

Paquet de programari per a teoria de la informació

Pol Bonastre i Romera

Resum— La teoria de la informació és una branca de les matemàtiques aplicades, l'enginyeria, les ciències de la computació i les telecomunicacions que estudia la transmissió i el processament de la informació. Impulsada per Claude E. Shannon a finals dels anys 40. Aquest projecte es basa en la implementació de diversos càlculs de la teoria de la informació utilitzant Python i wxPython per facilitar una llibreria de funcions que pugui ser utilitzada amb programes externs capaços de treballar amb classes Python, com ara Sage o MATLAB, i a més, oferir a l'usuari un executable amb una interfície gràfica amigable que li permeti utilitzar les funcions de la llibreria sense necessitat de conèixer Python.

Paraules clau—teoria informació, llibreria de funcions, Python, executable, interfície gràfica

Abstract—Information theory is a branch of applied mathematics, engineering, computer science and telecommunications that studies the transmission and processing of information. Driven by Claude E. Shannon in the late 40s. This project is based on the implementation of various calculations of information theory using Python and wxPython to provide a library of functions that can be used with external programs capable of working with Python classes, such as Sage or MATLAB, and also provide the user an executable with a friendly graphical interface that allows him to use the functions of the library without knowing Python.

Index Terms—information theory, library functions, Python, executable, GUI



1 INTRODUCCIÓ

LA història de la teoria de la informació es remunta a finals de la Segona Guerra Mundial impulsada per Claude E. Shannon [1]. De l'augment de la complexitat i de la massificació de les vies de comunicació, com ara els telèfons, sistemes ràdio i xarxes teletip va sorgir la necessitat de realitzar una base teòrica per a la tecnologia de la comunicació [2].

A partir de la formulació de lleis matemàtiques d'Andrei A. Markovi i posteriorment de Ralph V. L. Hartley, Shannon i Warren Weaver [3] van desenvolupar els principis de la teoria.

Actualment la teoria de la informació s'aplica també a les formes de transmissió i emmagatzematge del present com ara la televisió, els impulsos elèctrics transmesos entre ordinadors, compressió de dades, protocols per a xarxes de comunicacions, criptografia, teoria de codis, mòbils, DVD's, etc.

La motivació personal per dur a terme aquest projecte és la creació d'una llibreria de funcions que contenen càlculs matemàtics, aplicant i expandint els coneixements, tant de la teoria de la informació com de programació, que he adquirit durant el grau.

1.1 Objectius

L'objectiu principal del projecte és la creació d'una llibreria de funcions i d'una interfície d'usuari que actuï de *frontend* de la llibreria. Aquesta llibreria i el programa han de permetre el càlcul dels següents aspectes de la teoria de la informació:

- Informació.
- Entropia.
- Verificació de la desigualtat de Kraft/McMillan.
- Verificació de si un codi és instantani.
- Fita inferior de la longitud mitjana segons el primer teorema de Shannon.
- Eficiència d'un codi.
- Informació mútua d'un canal.
- Capacitat d'un canal si és sense pèrdua, determinista, sense sorolls, entrada i sortida independents o simètric.
- Probabilitat mitjana d'error en la descodificació.

Per assolir l'objectiu principal s'han proposat els següents subobjectius:

1. Facilitar una llibreria amb una part dels continguts donats a l'assignatura d'Informació i Seguretat.
2. Garantir la compatibilitat amb tots els Sistemes Operatius.
3. Oferir una interfície *friendly* per a la utilització de la llibreria.
4. Facilitar una documentació completa de cadascuna de les funcions.
5. Oferir un executable que no requereixi d'instal·lacions complementàries.

Els subobjectius definits ens permetran saber amb més facilitat el grau d'assoliment de l'objectiu principal.

- E-mail de contacte: pol.bonastrei@e-campus.uab.cat
- Menció realitzada: *Tecnologies de la Informació*.
- Treball tutoritzat per: *Joaquim Borges Ayats (dEIC)*
- Curs 2014/15

1.2 Organització del document

Aquest document està dividit en diferents apartats per facilitar la seva lectura. A continuació s'explica el contingut de cadascun dels apartats.

Al següent apartat, descripció dels càlculs, es defineixen i s'expliquen els càlculs que s'inclouen a la llibreria de funcions i al programa.

El tercer apartat, estat de l'art, es presenten les eines existents actuals similars al projecte a realitzar.

Seguidament, al quart apartat s'explica la metodologia utilitzada durant el desenvolupament del projecte, l'elecció del llenguatge de programació, el *framework* GUI utilitzat i la planificació.

Al cinquè apartat, desenvolupament del projecte, es detalla com s'ha dut a terme el projecte, explicant cadascuna de les eines utilitzades.

Al sisè apartat, es mostren els resultats obtinguts, explicant el grau d'assoliment dels objectius. A més, s'inclou un exemple d'importació de la llibreria de funcions en una plataforma matemàtica.

Al setè apartat, es presenten les conclusions extretes de la realització del projecte.

2 DESCRIPCIÓ DELS CÀLCULS

Com s'ha explicat anteriorment, la llibreria de funcions ha d'implementar un seguit de mètodes per realitzar diferents càlculs de la teoria de la informació. A continuació s'expliquen cadascun d'aquests càlculs que formen part del contingut de l'assignatura Informació i Seguretat [4], [5], [6], [7], i es troben documentats al llibre *Information Theory* de Robert B. Ash [8].

2.1 Informació

La informació és la mesura de la incertesa d'un esdeveniment m amb probabilitat $p(m)$ segons la mesura de Shannon.

$$I(m) = \log 1/p(m) = -\log p(m)$$

Quan volem utilitzar el bit com a unitat d'informació, utilitzarem el logaritme en base 2.

Cas que la base sigui 10, la unitat rep el nom de dit i representa el grau d'incertesa corresponent a 10 esdeveniments equiprobables.

Finalment si la base és e , la unitat de mesura rep el nom de nat.

2.2 Entropia

Per explicar el concepte d'entropia, primer és necessari saber què és una font d'informació. Es considera una font d'informació com una variable aleatòria discreta X que pot prendre n valors diferents $S = \{a_1, \dots, a_n\}$ tal que la probabilitat de cada valor a_i és $p_i = p(X = a_i) = p(a_i)$.

