REFEYTON
Mattias

SI 221 : TP chaînes de Markov

2. Chaîne de Markov

2.a Matrice de transitions

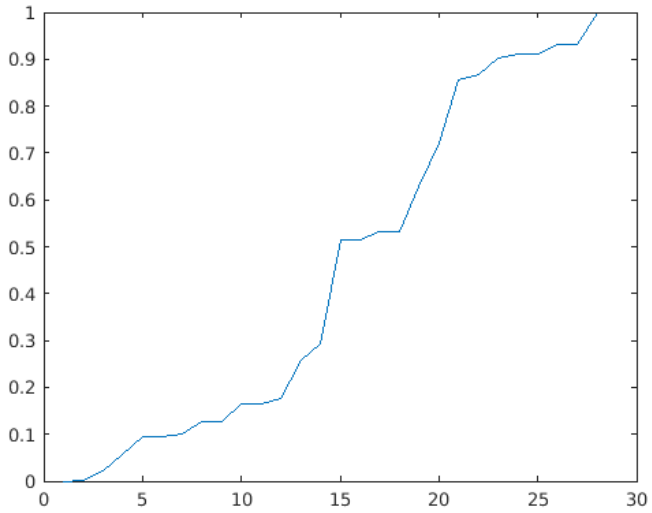
La première ligne de la matrice de transition donne la probabilité qu'une lettre suive le caractère spécial de début de mot ; autrement dit, c'est la probabilité qu'une lettre soit la première lettre d'un mot de la langue anglaise. La dernière colonne donne la probabilité qu'une lettre soit suivie du caractère spécial de fin de mot ; autrement, qu'elle soit la dernière lettre d'un mot.

Pour chaque lettre, la lettre qui a la plus grande chance de suivre est donnée ci-dessous (une absence de lettre indique la fin d'un mot):

a:n	b:e	c:o	d:	e:	f:	g:	h:e	i:n	j:o	k:	l:e	m:e	n:
	o:n	p:e	q:u	r:e	s:	t:h	u:r	v:e	w:a	x:t	y:	z:e	

2.b Générer un mot

La fonction de répartition est utile pour passer d'un état t à un état t+1 : on calcule la fonction de répartition de l'état t ; c'est une fonction f positive croissante telle que f(28) = 1, car il y a 28 caractères possibles en comptant les caractères spéciaux de début et de fin de mot. On tire ensuite un nombre aléatoire x entre 0 et 1, puis on détermine l'entier k tel que f(k+1) > x et f(k) ≤ x. Ce k est alors notre état t+1. Cette approche est correcte car la mesure de l'ensemble des x tels qu'un k particulier est sélectionné est égal à la probabilité que l'état t passe à l'état k, c'est à dire la cellule `matrice_trans(t, k)`.



Fonction de répartition pour la lettre 'a'

Voici 20 mots générés par cette méthode : cice, athespoftw, te, f, t, sutt, his, mele, hamivoupatharafareshonckitsond, g, brompll, soeangh, manong, non, hompend, fofowaris, cr, ds, owofoter, thalifange

3. Générer une phrase

On modifie la matrice de transition de manière à ce que l'état 28 (fin de mot) ait 90 % de chance d'amener à l'état 1 (début de mot) et 10 % de chance d'amener à l'état 29 (fin de phrase). De plus, l'état 29 a 100 % de chance d'amener à lui-même.

```
for i = 1 : 28
    matrice_trans(29, i) = 0;
end;
matrice_trans(29, 29) = 1;
matrice_trans(28, 1) = 0.9;
matrice_trans(28, 28) = 0;
matrice_trans(28, 29) = 0.1;
```

Des exemples de phrases générées sont ci-après :

lo hf prd oy anliales gr
tis anthicthes
mrersoreul funy
ernd cleg yomed alacom indans cat hedy are boo wasthono this onouesthastlf hearthuly chang
ge athevitourethis tol k braned figeicacomes the bly ofrs ng ghe vedurieconcthen ingnt
tivivins powand miorete then indenoun a vilat ccieanssty whe ly thean arveanftawe bloue he t silighe athovas f y ind who theby the atrecane f ama tols as git ofose phaiven lin a deepl or
cad t nomuntoff t t

4. Reconnaissance de la langue

La phrase « to be or not to be » en langue française a une vraisemblance de 5,9602e-30 ; et en Anglais 8,1129e-20 ; soit 10 milliards de fois plus probables (ou plutôt moins improbable).

En Français, « être ou ne pas être » a une vraisemblance de 1,1457e-19 en Français et de 4,4623e-24 en Anglais.