Chapitre IV

Codage par prédiction

Dans cette dernière leçon sur la compression, nous verrons :

- comment les probabilités peuvent permettre d'effectuer de la prédiction.
- pourquoi la prédiction peut être en mesure d'améliorer la compression.
- deux méthodes de codage prédictif.

1 Prédiction

Dans ce que nous avons vu pour l'instant :

- Le codage entropique tient compte de la fréquence de répétition des symboles, mais suppose indépendant les symboles consécutifs.
- Le codage par dictionnaire tient compte des répétitions contenues dans les motifs, mais sans tenir compte ni de leurs fréquences, ni des relations entre motifs.

L'idée du codage prédictif est d'utiliser les probabilités conditionnelles de symboles consécutifs afin de prévoir les codes à venir.

C'est une approche que nous utilisons naturellement lorsque nous lisons ne texte :

- nous reconnaissons immédiatement les mots les plus courants.
  les mots plus rare ou inconnu sont décodés plus lentement (exemple : apophthegme, argyraspide, héméralopie, immarcescible, irréfragable, kéraunographique, nychthémérique, sphragistique, synallagmatique, ...)
- lors de la lecture d'un mot, nous reconnaissons rapidement les phonèmes les plus courants (se, de, pa, la, te, une, eau, ...).

Donnons quelques exemples de loi statistique pour la langue anglaise.

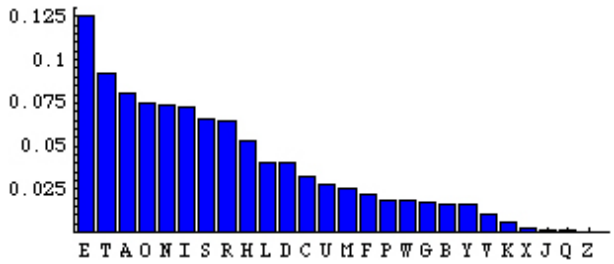


FIGURE IV.1 – Fréquence des lettres en anglais.

Exemple 1 : indépendante et identiquement distribuée (iid)

> XFOML RXKHRJFFJUJ ZLPWCFWKCYJ FFJEYVKCQSGXYD QPAAMKBZAACIBZLHJQD

Les lettres sont tirées indépendamment les unes des autres (= sans tenir compte des tirages déjà faits), et toutes les lettres ont la même chance de tirage.

Défaut : les voyelles sont sous-représentées, aucun mot n'est lisible.

Exemple 2 : loi d'ordre 0

> OCRO HLI RGWR NMIELWIS EU LL NBNESEBYA TH EEI ALHENHTTPA OOBTTVA NAH BRL

Les lettres sont tirées avec leurs fréquences d'apparition normales dans une phrase (par exemple, Pr[E] = 12.56%, Pr[T] = 9.16%, Pr[A] = 8.08%, ..., Pr[Z] = 0.07%), voir sur la figure IV.1 la fréquence des lettres dans la langue anglaise.

Défaut : si la fréquence des voyelles est normale, presque aucune syllabe n'est identifiable.

Exemple 3 : loi d'ordre 1

> ON IE ANTSOUTINYS ARE T INCTORE ST BE S DEAMY ACHIN D ILONASIVE TUCOOWE AT TEASONARE FUSO TIZIN ANDY TOBE SEACE CTISBE

La lettre suivante est tirée en fonction de la probabilité qu'elle a de suivre la lettre précédente. Cela revient ouvrir au hasard un passage dans un livre et à rechercher la lettre précédente, et lorsqu'on la trouve, choisir la lettre qui la suit.

Défaut : si les mots deviennent lisible, ils ne ressemblent toujours pas à de l'anglais.

Exemple 4 : loi d'ordre 2

> IN NO IST LAT WHEY CRATICT FROURE BERS GROCID PONDENOME OF DEMONSTURES OF THE REPTAGIN IS REGOACTIONA OF CRE

La lettre suivante est tirée en fonction de la probabilité qu'elle a de suivre les deux lettres précédentes.
Défaut : si les mots commencent à ressembler à de l'anglais, très peu sont identifiables.

Exemple 5 : loi d'ordre 3

> THE GENERATED JOB PROVIDUAL BETTER TRAND THE DISPLAYED CODE, ABOVERY UPONDULTS WELL THE CODERST IN THESTICAL IT DO HOCK BOTHE MERG. INSTATES CONS ERATION. NEVER ANY OF PUBLE AND TO THEORY. EVENTIAL CALLEGAND TO ELAST BENERATED IN WITH PIES AS IS WITH THE

La lettre suivante est tirée en fonction de la probabilité qu'elle a de suivre les trois lettres précédentes.

Défaut : si les mots ressemblent maintenant vraiment à de l'anglais (certains sont même corrects), il n'y a aucune structuration.

Donc, les statistiques d'ordre supérieur permettent de "capturer" un certain nombre de caractéristiques d'une langue, sa structure reste plus complexe.

Ceci peut se mesurer par le calcul de l'entropie : loi d'ordre 0 : $\log_2 27 = 4.76$ bits/lettre, d'ordre 1 : 4.03 bits/lettre, d'ordre 2 : 3.32 bits/lettre, d'ordre 3 : 3.1 bits/lettre, d'ordre 4 : 2.8 bits/lettre.

Des expériences semblent montrer que l'entropie de l'anglais est de 1.62 bits/lettre.

2 Théorie de l'information

Quelle sont les conséquences de ces dépendances sur les résultats que nous avions trouvé en théorie de l'information ?
Pour aborder cet aspect, il est nécessaire d'aborder deux points :

- Comment exprimer la dépendance entre deux symboles produits dans une chaîne ?
- Comment exprimer les dépendances sur une chaîne complète de symboles ?

avant d'en exprimer les conséquences.

Remarque : ne pas vous laisser effrayer par les notations, ce qui est dit reste très simple.

2.1 Probabilité jointe

Si X et Y sont deux variables aléatoires définies sur le même espace $\mathcal{S}$, on appelle probabilité jointe la probabilité :

$$p_{XY}(x_i, y_j) = \Pr\left[\{X = x_i\} \cap \left\{Y = y_j\right\}\right] = \Pr\left[X = x_i; Y = y_j\right]$$

La somme de p_{XY} sur toutes les valeurs possibles de (X, Y) est 1 :

$$\sum_i \sum_j p_{XY}(x_i, y_j) = 1$$

Chaque variable a également sa propre loi (ou loi marginale)

$$p_X(x_i) = \Pr\left[\{X = x_i\}\right] = \Pr\left[X = x_i\right]$$
$$p_Y(y_i) = \Pr\left[\{Y = y_i\}\right] = \Pr\left[Y = y_i\right]$$

La loi jointe a les propriétés suivantes :

$$p_X(x_i) = \sum_j p_{XY}(x_i, y_j) \text{ car } \cup_j \left\{X = x_i, Y = y_j\right\} = \{X = x_i\}$$
$$p_Y(y_j) = \sum_i p_{XY}(x_i, y_j) \text{ car } \cup_i \left\{X = x_i, Y = y_j\right\} = \left\{Y = y_j\right\}$$

Exemple 1

Deux cartes sont tirées au hasard dans un jeu de cartes. Soit X le nombre de piques et Y le nombre de coeurs parmi ces cartes. Quelle est la loi jointe ? Compter sachant que : Pr[♣] = Pr[♦] = Pr[♥] = Pr[♠] = 1/4.

