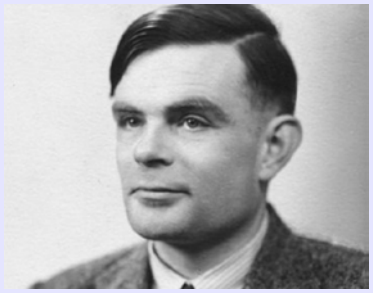
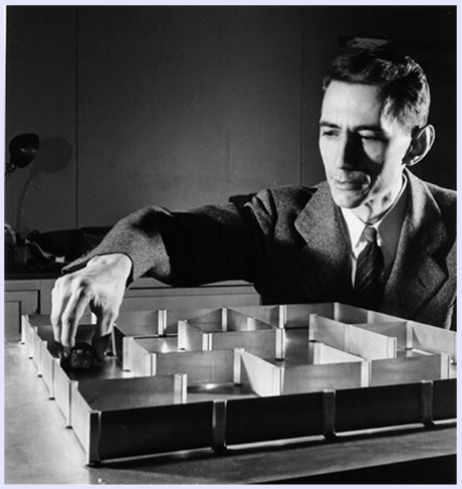
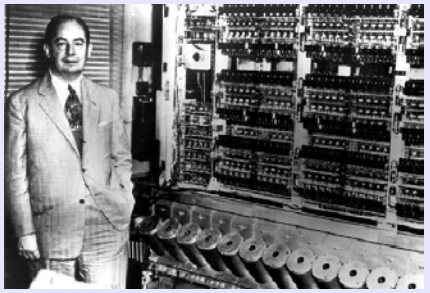




Information, Calcul et Communication

Leçon 2.3: Compression de données (1ère partie)

O. Lévêque – Faculté Informatique et Communications

Introduction

Lors des deux leçons précédentes, nous avons répondu aux questions suivantes:

- Comment représenter / capter la réalité physique avec des bits ?
- Comment restituer cette réalité à partir de bits ?

Lors des deux leçons à venir, nous allons tenter de répondre aux questions suivantes:

- Comment mesurer la quantité d'information présente dans des données ?
- Comment stocker des données en utilisant le moins d'espace possible (sans ou avec pertes) ?

Introduction (2)

Pourquoi donc vouloir comprimer des données ?

pour réduire l'espace-mémoire utilisé lors du stockage de données

pour réduire le temps de transmission et les problèmes de congestion lors de la transmission des données

Cependant, avec les progrès de la technique, ne suffit-il pas d'attendre un peu pour avoir de meilleures performances?

Certes, mais on veut toujours exploiter un système au maximum de ses capacités!

Introduction (3)

Le français est plein de redondance! Pour preuve, ce texte:

Sleon une édtue de l'Uvinertisé de Cmabrigde, l'odrre des ltteers dnas un mot n'a pas d'ipmrotncae, la suele coshe ipmrotnate est que la pmeirère et la drenèire soinet à la bnnoe pclae. Le rsete peut êrte dans un dsérorde ttoal et vuos puoevz tujoruos lrie snas porlbème. C'est prace que le creaveu hmauin ne lit pas chuaqe ltetre elle-mmêe, mias le mot comme un tuot.

Pourquoi donc tant de redondance dans la langue française ?

- Pour pouvoir se comprendre, tout simplement.
- Pour être capable de lire un texte même s'il contient des fautes d'orthographe...

Introduction (4)

On distingue deux types de compression:

la compression sans pertes, lorsqu'on désire retrouver l'intégralité des données stockées sous forme comprimée.

Exemples: billets pour un concert, déclaration d'impôts, bulletins de vote, articles scientifiques

la compression avec pertes, lorsqu'on n'est pas tant à cheval que ça sur les détails et qu'on s'autorise un peu de *distorsion*.

Exemples: émissions podcastées et morceaux de musique en format mp3, partage de photos sur le web, vidéos youtube...

Plan

- **Plan détaillé des deux leçons à venir:**
- **Aujourd'hui:**
  - notion d'entropie
  - compression sans pertes
  - algorithme de Shannon-Fano

- **La semaine prochaine (*possibles modifications*)**
  - algorithme de Shannon-Fano (bis)
  - analyse de performance – théorème de Shannon
  - compression avec pertes

Entropie

Voici une séquence de 16 lettres:

A B C D E F G H I J K L M N O P

Jeu N° 1: Vous devez deviner quelle lettre j'ai choisi au hasard en posant un *nombre minimum de questions*, auxquelles je ne peux répondre que par oui ou par non.

Entropie (2)

Voici une autre séquence de 16 lettres (sans compter les espaces):

I L F A I T B E A U A I B I Z A

Jeu N° 2: Le jeu est le même qu'avant!

Remarques:

Je choisis la *position* de la lettre au hasard, de manière uniforme.

Vous ne devez deviner que la lettre elle-même, *pas sa position*.

Combien de questions binaires sont-elles nécessaires en moyenne pour deviner la lettre?

Entropie (3)

I L F A I T B E A U A I B I Z A

Solution: classer les lettres dans l'ordre décroissant du nombre d'apparitions dans la séquence:

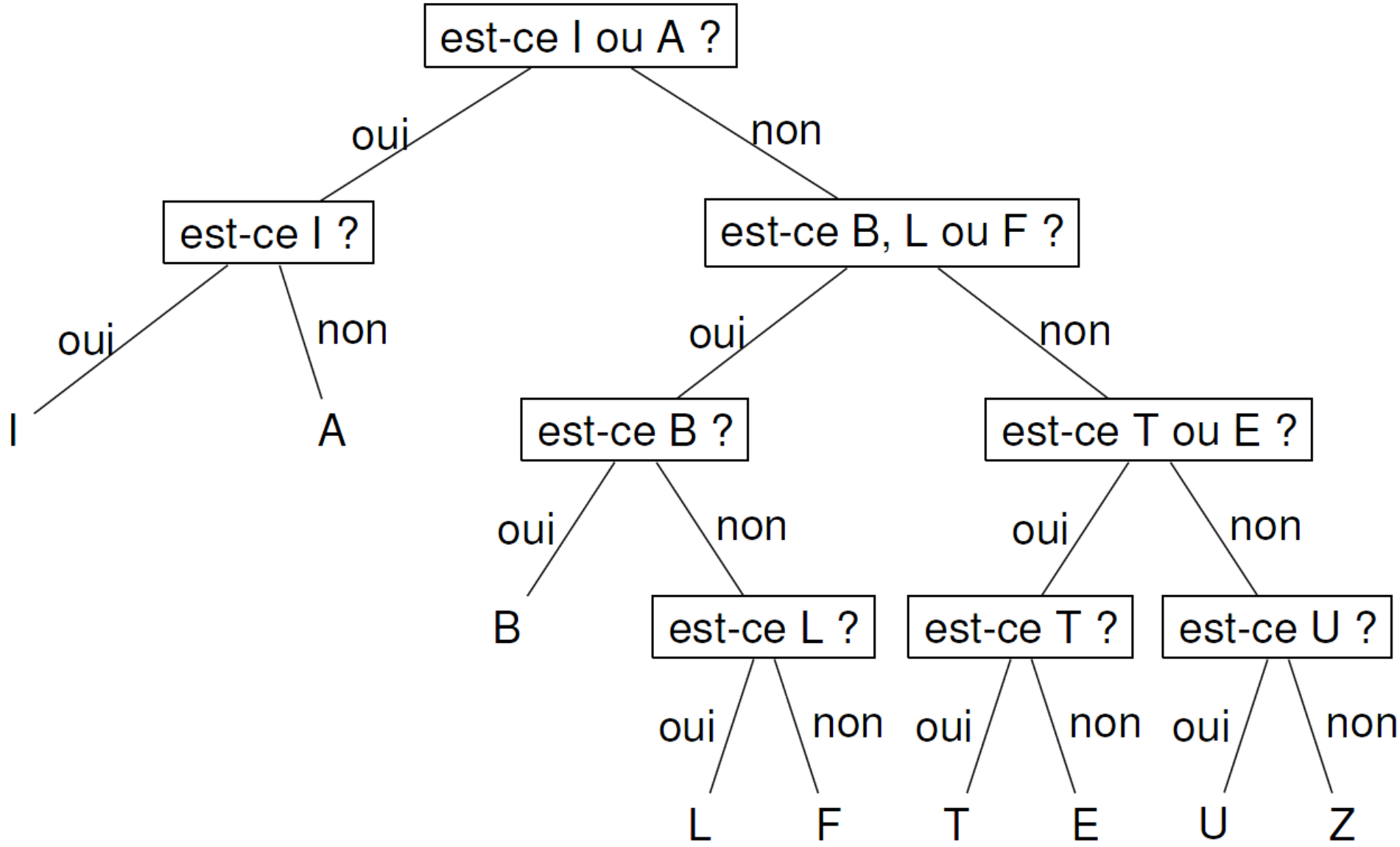
lettre	I	A	B	L	F	T	E	U	Z
nb d'apparitions	4	4	2	1	1	1	1	1	1

L'idée est la même que précédemment: on sépare l'ensemble des lettres en deux parties égales *en termes de nombre d'apparitions*, ce qui donne:

Question N° 1: est-ce que la lettre est un I ou un A?

Si la réponse est oui: **question n^0 2:** est-ce que la lettre est un I?

Si la réponse est non: **question n^0 2:** est-ce que la lettre est un B, un L ou un F? etc.


est-ce I ou A ?
oui
non
est-ce I ?
est-ce B, L ou F ?
oui
non
oui
non
I
A
est-ce B ?
est-ce T ou E ?
oui
non
oui
non
B
est-ce L ?
est-ce T ?
est-ce U ?
oui
non
oui
non
oui
non
L
F
T
E
U
Z

Entropie (4)

I L F A I T B E A U A I B I Z A

lettre	I	A	B	L	F	T	E	U	Z
nb d'apparitions	4	4	2	1	1	1	1	1	1
nb de questions	2	2	3	4	4	4	4	4	4

Nombre de questions à poser en moyenne:

$$= \frac{2 \times 4}{16} \times 2 + \frac{1 \times 2}{16} \times 3 + \frac{6 \times 1}{16} \times 4 = \frac{16 + 6 + 24}{16} = \frac{46}{16} = 2.875$$

On dit que l'entropie de cette séquence est égale à 2.875.

Entropie (5)

Voici encore une autre séquence de 16 lettres:

A A A A A A A A A A A A A A A A

Jeu n^0 3: Le jeu est encore le même qu'avant.

Cette fois-ci, *aucune* question n'est nécessaire pour deviner la lettre choisie!

On dit que l'entropie de cette séquence est égale à 0.

Entropie (6)

I L F A I T B E A U A I B I Z A

lettre	I	A	B	L	F	T	E	U	Z
nb d'apparitions	4	4	2	1	1	1	1	1	1
nb de questions	2	2	3	4	4	4	4	4	4

Pour deviner une lettre qui apparaît 1 fois sur 16,
on a besoin de 4 questions. $4 = \log_2(16)$

Pour deviner une lettre qui apparaît 2 fois sur 16 (i.e. 1/8),
on a besoin de 3 questions. $3 = \log_2(8)$

Pour deviner une lettre qui apparaît 4 fois sur 16 (i.e. 1/4),
on a besoin de 2 questions. $2 = \log_2(4)$

En résumé, pour deviner une lettre qui apparaît avec une probabilité p,
on a besoin de $\log_2(1/p)$ questions.