L'entropia representa la incertesa o informació mitjana d'una font. Sigui X una variable aleatòria discreta amb distribució de probabilitats $\{p_1, \dots, p_n\}$, ($p_i > 0$, $\sum_{i=1}^n p_i = 1$), l'entropia d' X és $H(X) = -\sum_{i=1}^n p_i \log p_i$.

De la mateixa manera que a la informació, podem tenir diferents unitats d'informació, les quals, depenent de la base del logaritme tindran el nom de bits/missatge, dits/missatge o nats/missatge.

2.3 Desigualtat de Kraft/McMillan

La verificació de la desigualtat de Kraft(/McMillan) permet comprovar l'existència un codi instantani (unívocament descodificable) D -ari de longituds L_1, \dots, L_n si i només si $\sum_{i=1}^n D^{-L_i} \leq 1$.

2.4 Verificació de si un codi és instantani.

Donat un conjunt de paraules-codi $C = \{c_1, \dots, c_n\}$, és un codi instantani si cap paraula-codi és prefix d'una altra. A més, si un codi és instantani, també és de descodificació única.

2.5 Fita inferior de la longitud mitjana

Segons el primer teorema de Shannon, qualsevol codi de descodificació única D -ari per a un conjunt de missatges $S = \{a_1, \dots, a_n\}$, amb longitud mitjana \bar{L} verifica $\bar{L} \geq H(S)/\log D$ amb igualtat si i només si $p_i = D^{-L_i}, \forall i = 1, \dots, n$.

2.6 Eficiència d'un codi

Donat un codi D -ari amb longitud mitjana \bar{L} per a un conjunt de missatges $S = \{a_1, \dots, a_n\}$, l'eficiència d'aquest codi és $\eta = H(S)/(\bar{L} * \log D) (\leq 1)$.

2.7 Informació mútua d'un canal

Per explicar el concepte d'informació mútua necessitem definir el concepte d'entropia condicionada.

L'entropia condicionada d' X donat Y , $H(X|Y)$, on X és l'entrada i Y la sortida, es defineix com la incertesa sobre X un cop es coneix Y . Donada una distribució de probabilitats conjuntes es calcula com:

$$H(X|Y) = -\sum_{i=1}^n \sum_{j=1}^m p(x_i, y_j) \log p(x_i|y_j).$$

Similarment, l'entropia condicionada d' Y donat X , $H(Y|X)$, donada una distribució de probabilitats conjuntes es calcula:

$$H(Y|X) = -\sum_{i=1}^n \sum_{j=1}^m p(y_j, x_i) \log p(y_j|x_i)$$

La informació mútua es defineix com $I(X, Y) = H(X) - H(X|Y) = H(Y) - H(Y|X)$. Aquest concepte és la mesura de la independència entre les variables aleatòries X i Y . També es pot definir com la mesura de la reducció de la incertesa d' X degut al coneixement del valor d' Y . Es calcula en bits.

2.8 Capacitat d'un canal

La capacitat del canal C és la quantitat màxima d'informació que es pot transportar per un canal de forma fiable.

$$C = \max_{0 < p_i \leq 1} I(X, Y)$$

Per saber la capacitat del canal primer hem de saber de quin tipus és.

Si la sortida determina de forma única l'entrada, $H(X|Y) = 0$, el canal és sense pèrdua. $C = \max H(X) = \log n$.

Si és a l'inrevés, que l'entrada determina de forma única la sortida, $H(Y|X) = 0$, el canal és determinista i es calcula com a $C = \max H(Y) = \log m$.

Cas que hi hagi una relació biunívoca entre l'entrada i la sortida, $H(X|Y) = 0$ i $H(Y|X) = 0$, en canal és sense sorolls. Es calcula com $C = \log n = \log m$.

Si la informació mútua entre X i Y és $I(X, Y) = 0$, per tant, no existeix cap relació de dependència entre les dues variables aleatòries, el canal és amb entrada i sortida independents.

dents. No serveixen per transmetre informació, ja que l'entrada i la sortida són totalment diferents, per tant la capacitat del canal és $C = 0$.

Finalment, un canal és simètric si ho és respecte a l'entrada i respecte a la sortida. Si les files de la matriu de probabilitats condicionades són iguals o contenen algun canvi d'ordre, el canal és simètric respecte a l'entrada. Cas que la matriu de probabilitats condicionades tingui les columnes iguals llevat de canvis d'ordre, el canal serà simètric respecte a la sortida. Es calcula com $C = \log m - H$, on H és l'entropia de qualsevol de les files de la matriu de probabilitats condicionades.

2.9 Probabilitat mitjana d'error

La probabilitat mitjana d'error associada a una regla f és:

$$\bar{p}_e = \sum_{j=1}^m p(b_j) p_e(b_j) = 1 - \sum_{j=1}^m p(f(b_j), b_j),$$

on $p(f(b_j), b_j)$ és la probabilitat de tenir b_j a la sortida i $f(b_j)$ a l'entrada.

Si la regla f és a mínima probabilitat d'error, a cada b_j assignem $f(b_j) = a_i$ tal que $p(a_i | b_j)$ sigui màxima.

En canvi, si la regla f és a màxima versemblança, a cada b_j assignem $f(b_j) = a_i$ tal que $p(b_j | a_i)$ sigui màxima.

En un canal amb distribució inicial de probabilitats equiprobable, les dues regles, a mínima probabilitat d'error i a màxima versemblança, coincideixen.

3 ESTAT DE L'ART

Actualment trobem alguns mòduls que permeten implementar algunes de les funcions en diferents llenguatges. Per a Python trobem `pyentropy` [9] i `DIT` [10] que permet calcular la informació mútua i entropies, entre d'altres aspectes. Per a C i Matlab trobem una altra eina, `MIToolbox` [11], amb la seva respectiva implementació amb Java, `JavaMI` [12]. Totes aquestes eines implementen només una part del que s'ha desenvolupat en aquest projecte.

Es poden trobar totes les funcions implementades individualment, però cap les unifica. A part, cap d'elles implementa una interfície gràfica que permeti la interacció amb les funcions sense necessitat de tenir coneixements de la seva execució.

