Théorie de l'Information (2)

Philippe Duchon

ENSEIRB

2007-08

Exercice : entropie et décision

Une naissance a eu lieu un jour de la semaine, sur lequel on n'a aucune information. On a le choix de recevoir la réponse à une question :

- Q_1 : "la naissance a-t-elle eu lieu (a) lundi, mardi ou mercredi, ou (b) jeudi, vendredi, samedi ou dimanche ?"
- Q_2 : "la naissance a-t-elle eu lieu (a) samedi, ou (b) dimanche, ou (c) un autre jour ?"
- Q_3 : "la naissance a-t-elle eu lieu (a) lundi ou mardi, ou (b) mercredi ou jeudi, ou (c) vendredi, samedi ou dimanche ?"

Décrire la situation par trois systèmes composés, et proposer un critère de choix de la question à poser.

Exercice : prévisions météorologiques (janvier 2005)

On teste un système de prévisions météorologiques pour lequel on a obtenu, sur un an, les fréquences de résultats suivantes :

	Temps : Pluie	Temps : Soleil
Prévu : Pluie	$1/12$	$1/6$
Prévu : Soleil	$1/12$	$2/3$

- Quelle est la probabilité que le système donne une prévision incorrecte ?
- Calculer l'information mutuelle entre le temps prévu et le temps effectif.
- Comparer à un système de "prévisions" qui prédit systématiquement du soleil. Commenter.

Exercice : test de dépistage

On considère une situation dans laquelle un test de dépistage pour une maladie est censé discriminer entre les situations $X \in \{V, nV\}$, en donnant un résultat $Y \in \{T^+, T^-\}$. Les probabilités respectives des différents cas sont les suivantes :

	T^+	T^-
V	0.07	0.01
nV	0.03	0.89

On définit l'*efficacité* du test comme étant $r = \frac{I(X,Y)}{H(X)}$.

- Quelles valeurs peut, *a priori*, prendre r ?
- Calculer $H(X)$, $I(X, Y)$ et r.
- À quoi correspondraient des valeurs $r = 0$? $r = 1$?

Source simple

- $k \geq 2$ entier, $(p_1, \ldots, p_k)$ distribution de probabilités
- On considère une "source" S qui émet des symboles $X_1, X_2, \ldots$ aléatoires

Source simple

- $k \geq 2$ entier, $(p_1, \ldots, p_k)$ distribution de probabilités
- On considère une "source" S qui émet des symboles $X_1, X_2, \ldots$ aléatoires
- **Source simple :** les X_i sont des variables aléatoires **indépendantes**, toutes **de même loi** $\mathbb{P}(X_i = s_j) = p_j$

Source simple

- $k \geq 2$ entier, $(p_1, \ldots, p_k)$ distribution de probabilités
- On considère une "source" S qui émet des symboles $X_1, X_2, \ldots$ aléatoires
- **Source simple :** les X_i sont des variables aléatoires **indépendantes**, toutes **de même loi** $\mathbb{P}(X_i = s_j) = p_j$
- **Entropie de la source :** $H(S) = H(X_1)$

Source simple

- $k \geq 2$ entier, $(p_1, \ldots, p_k)$ distribution de probabilités
- On considère une "source" S qui émet des symboles $X_1, X_2, \ldots$ aléatoires
- **Source simple :** les X_i sont des variables aléatoires **indépendantes**, toutes **de même loi** $\mathbb{P}(X_i = s_j) = p_j$
- **Entropie de la source :** $H(S) = H(X_1)$
- r-suite : la suite de r symboles $X_1, X_2, \ldots, X_r$

Source simple

- $k \geq 2$ entier, $(p_1, \ldots, p_k)$ distribution de probabilités
- On considère une "source" S qui émet des symboles $X_1, X_2, \ldots$ aléatoires
- **Source simple :** les X_i sont des variables aléatoires **indépendantes**, toutes **de même loi** $\mathbb{P}(X_i = s_j) = p_j$
- **Entropie de la source :** $H(S) = H(X_1)$
- r-suite : la suite de r symboles $X_1, X_2, \ldots, X_r$
- On a facilement (récurrence sur r)

$$H(X_1, X_2, \ldots, X_r) = rH(S).$$

Occurrences d'un symbole, suites typiques

- On note $N_i(r)$, le **nombre d'occurrences** du symbole s_i dans la r-suite $X_1, \ldots, X_r$.

