Error detection and correction

The structure of (Natural) language

Antoine Puissant

Enseignant : M. Filiol

2014 - 2015

Résumé

Ceci est le résumé

Table des matières

1 Introduction : Natural Languages as Mathematical Sources

Les langues naturelles sont très redondantes.
Voir stylométrie.
La plupart des logiciels conçut pour casser des mdp ne prennet pas en compte les accents.
L'approximation d'ordre 0 est bien, mais pas suffisante.
L'approximation d'ordre 1 tiens compte de la fréquence des lettres, on est alors plus précis.
L'approximation d'ordre 2 va alors faire des bigrammes : par exemple, en français, la lettre « q » et la lettre « u » sont quasiment tout le temps ensemble.
A environ $n = 5$, on a alors une très bonne approximation de la langue.
Slide 7 → TODO
En prenant des mots dépendant des uns des autres, on gagne en qualité du texte.

1.1 The entropy of English

1.1.1 A first approximation

On veut évaluer l'entropie de l'anglais H_E.
En utilisant le théorème de Shannon-McMillan (sur les sources ergodiques), on peut interpréter H_E en utilisant la formule suivante : $2^{nH_E} \approx t(n)$

1.2 Zipf law and word entropy

1.2.1 Introduction

Shannon regardait la fréquence relative aux mots.
La formule $H_E = \frac{H_W}{l(w)}$ n'est pas très précis. On a ici un formule qui est plus une approximation d'ordre 1.

1.2.2 Loi de Zipf

La probabilité d'occurence d'un mots dans un langage commence de manière très forte et descend rapidement.
Si p_n est la proba du mot n, elle est alors égale à $\frac{1}{n}$

1.3 Language redundancy

La compression de données consiste à réécrire les données avec un compte compact. Donc avant la compression, il faut connaître de langage pour connaître quelle information est redondante.
La redondance R du langage c'est ne nombre de bits utilisés pour envoyés le message moins le nombre de bits actuels du message.
Ainsi, la redondance est un pourcentage. On peut alors facilement écrire $l(n) \approx n(1 - \frac{R}{100})$.
On peut voir (c.f. slide 25) que la redondance est liée au corpus (par niveau de

langue) du texte. Dans l'exemple, on peut voir que la Bible a une redondance estimée à 41.4% tandis qu'un magasine littéraire tel que *The Atlantic Monthy* a une redondance estimée à 28.5%.
Il y a aussi une différence au sein des langues.
Slide 27 → à vérifier avec les formules
Il n'est pas systématiquement possible de comprendre un texte si $\frac{1}{4}$ des lettres sont enlevées. Cela dépend du quart des lettres qui sont enlevées. On ne doit pas le faire n'importe comment.
Avant de chiffre il est bon de compresser

1.4 Conclusion

La redondance est comprise entre 0.5 et 0.75.
L'entropie est comprise entre 1.19 et 2.38.
Il faut avoir la notion de corpus. Chaque langage est modelé en fonction du corpus.

1.5 Plus

Pour faire des essais, aller sur le projet Gutenberg.

1.6 Moodle

1.6.1 Question 1

$\forall p_1; P(p_1) = \frac{1}{27}$
$P(< space >) = \frac{1}{27}$
$Doncunmotestcomposede26caracteres.Onaalors$:
$l(w) = 26$

2 Introduction : The coding problem

On prend des « boules » de rayon ρ. Le volume c'est le nombre d'élements contenus dans cette sphère (éléments alors de 0 à $k-1$).
On appelle le « Packing radius » la plus grande valeur de rayon des boules de telle sorte que chaque centres des sphère soient disjointes.
On a alors le rayon de couverture. C'est le plus petit nombre pour des sphères de centres disjoints coupent l'espace. Un mot peut alors être dans une ou plusieurs sphères.
Après e-altération, on est capable de dire s'il y a eu dse erreurs.
Un code C est e-correcteur d'erreurs s'il peut décoder un code comprenant moins de e erreurs.

2.1 ...