$p_{XY}(x, y)$	$x = 0$	$x = 1$	$x = 2$
$y = 0$	4/16	4/16	1/16
$y = 1$	4/16	2/16	0
$y = 2$	1/16	0	0

On vérifie que $\sum_x \sum_y p_{XY}(x, y) = 1$.

La première loi marginale est :

x	0	1	2
$p_X(x)$	9/16	6/16	1/16

On vérifie que $p_X(x) = \sum_y p_{XY}(x, y)$.

La seconde loi marginale est :

y	0	1	2
$p_Y(y)$	9/16	6/16	1/16

On vérifie que $p_Y(y) = \sum_x p_{XY}(x, y)$.

Exemple 2

Deux dés sont tirés au hasard. Soit X la valeur du premier dé et Y la valeur la plus grande des deux. Quelle est la loi jointe ? (faire la table contenant l'ensemble des possibilités et compter).

$p_{XY}(x, y)$	$x = 1$	$x = 2$	$x = 3$	$x = 4$	$x = 5$	$x = 6$
$y = 1$	1/36	0	0	0	0	0
$y = 2$	1/36	2/36	0	0	0	0
$y = 3$	1/36	1/36	3/36	0	0	0
$y = 4$	1/36	1/36	1/36	4/36	0	0
$y = 5$	1/36	1/36	1/36	1/36	5/36	0
$y = 6$	1/36	1/36	1/36	1/36	1/36	6/36

On vérifie que $\sum_x \sum_y p_{XY}(x, y) = 1$.

x	1	2	3	4	5	6
$p_X(x)$	1/6	1/6	1/6	1/6	1/6	1/6

On vérifie que $p_X(i) = \sum_y p_{XY}(x, y)$.

y	1	2	3	4	5	6
$p_Y(y)$	1/36	3/36	5/36	7/36	9/36	11/36

On vérifie que $p_Y(y) = \sum_x p_{XY}(x, y)$.

2.2 Probabilité conditionnelle

Une probabilité conditionnelle est la probabilité d'un évènement sous condition de la condition de la réalisation d'un autre évènement.
Si $\Pr[B] > 0$, elle est définie par :

$$\Pr[A \mid B] = \frac{\Pr[A \cap B]}{\Pr[B]}$$

Exemple 3

En reprenant l'exemple 2, $\Pr[A \cap B]$ représente la loi jointe de (X, Y), à savoir, on prend $A = \{X = x\}$ et $B = \{Y = y\}$

1. quelle est la probabilité que la valeur du premier dé soit 1 sachant que la plus grande des deux est 3.

   $\Pr[X = 1 \mid Y = 3] = \frac{p_{XY}(1,3)}{p_Y(3)} = \frac{1/36}{5/36} = \frac{1}{5}$

2. quelle est la probabilité que la valeur la plus grande des deux soit 5, sachant que le premier dés est 4,

$$\Pr[Y = 5 \mid X = 4] = \frac{p_{XY}(4,5)}{p_X(4)} = \frac{1/36}{1/6} = \frac{1}{6}$$

Si les évènements sont indépendants $\Pr[A \cap B] = \Pr[A].\Pr[B]$.

En conséquence,

$$\Pr[A \mid B] = \frac{\Pr[A \cap B]}{\Pr[B]} = \frac{\Pr[A].\Pr[B]}{\Pr[B]} = \Pr[A]$$

$\Rightarrow$ la connaissance de B n'apporte aucune connaissance sur A.

Exemple 4

La loi jointe du tirage d'un dé suivi d'un tirage à pile ou face est :

$p_{XY}(x,y)$	1	2	3	4	5	6
P	1/12	1/12	1/12	1/12	1/12	1/12
F	1/12	1/12	1/12	1/12	1/12	1/12

Les lois marginales sont :

x	1	2	3	4	5	6
$p_X(x)$	1/6	1/6	1/6	1/6	1/6	1/6

et

y	P	F
$p_Y(y)$	1/2	1/2

On a bien $p_{XY}(x,y) = p_X(x).p_Y(y)$.

1. la probabilité que le dé fasse 3 si la pièce est pile est :

$$\Pr[X = 3 \mid Y = P] = \frac{p_{XY}(3,P)}{p_Y(P)} = \frac{1/12}{1/2} = \frac{1}{6} = \Pr[X = 3]$$

2. la probabilité que la pièce soit pile si le dé fait 3 est :

$$\Pr[Y = F \mid X = 3] = \frac{p_{XY}(3,P)}{p_X(3)} = \frac{1/12}{1/6} = \frac{1}{2} = \Pr[Y = P]$$

Comme $A \cap B = B \cap A$, si $\Pr[A] > 0$ et $\Pr[B] > 0$, on en déduit :

$$\Pr[A \cap B] = \Pr[A \mid B].\Pr[B] = \Pr[B \mid A].\Pr[A]$$

qui permet d'exprimer Pr[$A \mid B$] en fonction de Pr[$B \mid A$] (Bayes).

Exemple 5 : en reprenant l'exemple 3.

1. la probabilité que le premier dé soit 3 sachant que le maximum est 3 est : $\Pr[X = 3 \mid Y = 3] = \frac{p_{XY}(3,3)}{p_Y(3)} = \frac{3/36}{5/36} = \frac{3}{5}$
2. la probabilité que le maximum des deux dés soit 3 sachant que le premier dé a fait 3 est : $\Pr[Y = 3 \mid X = 3] = \frac{p_{XY}(3,3)}{p_X(3)} = \frac{3/36}{1/6} = \frac{1}{2}$

On a bien :

- $\Pr[X = 3; Y = 3] = p_{XY}(3,3) = \frac{3}{36} = \frac{1}{12}$
- $\Pr[X = 3 \mid Y = 3].Pr[Y = 3] = \frac{3}{5}.\frac{5}{36} = \frac{3}{36} = \frac{1}{12}$
- $\Pr[Y = 3 \mid X = 3].Pr[X = 3] = \frac{1}{2}.\frac{1}{6} = \frac{1}{12}$

Pour toute partition $\{A_i\}_{i=1\ldots n}$ de $(\Omega, \mathcal{A})$ (à savoir $\forall i \neq j, A_i \cap A_j = \emptyset$ et $\cup_i A_i = \Omega$), on a :

$$\sum_i \Pr[A_i \mid B] = 1$$

En effet, $\Pr[B] = \sum_i \Pr[B \cap A_i] = \sum_i \Pr[B] \Pr[A_i \mid B] = \Pr[B] \sum_i \Pr[A_i \mid B]$.