4 METODOLOGIA

El tipus de metodologia seguit per desenvolupar el paquet d'informació és del tipus iteratiu i incremental, que permet anar creant noves versions que solucionin les mancances de les anteriors a partir de fer repetides iteracions sobre les etapes de desenvolupament. A més, permet detectar i solucionar errors més ràpidament que no pas en el model en cascada.

Cada iteració consta de la fase de documentació i anàlisis dels requisits, disseny, codificació i test. Al finalitzar una iteració es crea una versió amb les noves funcionalitats desenvolupades durant la iteració.

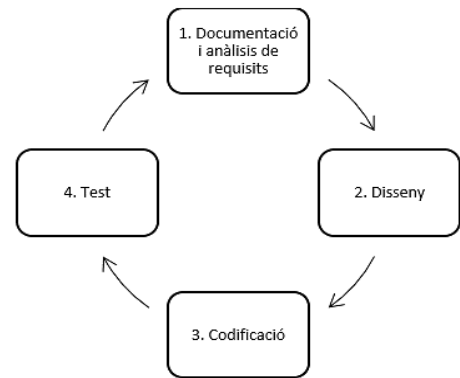


Fig. 1. Fases generals de la metodologia.

Degut a que Scrum no és una metodologia enfocada a projectes amb un sol membre, la metodologia seguida consisteix en una versió alleugerada d'SCRUM [13]. En aquesta versió lleugera s'inclouen els Sprints (setmanals), el product backlog (o tasques pendents) així com les reunions de planificació i revisió dels Sprints d'SCRUM.

4.1 Llenguatge de programació

Abans d'iniciar el desenvolupament del projecte s'havia de seleccionar el llenguatge de programació que s'adequés a les nostres necessitats. El llenguatge elegit és Python [14], degut a que és un llenguatge versàtil que ha vist generalitzat el seu ús gràcies a la seva senzillesa, rendiment i compatibilitat amb la majoria de sistemes. A més, ofereix algunes llibreries matemàtiques que el fan un potent llenguatge per realitzar càlculs [15].

Un dels motius pels quals m'ha fet decantar a triar-lo és que el programari matemàtic Sage [16], utilitzat àmpliament per la comunitat científica matemàtica, està desenvolupat en aquest llenguatge de programació i per tant permetria importar aquesta llibreria. A més, Python ofereix molta facilitat amb el tractament de matrius i llistes enfront d'altres llenguatges.

Sens dubte, en aquest projecte no hi ha cap limitació que prohibeixi cap llenguatge de programació i per tant s'ha triat el llenguatge que ofereix més facilitat, alhora que rendiment per programar tant la llibreria com la interfície gràfica.

La versió concreta de Python utilitzada per a la codificació és la 2.7.9. El desenvolupament de la llibreria de funcions s'ha realitzat des de zero, sense l'ús de cap base ni mòdul extern dels que s'ha parlat a l'apartat de l'estat de l'art.

4.2 Paquet de desenvolupament de la GUI

A part del llenguatge de programació, també s'havia de realitzar la selecció d'un *framework* per desenvolupar la interfície gràfica. En aquest cas es van valorar diferents alternatives ofertes per a Python. Els quatre paquets analitzats són: `Tkinter` [17], `frontend` per a Python inclòs en els instal·ladors oficials de Python; `wxPython` [18], `PyQT` [19], i `PyGTK` [20].

A continuació es mostra una comparativa de les quatre alternatives considerades:

	Tkinter	wxPython	PyQT	PyGTK
Instal·lació	Preinstal·lat amb Python	Instal·lació a part	Instal·lació a part	Instal·lació a part (a Windows s'ha de realitzar més d'una instal·lació)
Documentació	Completa	Completa i comunitat activa	Precària	Molt precària
Dificultat	Fàcil	Complex	Complex	Complex
Nombre d'elements gràfics	Limitats (sense llistes, ni arbres...)	Completa	Completa	Completa
Personalització del comportament de la interfície	Limitat	Flexible	Flexible	Flexible
Velocitat de creació d'elements	Lent (dibuixa cada element per separat)	Ràpid	Ràpid	Lent (dibuixa cada element per separat)
Aparença	Sense aparença nativa	Natiu del sistema operatiu	Natiu del sistema operatiu (només a Windows)	Intenta construir una aparença nativa (lent)

Fig. 2. Comparativa de les alternatives per al desenvolupament GUI.

Finalment m'he decantat per utilitzar wxPython, ja que a part de tenir una de les millors documentacions, dota al programa d'un aspecte natiu al sistema operatiu on s'executa i, a més, ofereix una flexibilitat major quant a elements gràfics i a la seva personalització, juntament amb un bon rendiment. El paquet wxPython es basa en wxWidgets [21], una llibreria C++ per a la creació de programes multiplataforma en aquest llenguatge de programació. La versió de wxPython utilitzada és la 3.0.

4.3 Planificació

Les activitats principals a realitzar per a l'obtenció dels objectius són les següents:

- Reunió de la documentació necessària de cada funció i de la seva implementació.
- Estudi dels diferents llenguatges de programació i diferents eines per desenvolupar la GUI.
- Desenvolupament de les funcions amb la documentació necessària.
- Fer proves d'unitat i d'integració de les funcions i la GUI.

A partir d'aquestes activitats es va realitzar la definició de les tasques concretes a fer, amb la seva estimació temporal. Les tasques definides van ser:

1. Estudi sobre la implementació del sistema.
 - 1.1. Elecció del llenguatge de programació.
 - 1.2. Elecció de la plataforma per desenvolupar la GUI.
2. Desenvolupament de la llibreria de funcions.
 - 2.1. Documentació prèvia i disseny.
 - 2.2. Programació.
 - 2.2.1. Codificació.
 - 2.2.2. Documentació del codi.
 - 2.2.3. Test.
3. Desenvolupament de la interfície gràfica.
 - 3.1. Documentació prèvia i disseny.
 - 3.2. Programació.
 - 3.2.1. Codificació
 - 3.2.2. Test individual
 - 3.3. Test d'integració complet.

La planificació inicial ha estat sotmesa a diferents canvis

temporals, sense patir modificacions en la data final proposada inicialment.