Entropie (7)

Définition générale:

Soit X une séquence de lettres provenant d'un alphabet $A = \{a_1, \ldots, a_n\}$. Soit p_j

la probabilité d'apparition de la lettre a_j dans la séquence

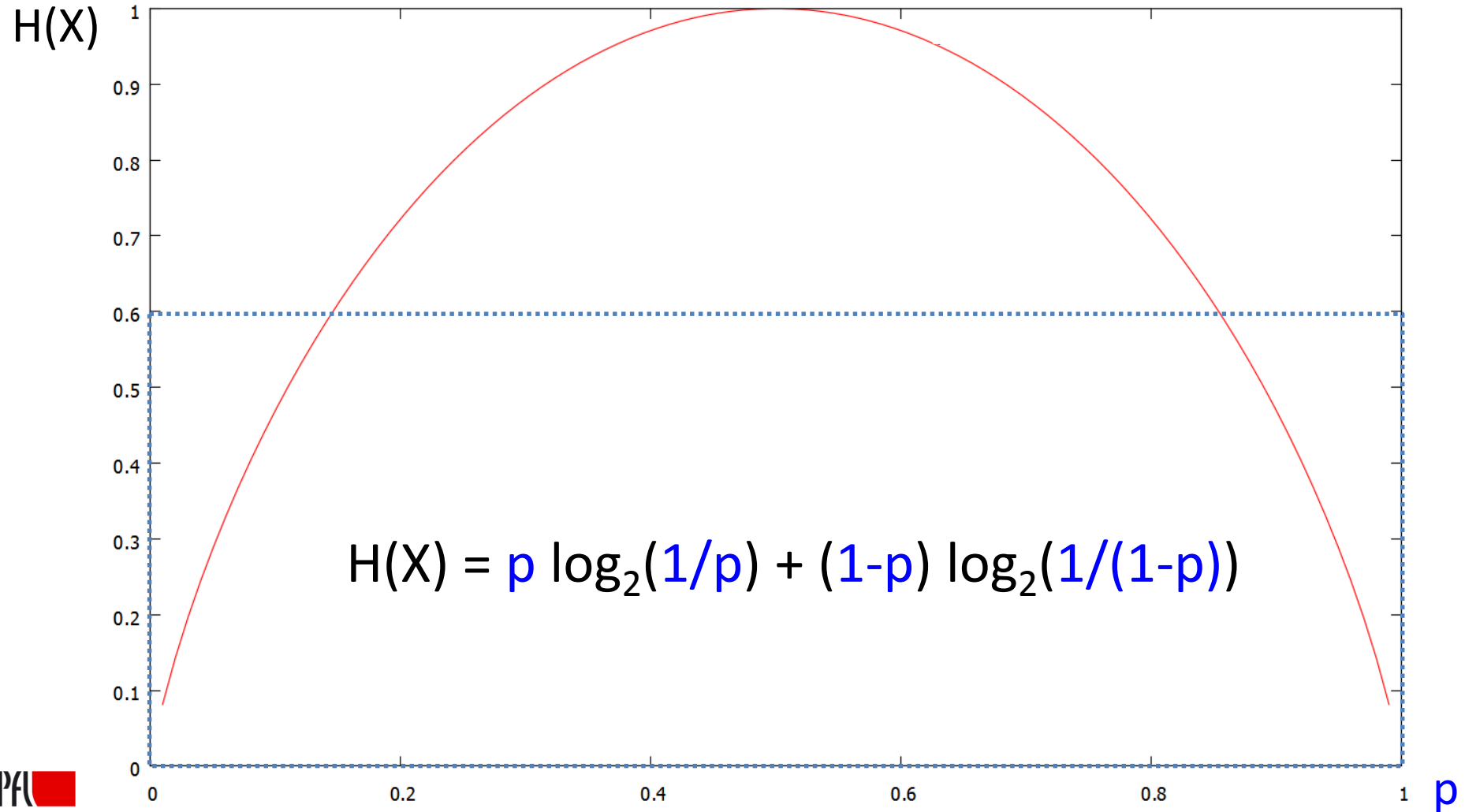
(remarquer que $0 \leq p_j \leq 1$ pour tout j et que $p_1 + \ldots + p_n = 1$).

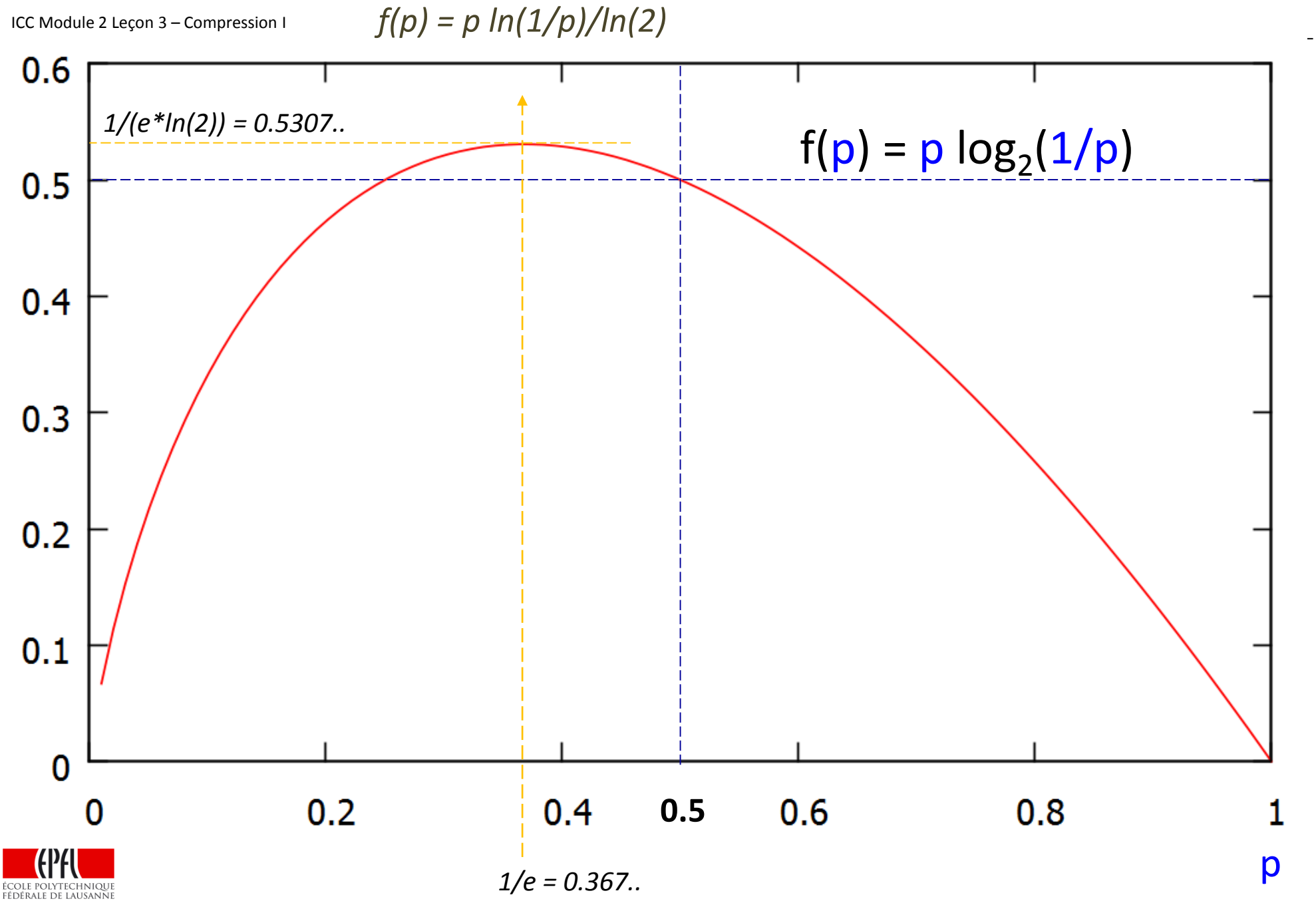
L' entropie de la séquence X est définie par