$\Rightarrow$ Pr est une mesure de probabilité sur $\{A_i|B\}$.

La somme des probabilités conditionnellement à un évènement est égale à 1.

Exemple 6

On reprend l'exemple 2 et l'on calcule les probabilités conditionnelles :

$\Pr[X = x \mid Y = y]$

$y\backslash x$	1	2	3	4	5	6
1	1	0	0	0	0	0
2	1/3	2/3	0	0	0	0
3	1/5	1/5	3/5	0	0	0
4	1/7	1/7	1/7	4/7	0	0
5	1/9	1/9	1/9	1/9	5/9	0
6	1/11	1/11	1/11	1/11	1/11	6/11

La somme des probabilités en ligne est toujours 1.

$\Pr[Y = y \mid X = x]$

$y\backslash x$	1	2	3	4	5	6
1	1/6	0	0	0	0	0
2	1/6	2/6	0	0	0	0
3	1/6	1/6	3/6	0	0	0
4	1/6	1/6	1/6	4/6	0	0
5	1/6	1/6	1/6	1/6	5/6	0
6	1/6	1/6	1/6	1/6	1/6	1

La somme des probabilités en colonne est toujours 1.

2.3 Entropie conjointe

Définition 16 (Entropie conjointe)**.** L'entropie conjointe $H(X, Y)$ d'un paire de variable aléatoire discrète (X, Y) de loi jointe $p_{XY}(x, y)$ est définie comme l'entropie de la loi jointe :

$$H(X, Y) = -\sum_x \sum_y p_{XY}(x, y) \log_2 p_{XY}(x, y)$$

Exemple 7

$$\begin{aligned} H(X,Y) = -\Bigg(&15 \times 0.\log_2 0 + 16 \times \frac{1}{36}\log_2\frac{1}{36} + \frac{2}{36}\log_2\frac{2}{36} \\ &+\frac{3}{36}\log_2\frac{3}{36} + \frac{4}{36}\log_2\frac{4}{36} + \frac{5}{36}\log_2\frac{5}{36} + \frac{6}{36}\log_2\frac{6}{36}\Bigg) \\ &= 4,007 \text{ bits} \end{aligned}$$

Donc, il faut environ 4 bits pour stocker la paire de variable aléatoire (X, Y).

Proposition 11 (Entropie conjointe)**.** *Pour toute paire de variables aléatoires discrètes* (X, Y)*, on a :*

$$H(X,Y) \leq H(X) + H(Y)$$

avec égalité si et seulement si les variables aléatoires X et Y sont indépendantes.

Démonstration:

$H(X) + H(y) = -\sum_x p_X(x)\log_2 p_X(x) - \sum_y p_Y(y)\log_2 p_Y(y)$

$= -\left(\sum_x \sum_y p_{XY}(x,y)\log_2 p_X(x) + \sum_x \sum_y p_{XY}(x,y) p_Y(y)\log_2 p_Y(y)\right)$

$= -\sum_x \sum_y p_{XY}(x,y)\log_2 (p_X(x).p_Y(y))$

Or, on a vu (cf leçon 1) que $\sum_{i=1}^n p_i \log_2 q_i \leq \sum_{i=1}^n p_i \log_2 p_i$ (avec égalité lorsque $p_i = q_i$). En l'appliquant à l'équation ci-dessus avec $p_i = p_{XY}(x,y)$ et $q_i = p_X(x).p_Y(y)$, on obtient l'inégalité suivante :

$-\sum_x \sum_y p_{XY}(x,y)\log_2 (p_X(x).p_Y(y)) \geq -\sum_x \sum_y p_{XY}(x,y)\log_2 p_{XY}(x,y)$.

D'où $H(X) + H(Y) \geq H(X,Y)$.

L'égalité est réalisée lorsque $p_{XY}(x,y) = p_X(x).p_Y(y)$, donc lorsque X et Y sont indépendants. □

Interprétation

Lorsque deux variables aléatoires (X, Y) sont dépendantes, alors le codage du couple (X, Y) nécessite toujours moins de bits que chaque variable aléatoire X et Y prise indépendamment.

Exemple 8

En reprenant l'exemple 2 :

$$H(X) = -\sum_x p_X(x)\log_2 p_X(x) = -6 \times \frac{1}{6}\log_2\frac{1}{6} = 2.58 \text{ bits}$$

$$H(Y) = -\sum_y p_Y(y)\log_2 p_Y(y)$$

$= -\frac{1}{36}\log_2\frac{1}{36} - \frac{2}{36}\log_2\frac{2}{36} - \frac{3}{36}\log_2\frac{3}{36} - \frac{4}{36}\log_2\frac{4}{36} - \frac{5}{36}\log_2\frac{5}{36} - \frac{6}{36}\log_2\frac{6}{36}$
$= 2.32$ bits
$H(X) + H(Y) = 2.58 + 2.32 = 4.9$ bits, ce qui est 0.9 bit en plus que l'entropie du couple $H(X, Y)$.
En conséquence, un codage que tient compte de la dépendance entre les variables X et Y sera plus efficace qu'un codage qui les considère indépendantes.

2.4 Entropie conditionnelle

Définition 17 (Entropie conditionnelle)**.** L'entropie conditionnelle $H(X, Y)$ d'un paire de variable aléatoire discrète (X, Y) de loi jointe $p_{XY}(x, y)$ est définie comme l'espérance de l'entropie de la loi conditionnelle :
$H(Y|X) = \sum_x p_X(x)H(Y|X = x) = -\sum_x p_X(x)\sum_y p_{Y|X}(x, y)\log_2 p_{Y|X}(x, y)$

Exemple 9

En reprenant l'exemple 2 (et ses lois conditionnelles dans l'exemple 6),
$H(Y|X = 1) = -6 \times \frac{1}{6}\log_2\frac{1}{6} = 2,58$ bits.
$H(Y|X = 2) = -4 \times \frac{1}{6}\log_2\frac{1}{6} - \frac{2}{6}\log_2\frac{2}{6} = 2,25$ bits.
$H(Y|X = 3) = -3 \times \frac{1}{6}\log_2\frac{1}{6} - \frac{3}{6}\log_2\frac{3}{6} = 1,79$ bits.
$H(Y|X = 4) = -2 \times \frac{1}{6}\log_2\frac{1}{6} - \frac{4}{6}\log_2\frac{4}{6} = 1,25$ bits.
$H(Y|X = 5) = -\frac{1}{6}\log_2\frac{1}{6} - \frac{5}{6}\log_2\frac{5}{6} = 0,65$ bits
$H(Y|X = 6) = -1\log_2 1 = 0$ bits.
$H(Y|X) = 2,58.\frac{1}{6} + 2,25.\frac{1}{6} + 1,79.\frac{1}{6} + 1,25.\frac{1}{6} + 0,65.\frac{1}{6} + 0.\frac{1}{6} = 1,42$ bits.
L'entropie conditionnelle de Y est inférieure de 0.9 bit à l'entropie de Y.