Occurrences d'un symbole, suites typiques

- On note $N_i(r)$, le **nombre d'occurrences** du symbole s_i dans la r-suite $X_1, \ldots, X_r$.
- La **loi** de $N_i(r)$ est la loi **binomiale** $\mathcal{B}_{r,p_i}$; espérance $r.p_i$, variance $r.p_i(1-p_i)$

Occurrences d'un symbole, suites typiques

- On note $N_i(r)$, le **nombre d'occurrences** du symbole s_i dans la r-suite $X_1, \ldots, X_r$.
- La **loi** de $N_i(r)$ est la loi **binomiale** $\mathcal{B}_{r,p_i}$; espérance $r.p_i$, variance $r.p_i(1-p_i)$
- Le **théorème de la limite centrale** affirme

$$\frac{N_i(r) - rp_i}{\sqrt{rp_i(1-p_i)}} \xrightarrow{(d)} \mathcal{N}(0,1)$$

Occurrences d'un symbole, suites typiques

- On note $N_i(r)$, le **nombre d'occurrences** du symbole s_i dans la r-suite $X_1, \ldots, X_r$.
- La **loi** de $N_i(r)$ est la loi **binomiale** $\mathcal{B}_{r,p_i}$; espérance $r.p_i$, variance $r.p_i(1-p_i)$
- Le **théorème de la limite centrale** affirme

$$\frac{N_i(r) - rp_i}{\sqrt{rp_i(1-p_i)}} \xrightarrow{(d)} \mathcal{N}(0,1)$$

- L'**inégalité de Tchebycheff** donne

$$\mathbb{P}\left(\frac{|N_i(r) - rp_i|}{\sqrt{rp_i(1-p_i)}} \geq \lambda\right) \leq \frac{1}{\lambda^2}$$

Occurrences d'un symbole, suites typiques

- On note $N_i(r)$, le **nombre d'occurrences** du symbole s_i dans la r-suite $X_1, \ldots, X_r$.
- La **loi** de $N_i(r)$ est la loi **binomiale** $\mathcal{B}_{r,p_i}$; espérance $r.p_i$, variance $r.p_i(1-p_i)$
- Le **théorème de la limite centrale** affirme

$$\frac{N_i(r) - rp_i}{\sqrt{rp_i(1-p_i)}} \xrightarrow{(d)} \mathcal{N}(0,1)$$

- L'**inégalité de Tchebycheff** donne

$$\mathbb{P}\left(\frac{|N_i(r) - rp_i|}{\sqrt{rp_i(1-p_i)}} \geq \lambda\right) \leq \frac{1}{\lambda^2}$$

- Une suite est λ-typique pour le symbole s_i, si son nombre d'occurrences du symbole satisfait l'inégalité

Occurrences d'un symbole, suites typiques

- On note $N_i(r)$, le **nombre d'occurrences** du symbole s_i dans la r-suite $X_1, \ldots, X_r$.
- La **loi** de $N_i(r)$ est la loi **binomiale** $\mathcal{B}_{r,p_i}$; espérance $r.p_i$, variance $r.p_i(1-p_i)$
- Le **théorème de la limite centrale** affirme

$$\frac{N_i(r) - rp_i}{\sqrt{rp_i(1-p_i)}} \stackrel{(d)}{\rightarrow} \mathcal{N}(0,1)$$

- L'**inégalité de Tchebycheff** donne

$$\mathbb{P}\left(\frac{|N_i(r) - rp_i|}{\sqrt{rp_i(1-p_i)}} \geq \lambda\right) \leq \frac{1}{\lambda^2}$$