$$H(X) = p_1 \log_2 \left(\frac{1}{p_1} \right) + \ldots + p_n \log_2 \left(\frac{1}{p_n} \right)$$

Par convention, si $p_j = 0$, alors on pose $p_j \log_2 \left(\frac{1}{p_j} \right) = 0$.

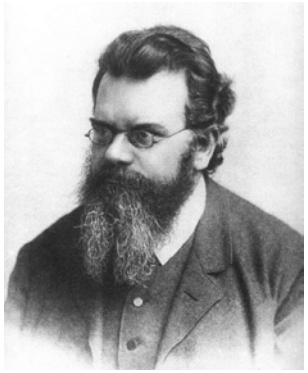
Exemple: entropie d'une séquence X avec DEUX lettres, de probabilités respectives p et (1-p)




f(p) = p ln(1/p)/ln(2)
1/(e*ln(2)) = 0.5307..
f(p) = p log2(1/p)
0.6
0.5
0.4
0.3
0.2
0.1
0
0
0.2
0.4
0.5
0.6
0.8
1
p
1/e = 0.367..

Entropie (8)

Origine en physique (Boltzmann, 1872):



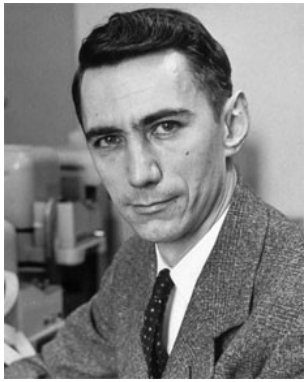
L'entropie mesure le **désordre** dans un système physique.

Ludwig Eduard Boltzmann (1844-1905)

ardent défenseur de l'existence des atomes

père de la physique statistique

Théorie de l'information (Shannon, 1948):



L'entropie mesure la **"quantité d'information"** contenue dans un signal.

Claude Edwood Shannon (1916-2001)

mathématicien, ingénieur électricien, cryptologue,

père de la théorie de l'information, jongleur...

Ou encore (comme on vient de le voir):

L'entropie est égale au **nombre moyen de questions** nécessaires pour deviner une lettre choisie au hasard dans une séquence.

Entropie (9)

Interprétation:

M A P J O F B G I L C K D E H N	$H(X) = 4$
I L F A I T B E A U A I B I Z A	$H(X) = 2.875$
A A A A A A A A A A A A A A A A	$H(X) = 0$

Plus il y a de lettres différentes, plus il y a de désordre,
plus il y a de *variété* et donc “d’information” dans le message.

Plus il y a de lettres semblables, moins il y a de désordre,
plus il y a de *redondance* et donc moins d’“information” dans le message.

Intermède

1) Lequel de ces deux mots a la plus grande entropie?

E N T R O P I E ou D E S O R D R E ?

2) Laquelle de ces communes vaudoises a la plus grande / la plus faible entropie?

A V E N C H E S, C O S S O N A Y, E C U B L E N S,
G R A N D S O N, L A U S A N N E ou M O N T R E U X ?

Entropie (10)

Une remarque, et quelques propriétés de l'entropie:

$$H(X) = p_1 \log_2 \left(\frac{1}{p_1}\right) + \ldots + p_n \log_2 \left(\frac{1}{p_n}\right)$$

Pour une probabilité d'apparition $0 \leq p \leq 1$ donnée, $\log_2(1/p) \geq 0$.

mais ça n'est pas forcément un nombre entier (ex: si $p = 1/3$).

$H(X) \geq 0$ en général, et $H(X) = 0$ si et seulement si l'ordre est total (c'est-à-dire si toutes les lettres sont les mêmes).

Si n est la taille de l'alphabet utilisé, $H(X) \leq \log_2(n)$ en général et $H(X) = \log_2(n)$ si et seulement si le désordre est total (c'est-à-dire si toutes les lettres sont différentes).