Théorème 12 (lien entre entropies)**.** $H(X, Y) = H(X) + H(Y|X)$

Démonstration:
On utilise le fait que $p_{XY} = p_X.p_{Y|X}$.

$$\begin{aligned} H(X,Y) &= -\sum_x\sum_y p_{XY}(x,y)\log_2 p_{XY}(x,y) \\ &= -\sum_x\sum_y p_{XY}(x,y)\log_2\left[p_X(x).p_{Y|X}(x,y)\right] \\ &= -\sum_x\sum_y p_{XY}(x,y)\log_2 p_X(x) - \sum_x\sum_y p_{XY}(x,y)\log_2 p_{Y|X}(x,y) \\ &= -\sum_x p_X(x,y)\log_2 p_X(x) - \sum_x p_X(x)\sum_y p_{Y|X}(x,y)\log_2 p_{Y|X}(x,y) \\ &= H(X) + H(Y|X) \end{aligned}$$

□

Corollaire 13 (chaîne d'entropie)**.** $H(X, Y|Z) = H(X|Z) + H(Y|X, Z)$

Démonstration:
Conséquence directe de $p_{X,Y|Z} = p_{X|Z} \times p_{Y|X,Z} \Leftrightarrow \frac{p_{XYZ}}{p_Z} = \frac{p_{XZ}}{p_Z} \times \frac{p_{XYZ}}{p_{XZ}}$. □

Théorème 14 (réduction de l'entropie par conditionnement)**.** *Pour toute paire de variables aléatoires* (X, Y)*, on a :*

$$H(X|Y) \leq H(X)$$

l'égalité se réalisant lorsque X et Y sont indépendants.

Démonstration:
D'après le théorème précédent $H(Y|X) = H(X, Y) - H(X)$.
Or, on a vu que $H(X, Y) \leq H(X) + H(Y)$.
Donc $H(Y|X) \leq H(X) + H(Y) - H(X) = H(Y)$ où l'égalité tient lorsque X et Y sont indépendants. □

Interprétation

L'entropie conditionnelle est toujours inférieure à l'entropie. Ce résultat avait déjà été donné dans l'exemple 9.
En conséquence, si les variables X et Y ne sont pas indépendantes, et que l'on connaît le contexte Y dans lequel est X, alors on a toujours besoin de moins de bit pour coder X que si l'on ne connaissait pas le contexte.

2.5 Information mutuelle

Définition 18 (Entropie relative (ou distance de Kullback-Leiber))**.** La distance entre deux lois de $p(x)$ et $q(x)$ est définie par :

$$D(p||q) = \sum_x p(x) \log_2 \frac{p(x)}{q(x)}$$

On utilise les conventions suivantes : $0 \log(0/0) = 0, 0 \log(0/q) = 0$ et $p \log(p/0) = \infty$. En conséquence, s'il y a un symbole x tel que $p(x) > 0$ et $q(x) = 0$ alors $D(p||q) = 0$.

Définition 19 (information mutuelle)**.** Soit deux variables aléatoires X et Y de loi jointe p_XY, et de loi marginale respective p_X et p_Y, alors l'information mutuelle est l'entropie relative entre la loi jointe et le loi produit, *i.e.*

$$I(X; Y) = D(p(x, y)||p(x)p(y)) = \sum_x \sum_y p_{XY}(x, y) \log_2 \frac{p_{XY}(x, y)}{p_X(x).p_Y(y)}$$

L'information de mutuelle permet de mesurer la quantité d'information commune aux deux variables aléatoires portée par la loi jointe (i.e. celle qui tient compte de la dépendance entre les deux lois) par rapport à la loi produit (i.e. celle qui suppose que les deux variables aléatoires sont indépendantes).

Théorème 15 (lien entre information mutuelle et entropie)**.** *On a les relations suivantes :*

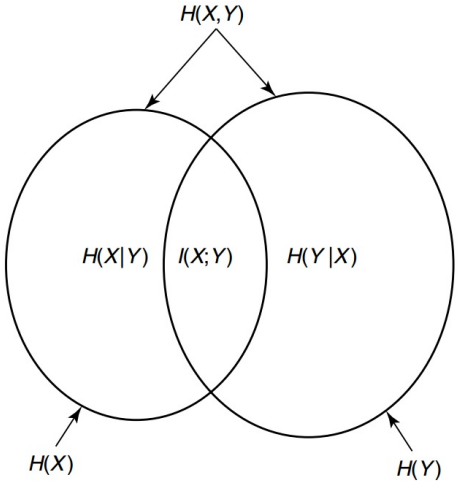
1. $I(X;Y) = H(X) - H(X|Y)$
2. $I(X;Y) = H(Y) - H(Y|X)$
3. $I(X;Y) = H(X) + H(Y) - H(X,Y)$
4. $I(X;Y) = I(Y;X)$
5. $I(X;X) = H(X)$

Démonstration:

1. obtenu en utilisant $p_{XY} = p_{X|Y}.p_Y$ dans la définition de $I(X;Y)$.
2. obtenu en utilisant $p_{XY} = p_{Y|X}.p_X$ dans la définition de $I(X;Y)$.
3. obtenu en utilisant la relation $H(X;Y) = H(X) + H(Y|X)$ dans 1 ou 2.
4. par commutativité de la multiplication ($p_X.p_Y = p_Y.p_X$).
5. car $p_{XX}(x,x) = p_X(x)$, donc :
$$\sum_x \sum_x p_X(x) \log_2 \frac{p_X(x)}{p_X(x).p_X(x)} = -\sum_x p_X(x) \log_2 p_X(x) = H(X). \qquad \square$$

Ces relations sont des relations très intuitives lorsque l'on considère la représentation suivante.

2.6 Relation entre entropies



$H(X,Y) = H(X) + H(Y|X) = H(Y) + H(X|Y)$.

$$I(X;Y) = H(X) - H(X|Y) = H(Y) - H(Y|X) = H(X) + H(Y) - H(X,Y).$$

Exercice 30: Probabilité jointe

Soit deux dès tirés au hasard. Soit X la somme des deux dès et Y la valeur de la plus grande des deux.