- Une suite est λ-typique pour le symbole s_i, si son nombre d'occurrences du symbole satisfait l'inégalité
- Une suite est λ-typique, si elle est λ-typique pour chaque symbole

Théorème de la source binomiale

- La probabilité que la source émette une r-suite qui soit λ-typique, est d'au moins $1 - k/\lambda^2$;

Théorème de la source binomiale

- La probabilité que la source émette une r-suite qui soit λ-typique, est d'au moins $1 - k/\lambda^2$;
- Dans la pratique, si r est grand, cette probabilité est **beaucoup plus proche que cela de** 1 lorsque λ dépasse quelques unités (TCL).

Théorème de la source binomiale

- La probabilité que la source émette une r-suite qui soit λ-typique, est d'au moins $1 - k/\lambda^2$;
- Dans la pratique, si r est grand, cette probabilité est **beaucoup plus proche que cela de** 1 lorsque λ dépasse quelques unités (TCL).
- **Théorème (admis) :** le nombre $T_{r,\lambda}$ de r-suites λ-typiques est "approximativement"

$$T_{r,\lambda} \simeq 2^{rH(S)}$$

(au sens logarithmique : $\log(T_{r,\lambda}) - rH(S) = o(r)$)

Théorème de la source binomiale

- La probabilité que la source émette une r-suite qui soit λ-typique, est d'au moins $1 - k/\lambda^2$;
- Dans la pratique, si r est grand, cette probabilité est **beaucoup plus proche que cela de** 1 lorsque λ dépasse quelques unités (TCL).
- **Théorème (admis) :** le nombre $T_{r,\lambda}$ de r-suites λ-typiques est "approximativement"

$$T_{r,\lambda} \simeq 2^{rH(S)}$$

(au sens logarithmique : $\log(T_{r,\lambda}) - rH(S) = o(r)$)

- Dans la pratique, le nombre de suites λ-typiques ne dépend quasiment pas de λ (si r est assez grand)

Théorème de la source binomiale

- La probabilité que la source émette une r-suite qui soit λ-typique, est d'au moins $1 - k/\lambda^2$;
- Dans la pratique, si r est grand, cette probabilité est **beaucoup plus proche que cela de** 1 lorsque λ dépasse quelques unités (TCL).
- **Théorème (admis) :** le nombre $T_{r,\lambda}$ de r-suites λ-typiques est "approximativement"

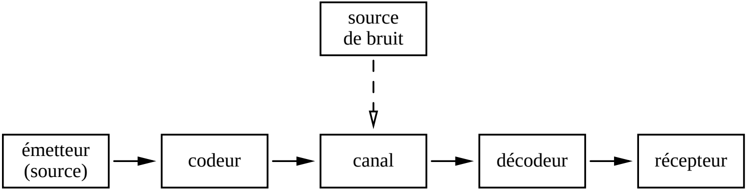
$$T_{r,\lambda} \simeq 2^{rH(S)}$$

(au sens logarithmique : $\log(T_{r,\lambda}) - rH(S) = o(r)$)

- Dans la pratique, le nombre de suites λ-typiques ne dépend quasiment pas de λ (si r est assez grand)
- La preuve se fait en torturant la formule de Stirling

Problématique du codage

Modélisation classique d'une transmission d'information :



Notion de code

- Une **source** S émet des **messages**, qui sont des suites de symboles pris dans un **alphabet de source** A

Notion de code

- Une **source** S émet des **messages**, qui sont des suites de symboles pris dans un **alphabet de source** A
- Un **code** est une **fonction** qui à chaque message $m \in A^*$, fait correspondre un mot $C(m) \in B^*$ où B est un **alphabet de codage**

Notion de code

- Une **source** S émet des **messages**, qui sont des suites de symboles pris dans un **alphabet de source** A
- Un **code** est une **fonction** qui à chaque message $m \in A^*$, fait correspondre un mot $C(m) \in B^*$ où B est un **alphabet de codage**
- Typiquement, en vue d'une transmission sur un canal donné, l'alphabet de codage est adapté au canal