Vérifications:

$$H(X) \geq 0$$

Remarquer simplement que : $0 \leq p_j \leq 1$

D'où $1/p_j \geq 1$

D'où $\log_2(1/p_j) \geq 0$

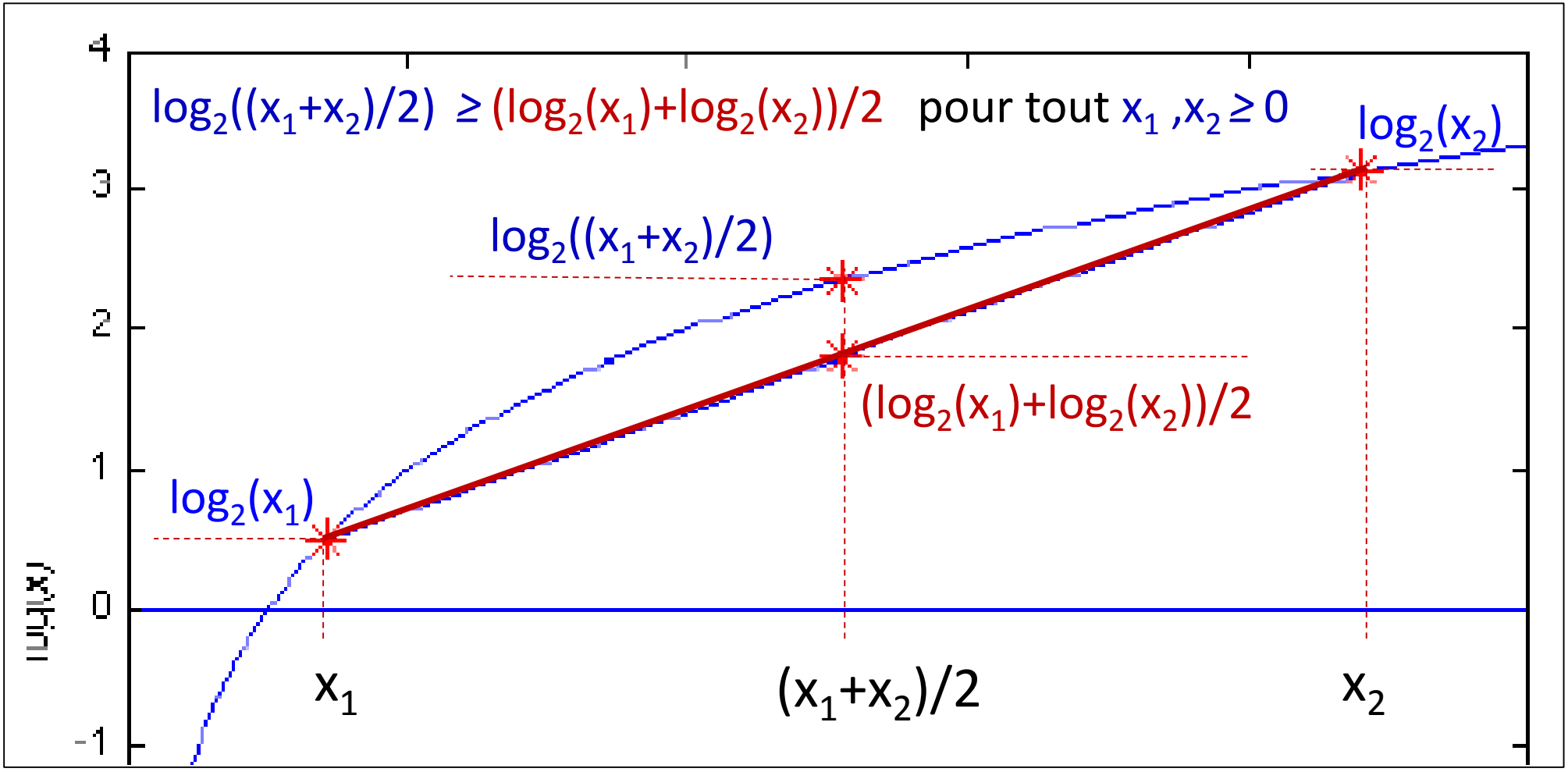
D'où $p_j \log_2(1/p_j) \geq 0$

Donc $H(X) \geq 0$.

Vérifications (2):

$H(X) \leq \log_2(n)$

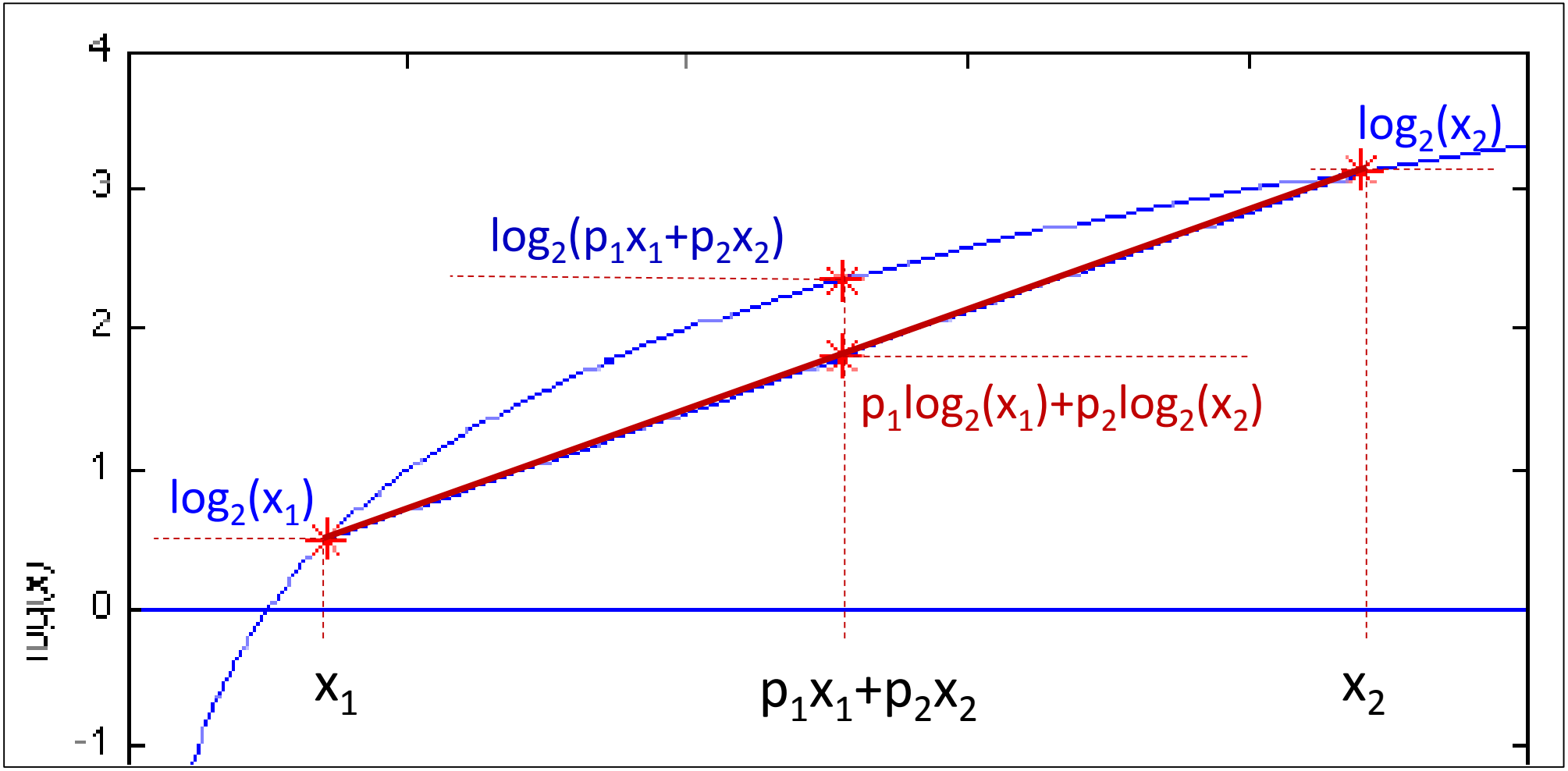
Remarquer que la fonction $f(x) = \log_2(x)$ est concave pour $x \geq 0$:



Vérifications (3):

Plus généralement, si $0 \leq p_1, \; p_2 \leq 1$ et $p_1 + p_2 = 1$, alors:

$\log_2(p_1x_1+p_2x_2) \geq p_1\log_2(x_1)+p_2\log_2(x_2)$ pour tout $x_1 , x_2 \geq 0$

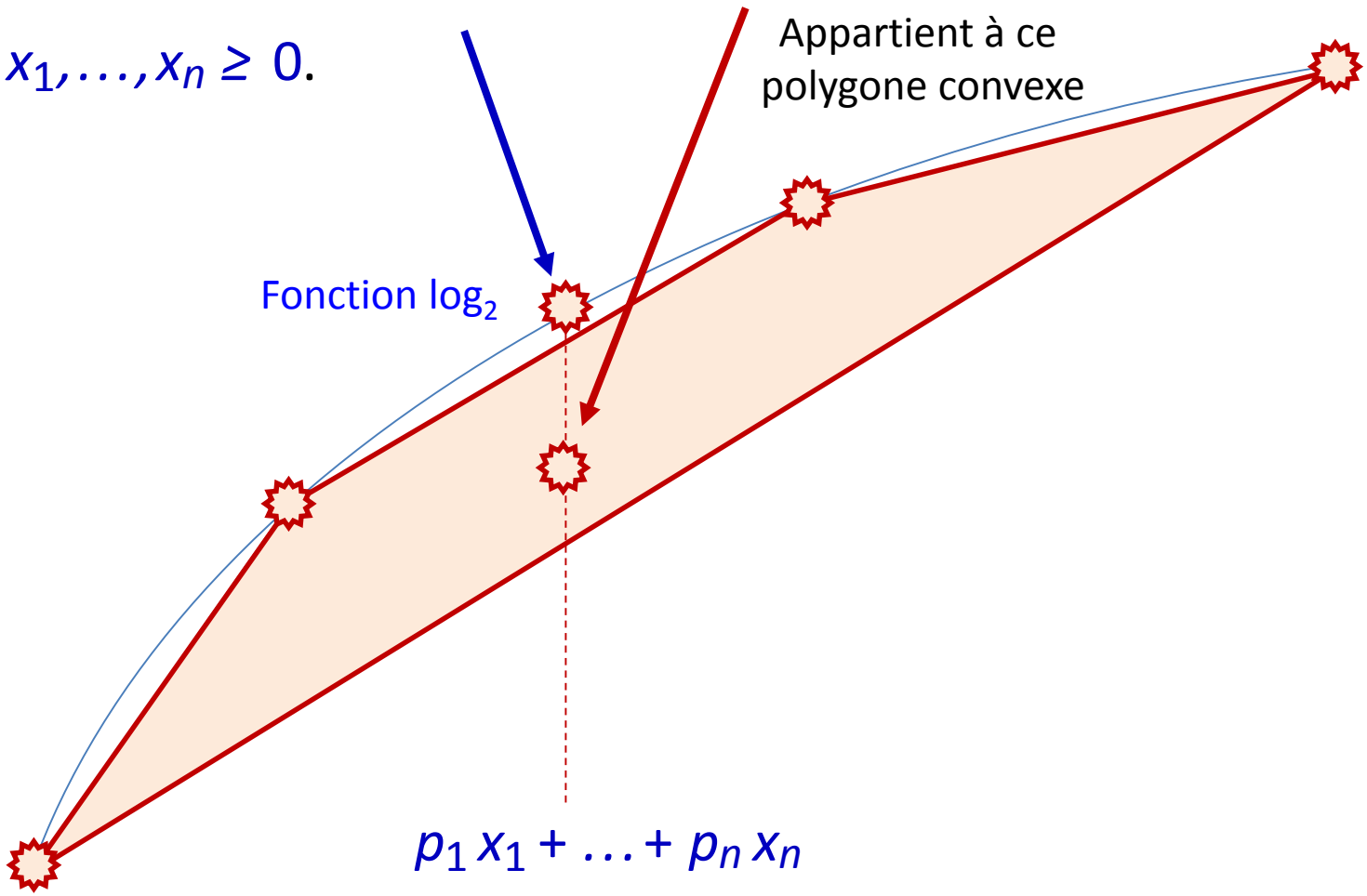


Vérifications: (4)

Plus généralement encore, si $0 \leq p_j \leq 1$ et $p_1 + \ldots + p_n = 1$, alors

$$\log_2(p_1 x_1 + \ldots + p_n x_n) \geq p_1 \log_2(x_1) + \ldots + p_n \log_2(x_n)$$

pour tous $x_1, \ldots, x_n \geq 0$.