1. Donner la loi p_X de X.
2. Donner la loi p_Y de Y.
3. Donner la loi $p_{X,Y}$ jointe de (X, Y). On vérifiera que $p_X(x) = \sum_y p_{XY}(x, y)$ et que $p_Y(y) = \sum_x p_{XY}(x, y)$.
4. Donner la probabilité que le maximum soit 2 sachant que la somme est 3.
5. Donner la probabilité que la somme soit 3 sachant que le maximum est 2.
6. Donner la probabilité que la somme soit 5 sachant que le maximum est 4.
7. Donner la probabilité que le maximum soit 3 sachant que la somme est 5.
8. Calculer l'entropie $H(X)$ et $H(Y)$.
9. Calculer l'entropie conjointe $H(X, Y)$.
10. Comparer $H(X, Y)$ avec $H(X) + H(Y)$.
11. Calculer les entropies conditionnelles $H(X|Y)$ et $H(Y|X)$. On les comparera avec $H(X)$ et $H(Y)$.
12. Calculer l'information mutuelle $I(X; Y)$.

2.7 Généralisation

Lors de la présentation que nous avons effectuées sur l'utilisation des lois statistiques afin de simuler un texte en langue anglaise.

Les exemples ont été générés en utilisant :

- pour la loi d'ordre 0 : $\Pr[X_i]$.
- pour la loi d'ordre 1 : $\Pr[X_i|X_{i-1}]$.
- pour la loi d'ordre 2 : $\Pr[X_i|X_{i-1}X_{i-2}]$.
- pour la loi d'ordre 3 : $\Pr[X_i|X_{i-1}X_{i-2}X_{i-3}]$.

La simulation de ces modèles s'effectue en utilisant des chaines de Markov (*i.e.* sorte de machine à état, où chaque état représente un symbole, et ou le choix des transitions s'effectue par choix aléatoire).

Théorème 16 (chaîne d'entropie généralisée)**.** *Soit $X_1, \ldots, X_n$ une suite de n variables aléatoires tirées suivant la loi jointe $p(x_1, \ldots, x_n)$ alors :*

$$H(X_1, \ldots, X_n) = \sum_{i=1}^{n} H(X_i|X_{i-1}, \ldots, X_1)$$

Démonstration:

On applique itérativement le corollaire sur les chaînes d'entropie. $H(X_1, X_2) = H(X_1) + H(X_2|X_1)$

$H(X_1, X_2, X_3) = H(X_1) + H(X_2, X_3|X_1) = H(X_1) + H(X_2|X_1) + H(X_3|X_2, X_1)$

$\vdots$

$$\begin{aligned} H(X_1, \ldots, X_n) &= H(X_1) + H(X_2|X_1) + \ldots + H(X_n|X_{n-1}, \ldots, X_1) \\ &= \sum_{i=1}^{n} H(X_i|X_{i-1}, \ldots, X_1) \end{aligned}$$

□

Corollaire 17 (Borne sur l'entropie généralisée)**.** *Soit $X_1, \ldots, X_n$ une suite de n variables aléatoires, alors :*

$$H(X_1, \ldots, X_n) \leq \sum_{i=1}^{n} H(X_i)$$

l'égalité n'ayant lieu que lorsque les variables sont indépendantes.

Démonstration:

Conséquence directement le théorème de réduction de l'entropie par conditionnement ($H(X|Y) \leq H(X)$) et de la chaîne d'entropie généralisée ($H(X_1, \ldots, X_n) = \sum_{i=1}^{n} H(X_i|X_{i-1}, \ldots, X_1)$).

Interprétation

Pour une chaîne constituée de variables aléatoires, la connaissance de la loi jointe permet toujours de réduire l'entropie, sauf si celles-ci sont indépendantes.

Nous allons maintenant voir comment utiliser ces résultats afin d'améliorer la compression.

3 Codage prédictif

Comme nous l'avons vu, la connaissance des lois jointes permet d'espérer d'obtenir une meilleure compression (plus un symbole a une probabilité importante, moins

on a besoin de bits pour stocker ce symbole).

Dans notre cas, l'ensemble des variables aléatoires que nous allons considérer tirent toutes leurs valeurs dans l'alphabet $\mathcal{A}$ qui contient l'ensemble des symboles.

La chaîne émise par la source est considérée comme étant une suite de variables aléatoires $(X_1, X_2, \ldots X_n)$.

La loi que nous voulons essayer de construire peut se formuler ainsi :

- je connais les p symboles précédents émis par la source,
- quelle est la loi du prochain symbole ?

Cette loi peut s'exprimer sous forme de la loi de probabilité conditionnelle :

$$\Pr\left[X_i | X_{i-1} X_{i-2} \ldots X_{i-p}\right]$$

C'est un processus aléatoire avec une mémoire d'ordre p.

Mais comment calculer une telle loi ?

L'ensemble des variables aléatoires discrète X_i prennent leurs valeurs dans $\mathcal{A}$, alors la loi jointe $p_{X_{i-p},\ldots,X_i}(x_{i-p}, \ldots, x_i)$ est une application de $\mathcal{A}^{p+1}$ dans $[0; 1]$.

Donc, $\mathcal{A}^{p+1}$ est un ensemble produit qui contient $(\#\mathcal{A})^{p+1}$ valeurs différentes.

La table suivante donne l'idée du nombre d'images de cette application en fonction de la mémoire choisie :

p	0	1	2	3	4
64^{p+1}	64	4096	262.144	16.777.216	1.973.741.824
256^{p+1}	256	65536	16.777.216	4.294.967.296	1.099.511.627.776

Autrement dit, sauf si l'alphabet est de petite taille, calculer cette fonction au-delà d'un ordre 2 n'est pas raisonnable du tout, sans compter que :

- la taille de l'échantillon nécessaire afin que l'estimation de $p_{X_{i-p},\ldots,X_i}$ ait un sens a toutes les chances d'être aussi déraisonnable,
- *a priori*, cette loi change à chaque fois que l'on compresse une chaîne différente (possiblement, cette fonction est à calculer pour chaque chaîne à compresser).
- si la loi $p_{X_{i-p},\ldots,X_i}$ n'est pas connue du décodeur, elle doit être reconstruite par ce dernier.

Pourtant, il serait souhaitable d'utiliser des contextes $X_{i-1} X_{i-2} \ldots X_{i-p}$ de grande taille afin de déterminer le symbole X_i à coder.

Mais comment utiliser de tels contextes sans estimer et stocker un très grand nombre de probabilités conditionnelles ?

3.1 PPM

3.1.1 Principe

Proposé par Cleary et Witten [CW84] (voir aussi [SMB10], p292), le codage PPM (partial prediction matching) est une méthode qui :

- construit les probabilités conditionnelles au fur et à mesure qu'elle rencontre les symboles,
- se place dans le contexte le plus adapté au contexte courant,
- utilise cette probabilité conditionnelle du contexte afin de connaître la probabilité d'apparition du symbole courant et en déduire son code.
- utilise un codage arithmétique adaptatif avec les probabilités du symbole dans le contexte.