Notion de code

- Une **source** S émet des **messages**, qui sont des suites de symboles pris dans un **alphabet de source** A
- Un **code** est une **fonction** qui à chaque message $m \in A^*$, fait correspondre un mot $C(m) \in B^*$ où B est un **alphabet de codage**
- Typiquement, en vue d'une transmission sur un canal donné, l'alphabet de codage est adapté au canal
- Le plus souvent : **codage binaire**, l'alphabet de codage est $B = \{0, 1\}$

Codage de source

Cas particulier dans l'ensemble des codes possibles

- Pour chaque symbole $s \in A$, on définit son codage $C(s) \in B^*$ (un mot de l'alphabet de codage)

Codage de source

Cas particulier dans l'ensemble des codes possibles

- Pour chaque symbole $s \in A$, on définit son codage $C(s) \in B^*$ (un mot de l'alphabet de codage)
- Si un message m est composé d'une suite de lettres,

$$m = s_1 s_2 \cdots s_\ell,$$

on définit son codage $C(m)$ par concaténation :

$$C(m) = C(s_1)C(s_2)\cdots C(s_\ell)$$

Codage de source

Cas particulier dans l'ensemble des codes possibles

- Pour chaque symbole $s \in A$, on définit son codage $C(s) \in B^*$ (un mot de l'alphabet de codage)
- Si un message m est composé d'une suite de lettres,

$$m = s_1 s_2 \cdots s_\ell,$$

  on définit son codage $C(m)$ par concaténation :

$$C(m) = C(s_1)C(s_2) \cdots C(s_\ell)$$

- Si l'on était grossier, on dirait qu'on a défini C comme un **morphisme du monoïde libre A^* vers le monoïde libre B^***

Codage de source

Cas particulier dans l'ensemble des codes possibles

- Pour chaque symbole $s \in A$, on définit son codage $C(s) \in B^*$ (un mot de l'alphabet de codage)
- Si un message m est composé d'une suite de lettres,

$$m = s_1 s_2 \cdots s_\ell,$$

on définit son codage $C(m)$ par concaténation :

$$C(m) = C(s_1)C(s_2)\cdots C(s_\ell)$$

- Si l'on était grossier, on dirait qu'on a défini C comme un **morphisme du monoïde libre** A^* **vers le monoïde libre** B^*
- Mais on va essayer de rester poli

Notations : longueurs

- La **longueur d'un mot** m est le **nombre de lettres** (pas forcément distinctes) qui le composent

Notations : longueurs

- La **longueur d'un mot** m est le **nombre de lettres** (pas forcément distinctes) qui le composent
- On la note $\ell(m)$, parfois $|m|$

Notations : longueurs

- La **longueur d'un mot** m est le **nombre de lettres** (pas forcément distinctes) qui le composent
- On la note $\ell(m)$, parfois $|m|$
- On note $\ell_s(m)$, ou $|m|_s$, le nombre d'occurrences de la lettre s dans le mot m

Notations : longueurs

- La **longueur d'un mot** m est le **nombre de lettres** (pas forcément distinctes) qui le composent
- On la note $\ell(m)$, parfois $|m|$
- On note $\ell_s(m)$, ou $|m|_s$, le nombre d'occurrences de la lettre s dans le mot m
- Pour définir un codage de source pour l'alphabet $A = \{s_1, \ldots, s_k\}$, on utilise la notation $C = \{M_1, \ldots, M_k\}$, ce qui sous-entend $C(s_i) = M_i$

Notations : longueurs

- La **longueur d'un mot** m est le **nombre de lettres** (pas forcément distinctes) qui le composent
- On la note $\ell(m)$, parfois $|m|$
- On note $\ell_s(m)$, ou $|m|_s$, le nombre d'occurrences de la lettre s dans le mot m
- Pour définir un codage de source pour l'alphabet $A = \{s_1, \ldots, s_k\}$, on utilise la notation $C = \{M_1, \ldots, M_k\}$, ce qui sous-entend $C(s_i) = M_i$
- Notation : $\ell_i = \ell(M_i)$