Vérifications: (5)

Plus généralement encore, si $0 \leq p_j \leq 1$ et $p_1 + \ldots + p_n = 1$, alors

$$\log_2(p_1 x_1 + \ldots + p_n x_n) \geq p_1 \log_2(x_1) + \ldots + p_n \log_2(x_n)$$

pour tous $x_1, \ldots, x_n \geq 0$. *(inégalité de Jensen)*

En appliquant cette inégalité avec $x_j = 1/p_j$ on obtient finalement:

$$H(X) = p_1 \log_2(1/p_1) + \ldots + p_n \log_2(1/p_n) \leq \log_2(p_1/p_1 + \ldots + p_n/p_n)$$

$$H(X) \leq \log_2(\ 1\ + \ldots + \ 1\ )$$

$$H(X) \leq \log_2(n)$$

Information, Calcul et Communication

Entropie

Mais à quoi tout cela peut-il bien servir?

Compression sans pertes

Comme on l'a vu dans l'introduction, la compression de données permet:

- de réduire l'espace-mémoire utilisé lors du stockage de données
- de réduire le temps de transmission et les problèmes de congestion lors de la transmission des données

Le principe de base derrière la **compression** de données est la **supression de la redondance** contenue dans ces données; les lettres ou les mots qui reviennent souvent dans un message sont abrégés.

Lorsqu'on parle de compression **sans pertes**, on demande que le message d'origine puisse être récupéré dans son intégralité après le processus de compression.

L'opération doit être réversible

Compression sans pertes

Exemples d'algorithmes de compression sans pertes:

Langage SMS: "slt", "tqt", "lol", etc; les mots qui reviennent souvent sont réduits à de courtes séquences de lettres.

Acronymes : EPFL, UNIL, MT, EL, etc...

Code Morse: "e" = **.** "a" = **.–** "s" = **...** "t" = **–**

tandis que "x"= **–..–** "z"= **– – ..**

Le concept d'entropie permet de comprimer des données de manière *systématique et optimale*.

Compression sans pertes

Revenons sur la plage à Ibiza, et supposons que vous vouliez envoyer ce message par SMS à un ami:

I L F A I T B E A U A I B I Z A

Le but du jeu est maintenant de *minimiser le nombre de bits* nécessaires pour représenter ce message.

→ **Il faut le coder plus efficacement que le code ASCII**

Remarques importantes:

La personne qui reçoit votre message doit être capable de le **décoder**!

Avant l'envoi du message, l'expéditeur et le destinataire doivent se mettre d'accord sur un dictionnaire commun de motifs binaires pour chaque lettre:

Exemple: A='01', B='10', etc…

Compression sans pertes

I L F A I T B E A U A I B I Z A

- **Solution N° 1:** Utiliser le code ASCII (cf. leçon 1.1): chaque lettre est représentée par 8 bits; $16 \times 8 = 128$ bits sont donc nécessaires pour représenter ce message.

- **Solution N° 2:** Remarquer que le message à représenter n'est composé que de 16 lettres: on n'a donc besoin que de 4 bits par lettre ($2^4 = 16$); et donc $16 \times 4 = 64$ bits au total.

- **Solution N° 3:** Remarquer que certaines lettres se répètent: il n'y a en fait que 9 lettres différentes. Et certaines sont plus fréquentes que d'autres...

Comment procéder?

Algorithme de Shannon-Fano

Pionnier de la théorie de l'information
Robert Fano (1917-2016)

Professeur au MIT

I L F A I T B E A U A I B I Z A

Revenons à notre jeu de questions:

lettre	I	A	B	L	F	T	E	U	Z
nb d'apparitions	4	4	2	1	1	1	1	1	1
nb de questions	2	2	3	4	4	4	4	4	4

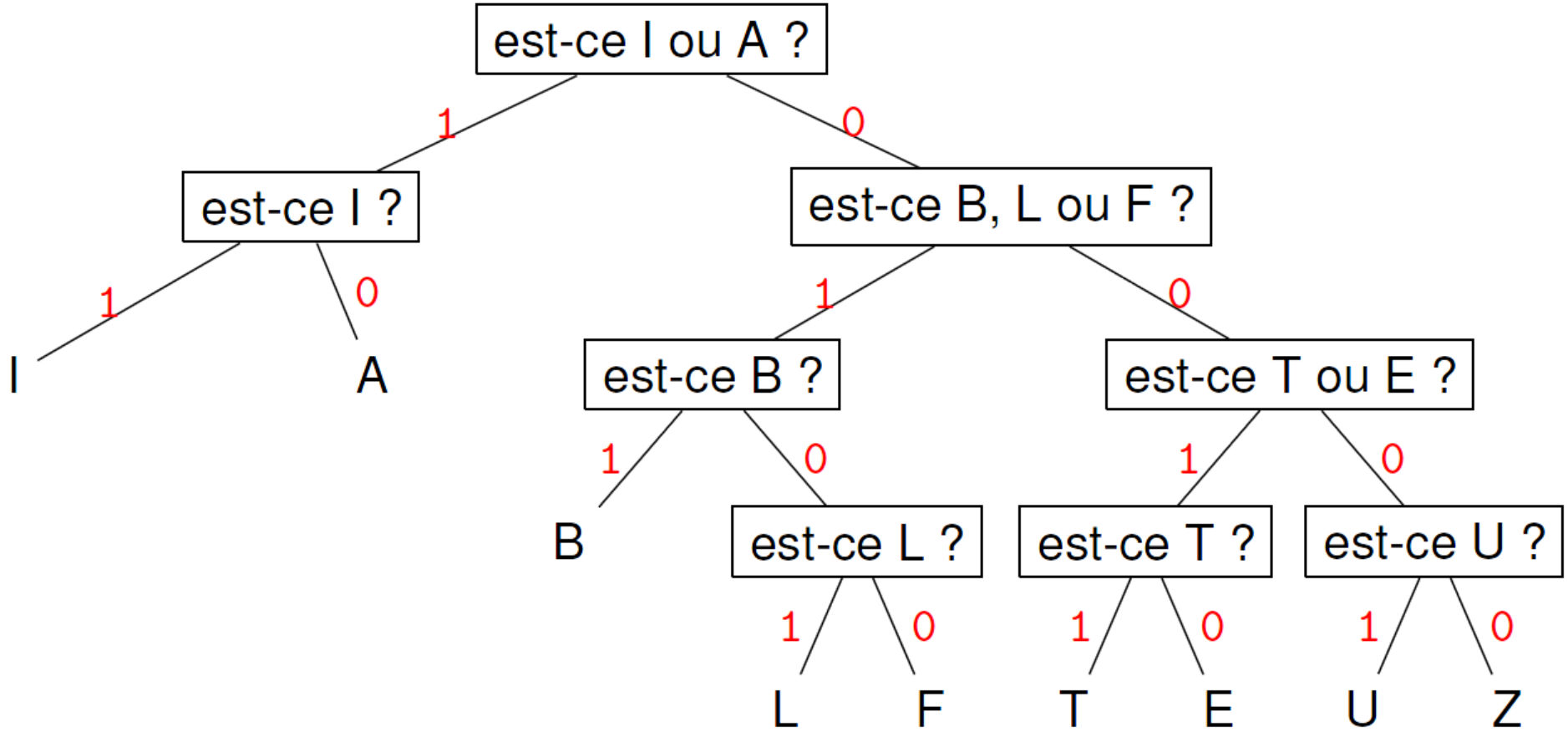
Pour attribuer une séquence de bits (un "mot de code") à chaque lettre, nous suivons les deux règles ci-dessous:

Règle n^0 1: le nombre de bits attribués à chaque lettre est égal au nombre de questions nécessaires pour la deviner.

Règle n^0 2: les bits 0 ou 1 sont attribués en fonction des réponses obtenues aux questions. Plus précisément, le bit N° j est égal à 1 ou 0 selon que la réponse à la j^e question était oui ou non.

I L F A I T B E A U A I B I Z A

lettre	I	A	B	L	F	T	E	U	Z
nb d'apparitions	4	4	2	1	1	1	1	1	1



Algorithme de Shannon-Fano

lettre	I	A	B	L	F	T	E	U	Z
mot de code	11	10	011	0101	0100	0011	0010	0001	0000

Le message “IL FAIT BEAU A IBIZA" s’écrit ainsi:

11 0101 0100 10 . . .

Observation N° 1:
Dans le dictionnaire, **aucun mot de code n’est le préfixe d’un autre.**
Donc le message reçu est parfaitement décodable, et ceci au fur et à mesure....

...sans attendre que la fin du message soit reçue

Algorithme de Shannon-Fano

lettre	I	A	B	L	F	T	E	U	Z
mot de code	11	10	011	0101	0100	0011	0010	0001	0000

Observation n^0 2: Le nombre de bits utilisés est égal à:

$$8 \times 2 + 2 \times 3 + 6 \times 4 = 16 + 6 + 24 = 46$$

Par rapport à la représentation 'brute' qui nécessite 64 bits, on économise donc environ 25% de bits.

Vu que le message a 16 lettres, le nombre moyen de bits utilisés par lettre est égal à $46/16 = 2.875 = H(X) =$ entropie du message !

Nous montrerons la semaine prochaine que l'entropie est une *borne inférieure* au nombre moyen de bits par lettre dont on a besoin: aucun autre algorithme de codage ne permet de descendre plus bas.

C'est l'énoncé du premier théorème de Shannon.

Algorithme de Shannon-Fano

Avec cet algorithme, on obtient les performances suivantes:

Dans un texte:

15-25% de réduction avec les lettres, comme vu précédemment
jusqu'à 75% de réduction si on utilise les mots à la place des lettres!

Dans un fichier de données: 25-30% de réduction
(p.ex: fichiers archivés .zip pour l'envoi d'emails)

Conclusion temporaire

L'*entropie* mesure la "quantité d'information" présente dans un message.

Pour représenter un message par une séquence de bits, l'algorithme de Shannon-Fano utilise un nombre de bits par lettre égal à l'entropie du message (*dans un cas particulier seulement*).

La semaine prochaine:

algorithme de Shannon-Fano (bis)

analyse de performance – théorème de Shannon

compression avec pertes

Annexe: entropie - définitions plus générales

La formule de l'entropie reste valable même si les pj ne sont pas estimés sur la séquence elle-même, mais proviennent d'une estimation sur un texte plus grand ou même de notre connaissance du procédé qui a généré un mot X.

Tout ce qui compte ce sont simplement les probabilités pj de chaque lettre (peut importe d'où elles viennent).

L'entropie calculée dans ce cours est simplement celle du « jeu » consistant à tirer une lettre du « mot » X au hasard.

Pour de vraies séquences, en tant que telles, la définition de l'entropie est bien plus complexe et intègre toutes les dépendances entre sous-séquences de lettres, i.e. les probabilités de toutes les sous-séquences de n lettres. Mais ceci nécessite des outils mathématiques plus avancés (probabilités conditionnelles et limites).

Exercices

1. Considérons le mot:

D E S O R D R E

a) Calculer l'entropie de ce mot.

b) Trouver une représentation binaire de ce mot.

c) Combien de bits sont-ils nécessaires pour représenter ce mot?

2. Considérons l'expression suivante (à nouveau, sans les espaces):

INFORMATION CALCUL ET COMMUNICATION

a)Calculer l'entropie de cette séquence.

b)Décrire la suite de questions optimales à poser pour deviner une lettre choisie au hasard dans cette séquence (attention, dans ce cas, toutes les probabilités ne sont pas des puissances inverses de 2).

c)A l'aide de ces questions, trouver une représentation binaire de cette séquence. Combien de bits sont-ils nécessaires?