5 DESENVOLUPAMENT DEL TREBALL

En aquest apartat s'explicarà com s'ha desenvolupat el projecte i les eines utilitzades per arribar als resultats desitjats.

5.1 Documentació i anàlisis dels requisits

Aquesta etapa consisteix en la recollida de la documentació i els requisits necessaris per facilitar la feina a les següents fases.

En aquesta fase s'inclou la reunió inicial amb el professor per decidir què havia de contenir la llibreria de funcions, d'on se'n van extraure els requisits. De les reunions posteriors, on es mostrava el progrés de la interfície gràfica, s'obtenia feedback sobre la seva usabilitat que permetien refinar la idea inicial.

A banda de les reunions també es va realitzar la cerca de la informació necessària per implementar cada funció, la documentació sobre el llenguatge de programació, els *frameworks* GUI i les eines necessàries per crear un executable.

5.2 Disseny

L'etapa prèvia a la codificació és el disseny, aquesta fase es pot dividir en dues parts: disseny de la llibreria de funcions i disseny de les pantalles de la interfície gràfica.

Es va decidir realitzar la llibreria com una sola classe, juntament amb una classe dedicada a les proves per verificar el correcte funcionament dels mètodes implementats. Per realitzar el disseny d'aquesta part s'ha utilitzat l'eina draw.io [22].

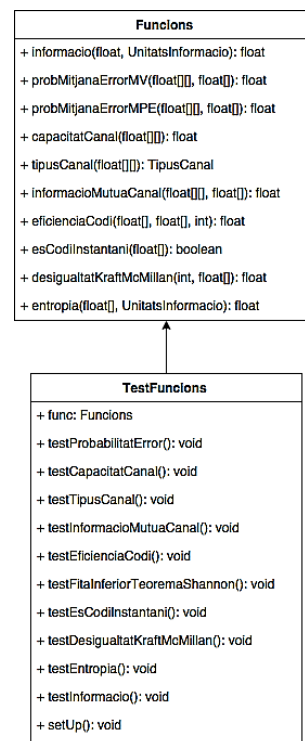


Fig. 3. Diagrama UML de la llibreria de funcions utilitzant draw.io.

Pel que fa al disseny de la interfície, s'ha utilitzat l'eina de disseny visual wxFormBuilder [23] que genera el codi Python de l'estructura dissenyada de forma automàtica, abstractant al programador de la implementació a alt nivell dels widgets de wxPython.

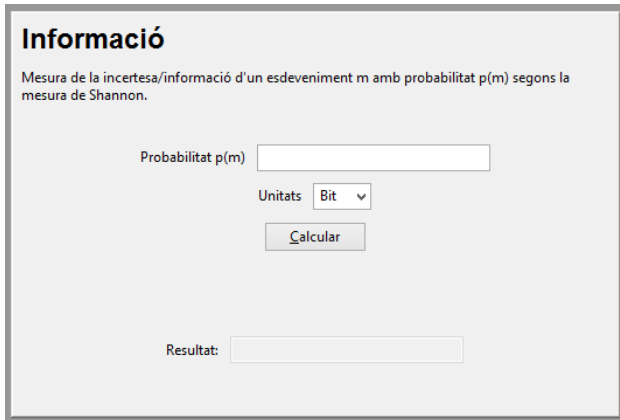


Fig. 4. Exemple de pantalla creada amb wxFormBuilder.

5.3 Codificació

Degut a que la IDE per defecte de Python anomenada IDLE [24] és molt bàsica i no permet gestionar projectes amb diferents arxius .py es va optar per buscar alternatives gratuïtes. Després de cercar informació sobre cadascuna d'elles [25], es va decidir utilitzar PyCharm [26] com a eina principal per fer la codificació tant del *backend* com del *frontend*. PyCharm és un potent editor intel·ligent de codi que facilita la codificació mitjançant el ressaltat dels errors de sintaxi en temps real, autosagnat, completador de codi, visió dels arxius del projecte en forma d'arbre i integració amb eines de test d'unitat, entre d'altres aspectes.

5.4 Test

A l'etapa de test s'han utilitzat diferents mètodes per realitzar les proves del programa i la llibreria. Per a la llibreria s'ha utilitzat el *framework* de proves d'unitat inclòs en Python, PyUnit [27] que és una adaptació del popular JUnit de Java. Aquest *framework* ofereix la possibilitat de crear *test cases* per comprovar, de forma automàtica, que els mètodes i classes implementades retornen els resultats esperats i que després de realitzar modificacions en el codi segueixen funcionant correctament.

```
def testEntropia(self):
    self.assertAlmostEqual(self.func.entropia([0.1,0.9]), 0.46899559)
    self.assertAlmostEqual(self.func.entropia([0.5,0.5]), 1)
    self.assertAlmostEqual(self.func.entropia([0.2,0.2,0.6]), 1.37095059)
    self.assertAlmostEqual(self.func.entropia([0.0, 1.0]), 0)
    self.assertAlmostEqual(self.func.entropia([0.5,0.5,0.0]), 1)

#Excepcions
self.assertRaises(Exception, self.func.entropia, [0.1,0.2])
self.assertRaises(Exception, self.func.entropia, 0.0)
self.assertRaises(Exception, self.func.entropia, 0.1)
self.assertRaises(ValueError, self.func.entropia, [2.0, -1.0])
self.assertRaises(ValueError, self.func.entropia, [-2.0, 3.0])
```

Fig. 5. Exemple de test cases utilitzant PyUnit.

Per a la interfície s'ha utilitzat el mètode de prova i error per qüestions de temps, degut a que crear els tests automàtics hagués portat més temps que no pas el temps que s'ha necessitat per fer les proves de la GUI mitjançant prova i error. Les entrades que l'usuari pot introduir es van repetint al llarg de les diferents pantalles fetes i, per tant, no ha estat necessari tornar a comprovar constantment si la introducció d'una certa dada d'entrada donava una sortida no esperada que havia estat solucionada en fases anteriors.