Les probabilités conditionnelles n'étant construite que pour les chaînes rencontrées, on limite donc la place nécessaire "qu'à" celle nécessaire pour stocker les contextes passés.

Exemple : contextes jusqu'à l'ordre 4 pour la chaine "xyzzxyxyzzx"

Lecture d'un symbole "z" : (chaîne "xyzzxyxyzzxz").
Contexte : $\Pr[X_i = z] = \frac{4}{11}$
La figure (a) présente la table des contextes associés.

Mise à jour des contextes avec le symbole "z" précédent.
Lecture d'un symbole "z" (chaîne "xyzzxyxyzzxzz").
Contexte : $\Pr[X_i = z] = \frac{5}{12}$, $\Pr[X_i = z | X_{i-1} = z] = \frac{2}{4} = \frac{1}{2}$
La figure (b) présente la table des contextes associés.

Mise à jour des contextes avec le symbole "z" précédent.
Lecture d'un symbole "x" (chaîne "xyzzxyxyzzxzzx").
Contextes :
$\Pr[X_i = x] = \frac{4}{13}$,
$\Pr[X_i = x | X_{i-1} = z] = \frac{1}{4}$,
$\Pr[X_i = x | X_{i-1} = z, X_{i-2} = z] = \frac{2}{2} = 1$.
La figure (c) présente la table des contextes associés.

Comme attendu, plus nous sommes dans un contexte restreint, plus la probabilité du symbole est élevé, et moins nous avons besoin de bits pour le stocker.

...

Problème

Si le codeur peut choisir le contexte le plus approprié pour un caractère à coder (parce qu'il le connaît), comment fait le décodeur pour déterminer le

Ordre 0	Ordre 1	Ordre 2	Ordre 3	Ordre 4
x 4	x→y 3	xy→z 2	xyz→z 2	xyzz→x 2
y 3	y→z 2	→x 1	yzz→x 2	yzzx→y 1
z 4	→x 1	yz→z 2	zzx→y 1	zzxy→x 1
	z→z 2	zz→x 2	zxy→x 1	zxyx→y 1
	→x 2	zx→y 1	xyx→y 1	xyxy→z 1
		yx→y 1	yxy→z 1	yxyz→z 1

(a) contexte après la lecture de la chaîne "xyzzxyxyzzx".

Ordre 0	Ordre 1	Ordre 2	Ordre 3	Ordre 4
x 4	x→y 3	xy→z 2	xyz→z 2	xyzz→x 2
y 3	→z 1	→x 1	yzz→x 2	yzzx→y 1
z 5	y→z 2	yz→z 2	zzx→y 1	→z 1
	→x 1	zz→x 2	→z 1	zzxy→x 1
	z→z 2	zx→y 1	zxy→x 1	zxyx→y 1
	→x 2	→z 1	xyx→y 1	xyxy→z 1
		yx→y 1	yxy→z 1	yxyz→z 1

(b) ajout d'un z.

Ordre 0	Ordre 1	Ordre 2	Ordre 3	Ordre 4
x 4	x→y 3	xy→z 2	xyz→z 2	xyzz→x 2
y 3	→z 1	→x 1	yzz→x 2	yzzx→y 1
z 6	y→z 2	xz→z 1	zxz→z 1	→z 1
	→x 1	yz→z 2	zzx→y 1	zzxy→x 1
	z→z 3	zz→x 2	→z 1	zzxz→z 1
	→x 2	zx→y 1	zxy→x 1	zxyx→y 1
		→z 1	xyx→y 1	xyxy→z 1
		yx→y 1	yxy→z 1	yxyz→z 1

(c) ajout d'un x.

FIGURE IV.2 – Mise-à-jour des contextes jusqu'à l'ordre 4.

changement de contexte ?
Il ne le peut pas sans information supplémentaire : il faut ajouter un symbole $\triangle$ pour signifier un changement de contexte.

- lors du codage, lorsque l'on passe à un contexte d'ordre n à $n-1$, on envoie un symbole $\triangle$ (codé arithmiquement).
- lors du décodage, lorsque l'on décode un symbole $\triangle$, on passe à un contexte plus court.

Remarquons que suivant le contexte, la fréquence d'utilisation du symbole $\triangle$ dépend de la fréquence avec laquelle on revient à un contexte plus court à partir du contexte courant.
Donc, il faut que le codage de $\triangle$ dépende aussi de sa probabilité d'utilisation.
Aussi, le symbole $\triangle$ sera traité comme un symbole à part entière, et chaque changement de contexte entraînera une mise à jour des contextes.

3.1.2 Codage

Exemple

Présentons l'algorithme de codage sur un exemple. Voici la table des contextes avec caractères de changement de contexte après lecture de la chaîne "assanissimassa".

Ordre 0	Ordre 1		Ordre 2		
a 4	a→s 2	n→i 1	as→s 2	an→i 1	si →m 1
s 6	→n 1	→△ 1	→△ 1	→△ 1	→△ 1
n 1	→△ 2	i→s 1	ss→a 2	ni→s 1	im→a 1
i 2	s→s 3	→m 1	→i 1	→△ 1	→△ 1
m 1	→a 2	→△ 2	→△ 2	is→s 1	ma→s 1
△ 5	→i 1	m→a 1	sa→n 1	→△ 1	→△ 1
	→△ 3	→△ 1	→△ 1		

Après la lecture de cette chaîne, le contexte est d'ordre 2, car la transition $s \rightarrow sa$ a déjà été rencontrée.
Voyons maintenant ce qui peut se passer lors de l'insertion d'un caractère supplémentaire. Il y 4 cas typique :

1$^{\text{er}}$ **cas** : si le symbole suivant est n
$sa \rightarrow n$ connu : $\Pr[n|sa] = \frac{1}{2}$
codage : n (1 bit).
MAJ : n = 2, a→n = 2, sa→n = 2

2$^{\text{ème}}$ **cas** : si le symbole suivant est s
$sa \rightarrow s$ inconnu
$sa \rightarrow \triangle$: $\Pr[\triangle|sa] = \frac{1}{2}$
codage : $\triangle$ (1 bit)
MAJ : sa→s = 1, sa→△ = 2

$a \rightarrow s$ connu : $\Pr[s|a] = \frac{2}{5}$
codage : s (+ 1.32 bit)
MAJ : a→s = 3, s = 7

3ème **cas** : si le symbole suivant est m
$sa \rightarrow m$ inconnu
$sa \rightarrow \triangle$: $\Pr[\triangle|sa] = \frac{1}{2}$
codage : $\triangle$ (1 bit)
MAJ : sa→m = 1, sa→$\triangle$ = 2
$a \rightarrow m$ inconnu
$a \rightarrow \triangle$: $\Pr[\triangle|a] = \frac{2}{5}$
codage : $\triangle$ (+ 1.32 bit)
MAJ : a→m = 1, a→$\triangle$ = 3
m connu : $\Pr[m] = \frac{1}{19}$
codage : m (+ 4.25 bit)
MAJ : m = 2