Notations : longueurs

- La **longueur d'un mot** m est le **nombre de lettres** (pas forcément distinctes) qui le composent
- On la note $\ell(m)$, parfois $|m|$
- On note $\ell_s(m)$, ou $|m|_s$, le nombre d'occurrences de la lettre s dans le mot m
- Pour définir un codage de source pour l'alphabet $A = \{s_1, \ldots, s_k\}$, on utilise la notation $C = \{M_1, \ldots, M_k\}$, ce qui sous-entend $C(s_i) = M_i$
- Notation : $\ell_i = \ell(M_i)$
- On a trivialement, pour tout mot $m \in A^*$,

$$\ell(C(m)) = \sum_{i=1}^{k} \ell_{s_i}(m)\ell_i$$

Notations : longueurs

- La **longueur d'un mot** m est le **nombre de lettres** (pas forcément distinctes) qui le composent
- On la note $\ell(m)$, parfois $|m|$
- On note $\ell_s(m)$, ou $|m|_s$, le nombre d'occurrences de la lettre s dans le mot m
- Pour définir un codage de source pour l'alphabet $A = \{s_1, \ldots, s_k\}$, on utilise la notation $C = \{M_1, \ldots, M_k\}$, ce qui sous-entend $C(s_i) = M_i$
- Notation : $\ell_i = \ell(M_i)$
- On a trivialement, pour tout mot $m \in A^*$,

$$\ell(C(m)) = \sum_{i=1}^{k} \ell_{s_i}(m)\ell_i$$

- Notion de *longueur maximale d'un code* : $L = \max_i(\ell_i)$

Notations : longueurs

- La **longueur d'un mot** m est le **nombre de lettres** (pas forcément distinctes) qui le composent
- On la note $\ell(m)$, parfois $|m|$
- On note $\ell_s(m)$, ou $|m|_s$, le nombre d'occurrences de la lettre s dans le mot m
- Pour définir un codage de source pour l'alphabet $A = \{s_1, \ldots, s_k\}$, on utilise la notation $C = \{M_1, \ldots, M_k\}$, ce qui sous-entend $C(s_i) = M_i$
- Notation : $\ell_i = \ell(M_i)$
- On a trivialement, pour tout mot $m \in A^*$,

$$\ell(C(m)) = \sum_{i=1}^{k} \ell_{s_i}(m)\ell_i$$

- Notion de *longueur maximale d'un code* : $L = \max_i(\ell_i)$
- Trivialement, $\ell(C(m)) \leq L\ell(m)$

Deux théories du codage

On peut essayer d'aborder la théorie du codage de deux manières différentes :

- **Purement combinatoire :** on a des mots qu'on réécrit en d'autres mots, et on s'intéresse à des propriétés combinatoires du code. Par exemple, la possibilité de retrouver le mot à partir de son codage (notion de déchiffrabilité)

Deux théories du codage

On peut essayer d'aborder la théorie du codage de deux manières différentes :

- **Purement combinatoire :** on a des mots qu'on réécrit en d'autres mots, et on s'intéresse à des propriétés combinatoires du code. Par exemple, la possibilité de retrouver le mot à partir de son codage (notion de déchiffrabilité)
- **Probabiliste :** on suppose que la source est un objet probabiliste qui émet des suites de symboles aléatoires, ce qui permet de s'intéresser à des propriétés ou à des grandeurs *en moyenne* ; c'est là qu'intervient fructueusement la notion d'entropie.

Notion de déchiffrabilité

Définition

*Un code C est dit **déchiffrable** (ou **uniquement déchiffrable**, ou **non ambigu**) s'il est injectif :*

$$\forall m \in A^*, \forall m' \in A^*, C(m) = C(m') \implies m = m'$$

*Un code qui n'est pas uniquement déchiffrable est dit **ambigu**.*

- Si un code est uniquement déchiffrable, il existe donc une **fonction de décodage** (fonction réciproque) $D : C(A^*) \to A^*$ telle que $D(C(m)) = m$ pour tout m ;
- En toute généralité, **il n'est pas toujours facile** de décider si un code est uniquement déchiffrable, ni, s'il l'est, de calculer la fonction de décodage.