5.5 Creació de l'executable

Per crear un executable d'un codi Python és necessari l'eina py2exe [28] que requereix crear un arxiu en el projecte on es configuren les opcions que s'utilitzen durant la creació del .exe. Aquest arxiu rep el nom de setup.py i un cop creat, mitjançant una consola del sistema, l'executem amb la comanda "python setup.py py2exe" que crearà una carpeta build amb els arxius temporals que ha generat durant l'execució, i una carpeta "dist" amb el resultat final.

L'objectiu d'aquesta eina és crear un seguit d'arxius on s'inclouen totes les dependències necessàries per fer que no sigui necessària la instal·lació de cap programa a part. En aquest cas, l'executable ha de contenir un dll amb la versió utilitzada de Python i wxPython. A més, es pot configurar per crear un sol .exe sense cap altre arxiu. Per configurar que només es creï un sol arxiu, hem de configurar la opció *bundle_files* a 1, que agrupa tots els arxius temporals en un de sol, i *zipfile* a None que inclou aquesta agrupació dins l'executable [29].

Nombre	Fecha de modifica...	Tipo	Tamaño
_hashlib.pyd	30/06/2014 17:04	Archivo PVD	699 KB
bz2.pyd	30/06/2014 17:04	Archivo PVD	67 KB
library	15/05/2015 17:31	Archivo WinRAR Z...	4.115 KB
MainFrame	15/05/2015 17:31	Aplicación	24 KB
python27.dll	30/06/2014 17:03	Extensión de la apl...	2.397 KB
select.pyd	30/06/2014 17:04	Archivo PVD	10 KB
unicodedata.pyd	30/06/2014 17:04	Archivo PVD	670 KB
wxppopen	30/06/2014 17:02	Aplicación	49 KB
wx_controls.pyd	27/11/2014 14:09	Archivo PVD	1.042 KB
wx_core.pyd	27/11/2014 14:09	Archivo PVD	1.149 KB
wx_gdi.pyd	27/11/2014 14:09	Archivo PVD	788 KB
wx_grid.pyd	27/11/2014 14:10	Archivo PVD	429 KB
wx_misc.pyd	27/11/2014 14:10	Archivo PVD	716 KB
wx_windows.pyd	27/11/2014 14:09	Archivo PVD	797 KB
wx_xrc.pyd	27/11/2014 14:10	Archivo PVD	146 KB
wxbase30u_net_vc90.dll	27/11/2014 14:06	Extensión de la apl...	152 KB
wxbase30u_vc90.dll	27/11/2014 14:06	Extensión de la apl...	1.983 KB
wxbase30u_xml_vc90.dll	27/11/2014 14:07	Extensión de la apl...	133 KB
wxmsw30u_adv_vc90.dll	27/11/2014 14:07	Extensión de la apl...	1.222 KB
wxmsw30u_core_vc90.dll	27/11/2014 14:07	Extensión de la apl...	4.685 KB
wxmsw30u_html_vc90.dll	27/11/2014 14:07	Extensión de la apl...	587 KB
wxmsw30u_xrc_vc90.dll	27/11/2014 14:08	Extensión de la apl...	671 KB

Fig. 6. Execució sense 'bundle_files':1 i zipfile = None.

Nombre	Fecha de modifica...	Tipo	Tamaño
Teoria_Informacio	04/06/2015 20:42	Aplicación	22.510 KB
wxppopen	30/06/2014 17:02	Aplicación	49 KB

Fig. 7. Resultat de l'execució amb la configuració final de py2exe.

A més, les opcions que s'han inclòs en aquest projecte són: 'script' que indica quina classe .py serà la que s'executi; i "dest_base" que indica el nom de l'executable final.

El problema va sorgir quan es volien incloure imatges, ja que la manera d'incloure arxius addicionals a l'executable no permetia agrupar-los i per tant copiava les imatges a la carpeta on es creava l'executable. Per solucionar aquest inconvenient, wxPython inclou una eina anomenada img2py [30] que converteix una imatge en format PNG en una

classe Python. Això ho aconsegueix a partir d'una codificació en base 64, que és un esquema de representació de dades binàries en format ASCII.

5.6 Documentació de les funcions

Per documentar les classes i mètodes necessaris per a la llibreria s'han comentat les classes i els mètodes creats de forma que s'explica què fa cadascuna de les funcions, els seus paràmetres, què retorna i les excepcions que pot llançar (en el cas que en llanci alguna).

```
def eficienciaCodi(self, S, C, D):
    """
    Mètode que retorna l'eficiència d'un codi.
    Suposem que els codis són de descodificació única.
     $n = H(S)/(^L \cdot \log(D))$  ( $<=1$ ).

    Paràmetres:
    S: distribució de probabilitats del conjunt de codis.
    C: llistat de codis (cada codi és un String).
      Exemple: ["0", "10"].
    D: tipus de codi D-ari (binari, terciari...).

    Retorna l'eficiència d'un codi [0.0, 1.0].

    Excepcions:
    Exception: Si @S no és una llista o si el nombre de
      probabilitats no és igual al nombre de codis.
    ValueError: Si la eficiència no retorna un valor comprès
      entre [0.0, 1.0].
    """
```

Fig. 8. Documentació d'un mètode fet amb pydocs.

Un cop realitzats els comentaris dins de cada mètode, Python inclou un mòdul, pydoc [31] que genera la documentació automàticament a partir d'aquests comentaris fets en el codi. Al executar la comanda "*python -m pydoc -w <classe>*" es crea la documentació en format HTML.

5.7 Ús de probabilitats fraccionàries

Per facilitar la introducció i visualització de probabilitats en el programa he cregut convenient que l'usuari pogués fer l'entrada d'un valor fraccionari en comptes de fer-ho només en format decimal, ja que en alguns dels casos, com per exemple 1/3, podria donar lloc a resultats inexactes. Això s'ha implementat utilitzant la classe fractions [32] de Python que permet convertir un flotant en una fracció. A més, s'ha incorporat la funcionalitat que a l'introduir un nombre que li correspon una fracció, per exemple 0.5, se li fa la conversió a 1/2. Aquest canvi només es realitza als nombres flotants que al convertir-los en fraccions tinguin un numerador menor a 10 i un denominador menor a 100 per evitar que es mostrin fraccions massa grans.