4ème **cas** : si le symbole suivant est d
$sa \rightarrow d$ inconnu
$sa \rightarrow \triangle$: $\Pr[\triangle|sa] = \frac{1}{2}$
codage : $\triangle$ (1 bit)
MAJ : sa→d = 1, sa→$\triangle$ = 2
$a \rightarrow d$ inconnu
$a \rightarrow \triangle$: $\Pr[\triangle|a] = \frac{2}{5}$
codage : $\triangle$ (+ 1.32 bit)
MAJ : a→d = 1, a→$\triangle$ = 3
d inconnu
$\triangle$: $\Pr[\triangle] = \frac{5}{19}$
codage : $\triangle$ (+ 1.93 bit)
MAJ : $\triangle$ = 6, d = 1
$\Pr[d] = 1/27$ (dans l'alphabet)
codage : d (+ 4.8 bit)

Alphabet = a-z + $\triangle$ (taille=27)

Algorithme de codage

L'algorithme de codage est présenté à la figure IV.3.

3.1.3 Décodage

Pour le décodage, il suffit d'effectuer les opérations en miroir :

```
while ( !EOF) do
    s = ReadSymbol()
    // recherche du contexte contenant le symbole lu
    while (s ∉ context) et (context ≠ ∅) do
        // codage de △ dans son contexte
        AdaptativeAlgebraicCoding(context,△)
        // mise à jour du contexte (insère si besoin et incrémente le
          compteur)
        UpdateContext(context,△)
        UpdateContext(context,s)
        // remonte au contexte parent
        ContextUp(context)
    if context = ∅ then
        // cas où le symbole s n'est pas connu
        AdaptativeAlgebricCoding(Alphabet,s)
    else
        // codage de s dans le contexte
        AdaptativeAlgebraicCoding(context,s)
        // mise à jour de tous les contextes
        UpdateAllAscendingContexts(context,s)
    // revient dans le contexte le plus local incluant s
    ContextDown(in,s)
```

FIGURE IV.3 – Algoritme de codage prédictif.

- on effectue le décodage algébrique pour déterminer le symbole codé en utilisant les probabilités du contexte local,
- s'il s'agit d'un changement de contexte, c'est que le symbole à insérer dans le contexte courant (noté `initial`) sera lu dans le premier contexte supérieur dans lequel il est défini.
  - ◇ on remonte au contexte supérieur (noté `last`),
  - ◇ on effectue le décodage algébrique du symbole s en utilisant les probabilités du contexte local,
  - ◇ on recommence tant que le caractère lu est $\triangle$
- mettre à jour tous les contextes ascendants {`initial`, ..., `last`} avec s et $\triangle$,
- mettre à jour tous les contextes descendants avec s,
- descendre dans le contexte s.

```
while ( !EOF) do
    s = AdaptativeAlgebraicDecoding::GetNextSymbol(context,input)
    initial = context
    // recherche du contexte dans lequel le caractère est codé
    while s = △ do
        last = context
        ContextUp(context)
        s =
          AdaptativeAlgebraicDecoding::GetNextSymbol(context,input)
    // mise à jour des contextes descendant
    UpdateAllDescendingContext(last,initial,△)
    UpdateAllDescendingContext(last,initial,s)
    // mise à jour des contextes ascendant
    UpdateAllAscendingContext(context,s)
    // passe dans le contexte incluant s
    ContextDown(context,s)
```

FIGURE IV.4 – Algoritme de décodage prédictif.

Ce qui se reformule de la manière suivante :

L'algorithme de décodage est présenté à la figure IV.4.

EXERCICE 31: Codage PPM

1. Calculer les contextes jusqu'à l'ordre 4 pour la chaine "aababbaabbaabbba".
2. Appliquer le codage PPM à la même chaîne avec un contexte d'ordre 3.
3. Donner l'ensemble des caractères envoyés avec leurs nombres de bits (i.e. pour chaque codage, déterminer le contexte de codage).
4. Utiliser ces caractères envoyés pour effectuer le décodage PPM d'ordre 3.
5. Quel aurait été le nombre de bits envoyé par un codage algébrique de la même séquence ? Comment l'expliquer ?

3.1.4 Remarque

Reprenons l'exemple précédent : si le contexte sa et le symbole lu est s

- le contexte sa→s n'existe pas, on passe donc au contexte a

- mais, pour tout x qui existe dans le contexte sa (i.e. ceux pour lesquels il existe une transition sa→x), nous sommes certains que les transitions a→x ne seront pas valides non plus.
- en conclusion, on peut exclure tous ces symboles du contexte a puisqu'ils sont nécessairement impossible.

Donc, dans le contexte a, on sait que la transition a→n est impossible (puisque la transition sa→n) n'est pas possible. La probabilité de la transition peut être calculée en excluant la transition a→n des calculs.

$\Pr[a|s] = \frac{2_s}{2_s+2_\triangle} = \frac{1}{2}$ au lieu de $\Pr[a|s] = \frac{2_s}{2_s+2_\triangle+1_n} = \frac{2}{5}$.

Rappel : si on sait bien que le caractère suivant est s, le décodeur lui ne le sait pas. On ne peut donc exclure que les transitions utilisant les symboles dont le décodeur peut, en raison du changement de contexte, en déduire leur impossibilité. On parle alors d'exclusion.

3.2 DMC

Le codage DMC (pour Dynamic Markov Coding) est une méthode dûe à Cormack et Horspool [CH86].

Elle produit une compression excellente, comparable à celle obtenu par PPM, mais elle est plus rapide.

Son principe est basé sur l'utilisation conjointe :

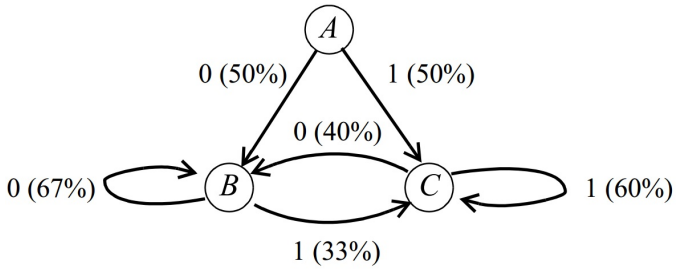
- d'une modèle de Markov mise à jour dynamiquement en fonction des symboles lus
- d'un codage algébrique pour coder le symbole en utilisant les probabilités de l'état courant.

Nous n'exposerons que le principe général de cette méthode. Pour plus de détail, voir [SMB10] ou [CH86].

Qu'est ce qu'un modèle de Markov ?

Un modèle de Markov est une machine à état dans laquelle une probabilité est affectée à chaque transition.