C'est-y déchiffrable ?

- **Prouver** qu'un code est ambigu est conceptuellement simple : "il suffit" d'exhiber deux mots distincts qui ont le même codage

C'est-y déchiffrable ?

- **Prouver** qu'un code est ambigu est conceptuellement simple : “il suffit” d'exhiber deux mots distincts qui ont le même codage
- Pour **prouver** qu'un code est uniquement déchiffrable, c'est *a priori* plus compliqué : moralement, il faut exhiber la fonction de décodage et prouver qu'elle est correcte

C'est-y déchiffrable ?

- **Prouver** qu'un code est ambigu est conceptuellement simple : "il suffit" d'exhiber deux mots distincts qui ont le même codage
- Pour **prouver** qu'un code est uniquement déchiffrable, c'est *a priori* plus compliqué : moralement, il faut exhiber la fonction de décodage et prouver qu'elle est correcte
- Dans la pratique, on verra qu'il existe des conditions suffisantes pour qu'un code soit uniquement déchiffrable, et que l'ensemble des codes u.d. qui satisfont ces conditions contient "presque" tous les codes "utiles".

Des exemples !

Pour chaque code, donné par l'ensemble des mots de code des symboles, dire s'il est ou non uniquement déchiffrable. S'il l'est, donner l'algorithme de décodage ; sinon, donner deux mots ayant le même codage.

- $\mathcal{C}_1 = \{0, 11, 101\}$
- $\mathcal{C}_2 = \{00, 01, 001\}$
- $\mathcal{C}_3 = \{0, 01, 10\}$
- $\mathcal{C}_4 = \{000, 001, 01, 1\}$
- $\mathcal{C}_5 = \{000100, 100101, 010101, 111000\}$
- $\mathcal{C}_6 = \{0, 01, 11\}$

D'autres propriétés combinatoires

- Les propriétés utiles des codes qui sont de nature combinatoire ne s'arrêtent pas à la déchiffrabilité. Même si un code est non ambigu, il peut être difficile de prendre une séquence codée "en marche" : c'est le problème de la synchronisation, ou du formatage.

D'autres propriétés combinatoires

- Les propriétés utiles des codes qui sont de nature combinatoire ne s'arrêtent pas à la déchiffrabilité. Même si un code est non ambigu, il peut être difficile de prendre une séquence codée "en marche" : c'est le problème de la synchronisation, ou du formatage.
- Autre problème potentiel, le "délai" : nombre de symboles de code qu'on peut avoir besoin d'examiner au-delà de ceux qui codent une lettre, avant de pouvoir décoder cette lettre.

D'autres propriétés combinatoires

- Les propriétés utiles des codes qui sont de nature combinatoire ne s'arrêtent pas à la déchiffrabilité. Même si un code est non ambigu, il peut être difficile de prendre une séquence codée "en marche" : c'est le problème de la synchronisation, ou du formatage.
- Autre problème potentiel, le "délai" : nombre de symboles de code qu'on peut avoir besoin d'examiner au-delà de ceux qui codent une lettre, avant de pouvoir décoder cette lettre.
- Exemple de $\mathcal{C}_6 = \{0, 01, 11\}$: déchiffrable, mais pour décoder $011 \cdots 1$ il faut connaître la **parité** du nombre de 1 consécutifs avant d'être capable de décoder la première lettre (délai non borné).

Codes préfixes

Un mot m est un **préfixe** d'un mot m', s'il existe un mot w tel que $m' = m.w$.

Définition

*Un code $C = \{M_1, \ldots, M_k\}$ est dit **préfixe** s'il n'existe pas deux mots de code M_i, M_j (autres que $i = j$) tels que M_i soit un préfixe de M_j.*

Théorème

Tout code préfixe est uniquement déchiffrable.