L'ús de nombres racionals fraccionaris permet treballar amb millor precisió i sense introduir errors produïts per la limitació de la mida dels tipus de variables.

A més, si tots els nombres introduïts són fraccionaris o se'ls ha pogut realitzar la conversió comentada, el resultat també serà en forma de fracció.

Ara mostraré un petit exemple per veure quin és l'efecte d'utilitzar o no Fraction. Tenim un nombre a que té com a valor 0.3333333 (7 decimals), si utilitzem la funció Fraction per convertir a a fracció surt com a resultat 1/3. Seguidament, realitzem la multiplicació $a*a$ i la multiplicació entre els dos nombres fraccionaris. El resultat és el següent:

```
a = 0.3333333
Fraction(a) = 1/3
a*a = 0.111111088889
Fraction(a)*Fraction(a) = 1/9
```

Com es pot apreciar, l'error comença a aparèixer a partir del 7è decimal, mentre que amb Fraction se segueix mantenint el valor amb precisió.

6 RESULTATS

El resultat del projecte és una llibreria de funcions amb la seva corresponent interfície gràfica codificada en el llenguatge Python que permet fer el càlcul de les funcions comentades a l'apartat d'introducció d'aquest document.

Està dissenyada per tal de poder-se utilitzar amb qualsevol sistema operatiu ja que s'ha utilitzat Python i el *framework* wxPython que tenen les seves versions per a cadascun dels tres sistemes operatius actuals més importants (Windows, Linux, Mac). A més, es proveeix d'un executable per als usuaris de Windows que permet utilitzar el programa sense necessitat de tenir instal·lat Python ni wxPython a l'ordinador mitjançant la compilació amb py2exe.

L'idioma tant de la documentació com de la interfície és en català.

Per exposar els resultats obtinguts, es donarà el grau d'assoliment dels objectius proposats. Els objectius del projecte són:

1. Facilitar una llibreria amb una part dels continguts donats a l'assignatura d'Informació i Seguretat.

Pel que fa a aquest primer objectiu, el grau d'assoliment és satisfactori donat que s'ha aconseguit desenvolupar el codi de la llibreria i realitzar tests d'unitat per validar el seu funcionament.

A més, aquesta llibreria es pot utilitzar en altres entorns com ara Sage (s'explicarà al primer subapartat d'aquesta secció), MATLAB [33], C/C++ [34], i Java [35].

2. Garantir la compatibilitat amb tots els Sistemes Operatius.

La selecció del llenguatge de programació i del *framework* GUI ha estat clau per assolir aquest objectiu. Python és un potent llenguatge present a totes les plataformes, fins i tot trobem tutorials per instal·lar-lo en telèfons mòbil Android i iOS.

3. Oferir una interfície *friendly* per a la utilització de la llibreria.

Aquest és l'objectiu que potser és més difícil de saber si s'ha assolit, encara que s'ha apostat pel disseny d'una interfície fàcil d'utilitzar i al tenir poques opcions no ha de resultar difícil el seu ús. A més, s'ha afegit la funcionalitat de poder introduir directament probabilitats en format de fracció i també el retorn dels resultats com a nombres fraccionaris. D'altra banda, el rebre feedback per part del tutor del projecte ha ajudat a millorar-ne el seu aspecte i l'experiència d'usuari.

Com a punts clau per aconseguir aquest objectiu, s'ha dotat al programa de matrius en format de taules que donen l'opció a l'usuari d'introduir les matrius de probabilitats condicionades entre l'entrada i la sortida manualment de manera agradable. De la mateixa manera, a l'importar

una matriu des d'un arxiu .txt, es pot visualitzar que la importació de la matriu s'ha realitzat correctament i amb els valors desitjats. Cas contrari es permet realitzar modificacions abans de fer el càlcul de l'operació.

També s'ha afegit l'opció d'utilitzar dreceres del teclat per moure's pels menús i utilitzar el botó de "Calcular" de cadascuna de les pantalles sense haver de fer clic amb el ratolí. La tecla modificadora utilitzada és l'ALT, i al prémer-la es subratlla la lletra que cal pulsar per aplicar la drecera.

4. Facilitar una documentació completa de cadascuna de les funcions.

Aquest objectiu s'ha pogut completar mitjançant l'ús de la nomenclatura pydoc, la utilització de noms de funcions i variables autocomentades i també s'han inclòs comentaris *inline* per aclarir certs aspectes del codi.

S'inclouen, a més, uns .html de la documentació generada a partir dels comentaris pydoc, per tenir sempre la documentació a mà i no haver d'obrir el codi font de la llibreria.

5. Oferir un executable que no requereixi d'instal·lacions complementàries.

Amb la utilització de l'eina py2exe que compila classes escrites en Python amb les seves dependències corresponents oferint un executable compatible amb totes les versions dels sistemes operatius Windows. La compilació amb l'eina py2exe només es pot realitzar en entorns Windows, ja que necessita tenir instal·lat la llibreria Visual C++ 2008. Aquest executable és capaç d'obrir el programa sense tenir instal·lat Python i/o el mòdul wxPython a l'ordinador on s'executa.

6.1 Exemple d'ús

En aquest apartat es mostra un exemple de com s'utilitza la llibreria de funcions en una eina externa com és Sage.

Utilitzant Sage Cloud [36] se'ns permet importar arxius .py directament utilitzant la pestanya New. Un cop dins d'aquesta pestanya, introduïm el nom que li volem donar al fitxer, seguidament pugem l'arxiu Funcions.py que conté la llibreria de funcions i finalment seleccionem Python (.py) com a tipus d'arxiu.

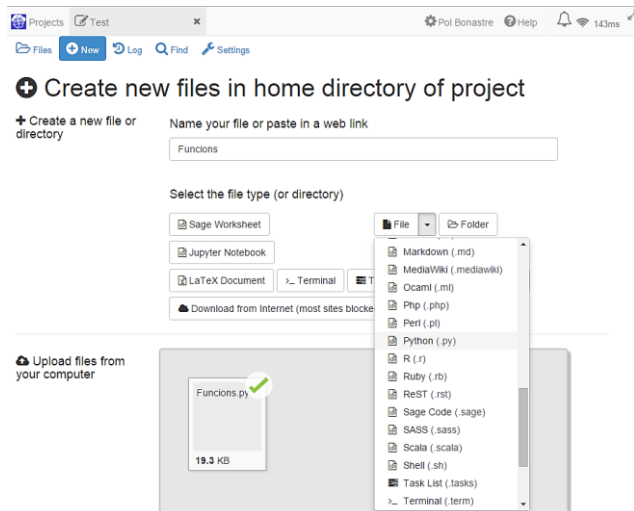


Fig. 9. Importació d'un arxiu Python a Sage Cloud.

