printed to the terminal (Figure 3). The output is colored in the terminals capable to use colors. The blue texts are the short description of the element of the communication system. The other rows are the symbol or bit sequences generated in this run. This is a compact view, the sequences of the to direction is next to each other. The direction of the triangles shows the direction. The sequences with downward triangles are generated in the first half of the run, than the others. The differences are highlighted with red.

To install Python with the pylab modules, and to get familiar with it needs some effort. We have used ipython earlier as ipython notebook[4]. This is a nice web interface for the ipython interactive shell. However it is a security whole for the server ipython notebook runs on. On most Linux distribution there is a quite straightforward way to install ipython3 and pylab. pyCodes can be installed from its github repository[5]. We have used a Lubuntu Linux preinstalled with ipython and pylab in VirtualBox® as well. One can install it also on Windows. The easiest way is perhaps to install the Entought Canopy[6].

IV. The web page of coders

Because of the difficulties described in the end of the previous section I developed a web page for investigating sources, codes and communication systems. It is available through the Internet[7].

It is based on the Django web framework[8]. Django uses the model-view-controller pattern, and is written in Python.

The author used the test-driven development[9] process to write the page. Functional tests was written using the Selenium[10] software testing framework, and unit tests with the standard module of Python 3. Using these tests during the initial writing of the functions and the refactorization the web site become more robust.

Szimbólum	D	A	U	I	Y	L	V	O	_	E
Valószínűség	0,01	0,01	0,03	0,04	0,05	0,06	0,09	0,13	0,25	0,33
Információtartalom	6,644 bit	6,644 bit	5,059 bit	4,644 bit	4,322 bit	4,059 bit	3,474 bit	2,943 bit	2,0 bit	1,599 bit
Kód	000101	000100	00011	0000	0100	0101	001	011	10	11

(a)

10	I_LOVE_YOU
32	00001001010110011110010001100011
32	00001001010110011110010101100011
10	I_LOVE_LOU

(b)

10	I_LOVE_YOU
32	00001001010110011110010001100011
48	10010000100111011010100101101101010000011100011
48	10010000100111011010100111011010100000111000011
32	00001001010110011110010001100011
10	I_LOVE_YOU

(c)

Figure 4. The parts of the web pages belonging to the story of Anakin

A. *The story*

A story belongs to the web page to motivate the students. In this section we describe the story, and reference the figures of the web pages when appropriate. The web page is just in Hungarian language yet, but translation with the gettext library is relatively easy. This library have been used extensively in the internationalization of programs of the Linux distributions.

Anakin Skywalker and Obi wan Kenobi must go to a long trip, and they need to work out, how to communicate with the Jedi Council. The use an ancient languages with only ten symbols. Anakin go to the library of the Jedi Order, and makes a statistics about the appearance of the symbols in the texts written on this book. He as padawan learned how to code the symbols to be the coded text as short as can be. He maps longer codes to the less frequent symbols, and shorter codes to the more frequent symbols (Fig. 4 a). (This was a long time ago in a galaxy far, far away. In Earth we call this method Huffman coding.)

He tells Obi wan, what he have done. Obi wan says him, that the Lords of the Dark Power can change some bits in the message. But they have not the power to alternate more than one bits in each twelve bits in the message. This can lead to transform the message to mean the opposite of its original meaning (Fig. 4 b). And he speaks about the algorithms of the Jedi Masters to defend the messages. (Now we call it Hamming coding.) With this algorithm Anakin are able to make a second step of codings to make sure the receiver gets the senders message intact (Fig. 4 c).

B. *What can we do with the web site?*

On the web site we can choose some sources to investigate. The web page shows the main properties of the source than: self-information of the symbols, entropy, efficiency, redundancy. Then we can choose a code to the choosen source. If we have chosen one, than the server generates 50 sample of 100-symbols-long messages, and writes the length statistics of the source coded message. If we choose to study the communication system, than we can add or remove Hamming-code to the chain; change the source, the source code or the channel. We can set the number of errors or the probability of the errors for the channel. We can give a fix source as well, that send always the same message.

Beside investigating sources and communication systems, we can study arithmetic coding and encoding as well. This page generates exercises. For coding it writes a 10-symbol-long words to take the symbol distribution from, and a shorter word to code. For decoding the exercise gives the 10-symbol-long word for the statistics, the arithmetic code of the word, and the length of the word. The web page does not gives a step-by-step solution yet, but it gives the whole solution when we click on the solution button.

However there is no step-by-step introduction, there are videos in the web page showing how to code or decode using arithmetic coding, Hamming coding, and how to create the source codes eith the Huffman algorithm.

V. Summary and Outlook

pyCodes and pylab modules of the Python 3 programming language provide an easy way to study information sources and communication systems. The installation of the latter module is not straightforward so a web page was developed to make the studiing of the communication systems easier. The students can use the web page to investigate information sources, and they can create their own communication systems to study. With the web page they can practise arithmetic coding and decoding as well.

References

[1] C. E. Shannon, “A mathematical theory of communication,” *The Bell System Technical Journal*, vol. 27, pp. 379–423, 1946.

[2] “Official Python site.” [Online]. Available: http://www.python.org

[3] M. Summerfield, *Programming in Python 3: A Complete Introduction to the Python Language*. Addison-Wesley Professional, 2010.

[4] “About notebook at the web page of IPython.” [Online]. Available: http://ipython.org/notebook.html

[5] “The pyCodes repository at GitHub.” [Online]. Available: https://github.com/horvatha/coding

[6] "The Canopy at the site of Enthought Scientific Computing Solutions." [Online]. Available: https://www.enthought.com/products/canopy

[7] "The pyHuffman site." [Online]. Available: http://django.amk.uni-obuda.hu/django/sources

[8] "The official Django Site." [Online]. Available: https://docs.djangoproject.com

[9] H. Percival, *Test-Driven Development with Python*. O'Reilly Media, 2014.

[10] "Selenium site." [Online]. Available: http://www.seleniumhq.org/