On a un message :

$$m = [x_1, x_2, ..., x_7] \tag{1}$$

On ajoute un bit de parité :

$$f(m) = [x_1, x_2, ..., x_2, p] \tag{2}$$

Contrôle de la parité :

$$p = x_7 = (+)_{i=0}^{6} x_i = (\Sigma_{i=0}^{6} x_i) \tag{3}$$

Les codes de cartes bleues se font de la manière suivante :

$$456100324001236P \tag{4}$$

On va alors multiplier certain chiffres du code :

$$(4 * 2) \equiv 8[9] + 5 + ((6 * 2)\%9) + 1 + ... = 44 \tag{5}$$

On doit aussi avoir

$$p \equiv 0[10] \rightarrow p = 10 - (44\%10) = 6 \tag{6}$$

Ainsi, $p = 6$ et on a :

$$4561003240012366 \tag{7}$$

La distance minimale d'un code c'est la plus petite distance entre deux mots de code. On a code $R-$répétition. On a $q = 2$. C'est un code binaire. On a deux mots $000...R - fois, 111...R - fois$. On dit que C peut corriger $(d-1)$ mais pas d erreurs. En prenant notre code à $R-$répétition, si tous les 0 d'un mots sont changé (R erreurs), il est impossible de savoir que c'était des 0 à l'envoie.
On note un code de la manière suivante :

$$(n, M, d) \tag{8}$$

où n est la taille des mots, M les codewords et d est la distance minimale.
Pour des codes qui sont simultanément e-correcteur d'erreur et e'-détecteur d'erreurs, alors $d(C) \geq 2e + e' + 1$.

$$d - 1 = 2e + e' > e' \tag{9}$$

$$\frac{d-1}{2} = \frac{2e+e'}{2} = e + \frac{1}{2}e' > e \tag{10}$$

un code est dit maximal si, lorsque l'on rajoute un mot de code on change la distance minimale.
La valeur majorante (slide 11) est stable.
Quand on a un code (n, M, d), quel est le nombre de mots de code ?

2.2 Codes équivalents

Prodécé récursif de construction des codes.
Code équivalents = Si un code C' est obtenu en faisant des permutation de colonnes et de symboles d'un code C, alors ces deux codes sont équivalents.

2.3 Perfect codes

Situation idéale = partition parfaite
Code parfait quand les sphère de diamètre *rho* réalisent une partition ...
Unz condition nécessaire mais pas suffisante pour avoir un code parfait et d'avoir d impair.
Si le rayon de pavage est égal au rayon de couverture alors on a un code parfait.
Si n est odd, alors il pourra être parfait.

2.4 Making new codes from old ones

2.4.1 Extension de code

On va augmenter la taille des vecteurs.

2.4.2 Poinçonnage de code

On va poinçonner les codes. On va enlever certaines parties des vecteurs. On conserve la taille minimale en envoyant moins d'information.
$N \leq M$

2.4.3 Raccourcissement d'un code

On garde seulement les mots de code d'un code qui ont à un endroit donné un symbole donné. On touche alors au nombre de vecteurs et à la taille de ces derniers. La distance minimale augmente donc on peut corriger plus.

2.4.4 Augmentation d'un code

2.4.5 Somme de deux codes

2.5 Conclusion

3 Linear error-correcting codes

3.1 Introduction

On va structurer les codes.
Un code linéaire est n'importe quel EV d'un EV général.
On note ainsi un code

$$[n, k, d]$$

Avec k la dimension de l'espace.
Dimension k = base fait k.
Un code linéaire $[nk, d]$ peut s'écrire comme code général de la façon suivante : (n, q^k, d).
K vecteur de taille n (k lignes, n colonnes.
Poids de Haming = nombre de données non nulles dans un code binaire.
La matrice de parité peut être plus petite qua la matrice génératrice.

3.2 Using linear codes (encoding)

$$\Sigma_{i=1}^{k} a_i r_i k \tag{11}$$

3.3 The minimum distance of linear codes

Coset = translaté

3.4 Minimum-distance Decoding for Linear Codes - Syndrome

Coset leader = coset au poids minimal

3.5 Application

3.6 Conclusion

4 Exercice corrigé