Un cop pujat es pot visualitzar el contingut de la llibreria i modificar-la si fos necessari. En nostre cas, la tancarem i crearem un Sage Worksheet que serà des d'on es realitzen les operacions.

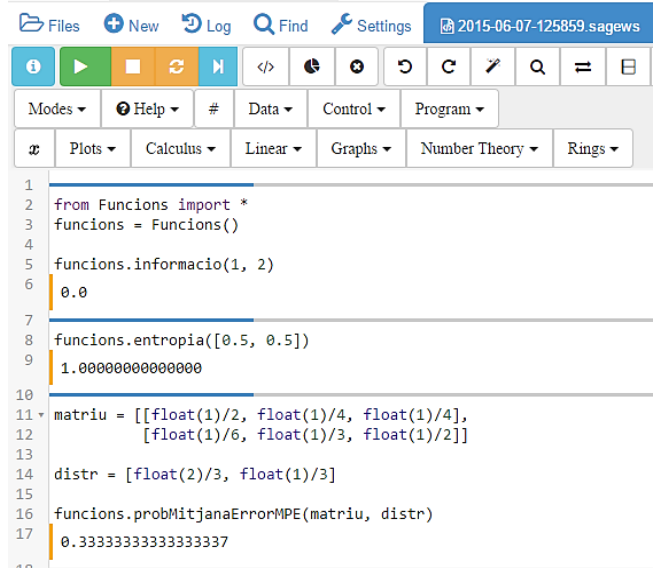


Fig. 10. Utilització de la llibreria importada.

Com es pot veure, amb pocs passos s'ha pogut importar i començar a utilitzar. Pel que fa a Sage Notebook, la plataforma antiga que s'utilitza a la universitat en diferents assignatures, m'he trobat amb el problema que a l'importar l'arxiu .py el navegador web retorna un error intern o una pàgina en blanc en el seu defecte. Això és degut a que la llibreria de funcions conté accents i símbols en els comentaris i s'origina un error de codificació al intentar pujar l'arxiu al servidor. Com a mètodes alternatius, podem copiar i enganxar el codi de la llibreria dins d'un Worksheet.

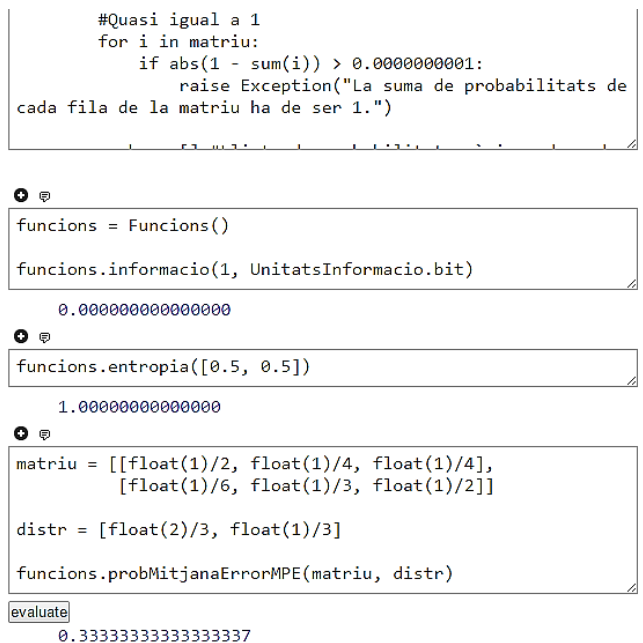


Fig. 11. Utilització de la llibreria copiada i enganxada al Sage.

O bé, si es té accés físic al servidor es pot mantenir una còpia de la llibreria a un directori conegut i posar en el worksheet “*attach /ruta/arxiu.py*” per carregar-la i poder utilitzar-la.

```

attach "/home/sage/sage/Funcions.py"

funcions = Funcions()
funcions.informacio(1, UnitatsInformacio.bit)

0.0

funcions.entropia([0.5, 0.5])

1.0000000000000000

matriu = [[float(1)/2, float(1)/4, float(1)/4],
           [float(1)/6, float(1)/3, float(1)/2]]

distr = [float(2)/3, float(1)/3]

funcions.probMitjanaErrorMPE(matriu, distr)

0.3333333333333333

```

Fig. 12. Exemple d'utilització de la comanda *attach*.

7 CONCLUSIONS

Un cop finalitzat el projecte es pot comprovar que s'han anat complint els objectius marcats des d'un bon principi. A més, s'ha seguit la planificació proposada inicialment sense haver de fer grans canvis, ajustant-se sempre a les dates d'entrega. La metodologia seguida potser té algunes mancances sobretot en l'aspecte de no disposar d'un històric de tasques fetes i pendents a fer, però al tractar-se d'un projecte petit i amb un sol desenvolupador no ha suposat cap inconvenient. A més, haver utilitzat algun control de versions com ara *Git* hagués facilitat la feina. Tot i així s'ha complert amb els requisits desitjats.

En un principi el projecte semblava fàcil i ràpid de fer. El desenvolupament de la llibreria de funcions, la seva documentació i el test estaven acord amb els recursos i temps disponibles per acabar el projecte ajustant-se als requisits i restriccions. Sent un sol desenvolupador, la feina de disseny i codificació de la interfície gràfica ha suposat un repte per ajustar-se a la planificació.

A part d'això, el projecte m'ha aportat un nou aspecte que fins ara no havia aplicat per realitzar cap altre projecte, la capacitat d'analitzar i seleccionar el llenguatge de programació que s'adequa més a les necessitats del programa, així com la capacitat de seleccionar entre diferents *frameworks* GUI per realitzar la interfície gràfica. A més, m'ha permès incrementar els meus coneixements amb el llenguatge de programació Python i descobrir wxPython per fer interfícies gràfiques, com el disseny i creació d'aplicació compilable a tots els sistemes operatius, i a la creació d'executables per a Windows a partir d'un projecte Python.