Exemple :



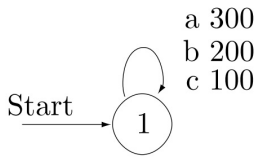
Fonctionnement :

- Dans un état donné, les probabilités de transition sont indiquées sur les arête.
- Une fois la transition choisie, on se retrouve dans un nouvel état avec de nouvelles probabilités de transition.

Cette machine est utilisée pour stocker les statistiques locales

Exemple

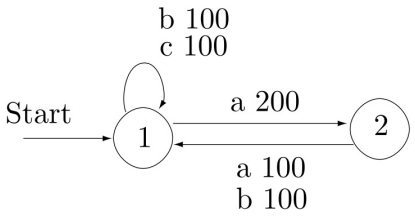
Considérons un alphabet à 3 symboles $\mathcal{A} = \{a, b, c\}$.
Supposons la chaine à coder soit le motif *aaabbc* répété 100 fois.
Si on considère un modèle à un état, et en mettant à jour les comptages des symboles au fur et à mesure, on aboutit aux probabilités suivantes :



$\Pr[a] = \frac{300}{600}$
$\Pr[b] = \frac{200}{600}$
$\Pr[c] = \frac{100}{600}$

Entropie : $H(X) = 1.46$ bit/symbole

Avec un modèle à deux états :



Pour l'état 1 :
$\Pr[a] = \frac{200}{600}$, $\Pr[b] = \frac{100}{600}$, $\Pr[c] = \frac{100}{600}$
$H_1(X) = 1.3899$ bits/symbol
Pour l'état 2 :
$\Pr[a] = \frac{100}{600}$, $\Pr[b] = \frac{100}{600}$
$H_2(X) = 0.8616$ bits/symbol

Entropie : $H(X) = \frac{400}{600}H_1(x) + \frac{200}{600}H_2(x) = 1.21$ bit/symbol.

En conséquence, un codage algébrique du deuxième modèle permet d'obtenir un meilleur codage qu'un codage entropique classique.

L'idée du DMC est donc de partir d'une machine à un état, puis pour chaque symbole lu :

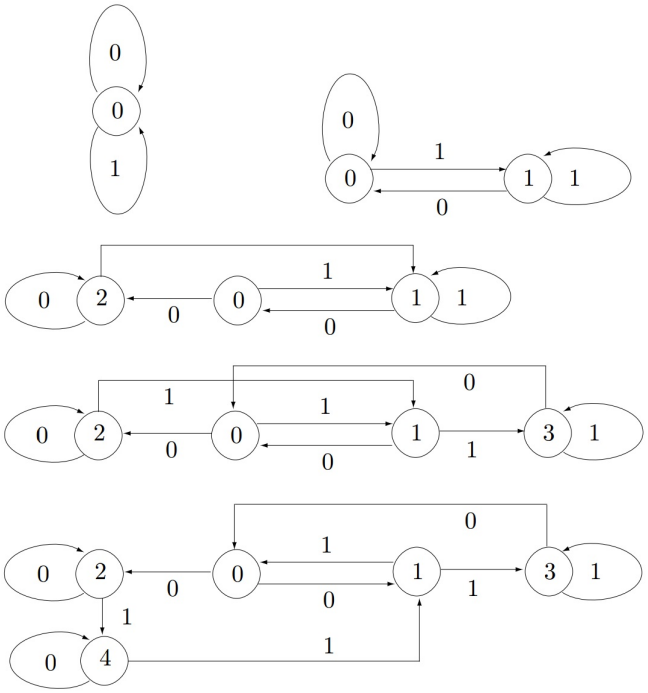
1. de mettre à jour les probabilités des transitions (par comptage et renormalisation),
2. d'ajouter des états si nécessaire,

   si les transitions $A \to C$ et $B \to C$ ont des comptages significativement différents, on crée une copie de l'état C, et on copie les états sortant de C, *i.e.* $A \to C \to (D, E)$ et $A \to C' \to (D, E)$.
3. effectuer un codage algébrique du choix de la transition en utilisant les probabilités de la machine à état pour l'état courant.

Le décodeur fonctionne en miroir (les mises à jour de la machine à état n'ont lieu qu'après codage/décodage sous les mêmes conditions).

L'idée est donc similaire au PPM. Ici, les probabilités conditionnelles sont construites sur la base d'une machine à état, à la différence que près que les mises à jour ne sont effectuées que lorsqu'une différence significative est mesurée sur les transitions.
Voir transparent ci-après pour un exemple d'évolution du modèle de Markov en cours de compression.

Exemple d'évolution de la machine à état



4 Conclusion

Les techniques de codage prédictif font partie des méthodes de compression les plus modernes et efficaces, car elles :

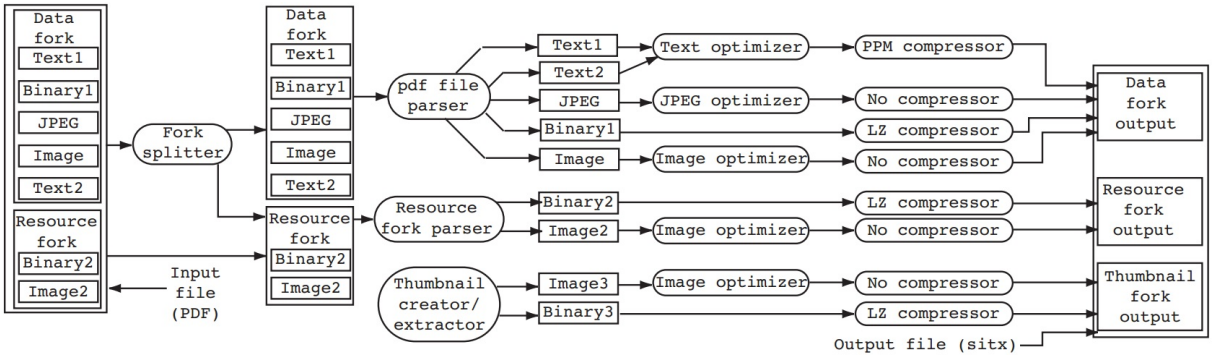
- tiennent compte des dépendances entre symboles ou suites de symboles,
- mettent à jour automatiquement les probabilités de transition en fonction de celles déjà rencontrées.

Les techniques de transformation de code peuvent parfois aider les encodeurs à trouver les redondances.

D'une façon générale, les compresseurs commerciaux utilisent des combinaisons de techniques vues depuis le début du cours :

- codage en plusieurs passes utilisant par exemple un codage par dictionnaire (pour supprimer les redondances) suivi d'un codage de Huffman (pour optimiser un codage avec peu de redondance spatiale).
- choix de l'encodeur en fonction des données à compresser.

Exemple (compression d'un fichier pdf avec StuffIt)



Cette leçon termine la partie du cours sur la compression.

Deuxième partie : Cryptographie