Una de les possibles línies futures del projecte seria l'adició de noves funcions relacionades amb la teoria de la informació en l'àmbit de la compressió de dades i control i

detecció d'errors. També es podria implementar la internacionalització a la interfície per tal d'afegir nous idiomes.

8 AGRAÏMENTS

Agrair al tutor Joaquim Borges per guiar-me durant la realització del projecte, així com les seves propostes de millora i per l'atenció donada.

9 BIBLIOGRAFIA

- [1] C.E. Shannon. "A Mathematical Theory of Communication". Bell System Technical Journal, 1948
- [2] A. López, A. Parada, i F. Simonetti. "Introducción a la Teoría de la Información". Internet: http://cs.uns.edu.ar/~ldm/data/ss/info/teoria_de_la_informacion2.pdf [1 de juny, 2015]
- [3] C. E. Shannon, W. Weaver. "The Mathematical Theory of Communication". Univ of Illinois Press, 1949.
- [4] Informació i Seguretat. "Tema 2: Conceptes bàsics de la teoria de la informació", 2013. PDF. [21 febrer, 2015]
- [5] Informació i Seguretat. "Tema 3 : Codificació de la font", 2013. PDF. [21 febrer, 2015]
- [6] Informació i Seguretat. "Tema 5: Codificació del canal", 2013. PDF. [21 febrer, 2015]
- [7] Teoria de la Informació. "Capítol 5 Transmissió d'informació per canals amb sorolls", 2012. PDF. [18 març, 2015]
- [8] R.B. Ash. "Information Theory". John Wiley and Sons Inc, 1965.
- [9] Pyentropy. "Entropy and Information Theory Toolbox for Python". Internet: <https://code.google.com/p/pyentropy/> [2 de juny, 2015]
- [10] Dit. "Python package for information theory". Internet: <https://github.com/dit/dit> [2 de juny, 2015]
- [11] G. Brown, A. Pocock, M. Zhao, i M. Luján. "MITtoolbox for C and MATLAB". Internet: <http://www.cs.man.ac.uk/~pococka4/MITtoolbox.html> [2 de juny, 2015]
- [12] A. Pocock. "JavaMI Toolbox". Internet: <http://www.cs.man.ac.uk/~pococka4/JavaMI.html> [2 de juny, 2015]
- [13] X. Albaladejo. "Requisitos para poder utilizar Scrum". Internet: <http://www.proyectosagiles.org/requisitos-de-scrum> [14 de maig, 2015]
- [14] Python Software Foundation. "Python". Internet: <https://www.python.org/> [11 de febrer, 2015]
- [15] CACHEM. "Python para ingenieros". Internet: <http://caccheme.org/python-para-ingenieros/> [22 de febrer, 2015]
- [16] Sage. "SageMath". Internet: <http://www.sagemath.org/> [2 de Juny, 2015]
- [17] Python. "TKinter". Internet: <https://wiki.python.org/moin/TkInter> [23 de març, 2015]
- [18] wxPython. "wxPython". Internet: <http://www.wxpython.org/> [23 de març, 2015]
- [19] Riverbank. "PyQT". Internet: <http://www.riverbankcomputing.co.uk/software/pyqt/intro> [23 de març, 2015]
- [20] The GNOME Project i PyGTK Team. "PyGTK: GTK+ for Python" <http://www.pygtk.org/>, 30 desembre, 2009. [23 de març, 2015]
- [21] wxWidgets. "wxWidgets: Cross-Platform GUI Library". Internet: <https://www.wxwidgets.org/> [2 de juny, 2015]

- [22] draw.io. "Flow Chart Maker & Online Diagram Software. Internet: <https://www.draw.io/> [11 de febrer, 2015]
- [23] SourceForge. "wxFormBuilder" Internet: <http://sourceforge.net/projects/wxformbuilder/>, 06 juny 2014 [22 de març, 2015]
- [24] Lester I. McCann. "Using IDLE (Python's IDE)". Internet: <http://www.cs.arizona.edu/people/mccann/usingidle> [23 de març, 2015]
- [25] J. Fruit "Comparison of Python IDEs for Development". Internet: <http://www.pythoncentral.io/comparison-of-python-ides-development/> [12 de febrer, 2015]
- [26] JetBrains. "PyCharm". Internet: <https://www.jetbrains.com/pycharm/> [20 de febrer, 2015]
- [27] Python. "unittest – Unit testing framework". Internet: <https://docs.python.org/2/library/unittest.html>, 10 maig 2015 [14 maig, 2014]
- [28] py2exe. "FrontPage". Internet: <http://www.py2exe.org/>, 2 setembre 2014. [22 març, 2015]
- [29] py2exe. "SingleFileExecutable". Internet: <http://www.py2exe.org/index.cgi/SingleFileExecutable> [14 de maig, 2015]
- [30] Epydoc. "Module wx.tools.img2py". Internet: <http://www.wxpython.org/docs/api/wx.tools.img2py-module.html> [12 de maig, 2015]
- [31] Python Software Foundation. "pydoc - Documentation generator and online help System". Internet: <https://docs.python.org/2/library/pydoc.html>, 10 maig 2015 [14 de maig, 2015]
- [32] Python Software Foundation. "fractions – Rational numbers". Internet: <https://docs.python.org/2/library/fractions.html> [14 d'abril, 2015]
- [33] ALGOholic. "matpy - call Python from MATLAB". Internet: <http://alcoholic.eu/matpy/> [12 de juny, 2015]
- [34] Python Software Foundation. "Embedding Python in Another Application". Internet: <https://docs.python.org/2/extending/embedding.html> [12 de juny, 2015]
- [35] Python Software Foundation. "Jython and Java Integration". Internet: <http://www.jython.org/jythonbook/en/1.0/JythonAndJavaIntegration.html> [12 de juny, 2015]
- [36] Sage. "SageCloud". Internet: <https://cloud.sagemath.com> [4 de Juny, 2015]