

Travaux Pratiques – Chaînes de Markov

CES Data Science

Septembre 2015

Laurence Likforman-Sulem, Chloé Clavel, Marc Sigelle

TELECOM ParisTech

1. Fonctions dont vous pouvez avoir besoin

cumsum : fonction de répartition : somme cumulative des termes du vecteur p

```
import numpy as np  
np.cumsum(p)
```

random : génère un nombre aléatoire entre [0 1] selon une loi uniforme.

```
import numpy as np  
np.random.random()
```

load : chargement d'un fichier

```
filename_A= 'bigramenglish.txt'  
np.loadtxt(filename_A)
```

2. Chaîne de Markov

On veut générer des mots dans une langue donnée en modélisant la formation de ces mots par une chaîne de Markov. Les 28 états du modèle correspondent aux 26 lettres de l'alphabet auxquelles on ajoute un état 'espace initial' (état 1) et un état 'espace final' (état 28) qui sert à terminer les mots.

La correspondance entre la valeur numérique d'un état et un caractère est la suivante : l'état 1 correspond à un espace (avant le début d'un mot) et l'état 28 à celui d'un espace en fin de mot. Les états 2 à 27 correspondent aux caractères de a à z.

On utilisera une chaîne de Markov ergodique entre les 26 états correspondants aux lettres de l'alphabet.

2.a Matrice de transitions

'bigramenglish' est la matrice des transitions pour l'anglais (bigrams) entre deux symboles (caractères ou espaces). Le terme générique (i,j) de la matrice de transition correspond à la probabilité de transiter vers l'état j à partir de l'état i .¹

¹

A quelles probabilités correspond la première ligne de la matrice de transition ? et celles de la dernière colonne ?

Pour chaque lettre de l'alphabet, indiquer la transition la plus fréquente depuis cette lettre.

2.b Générer un mot

On veut générer un mot à partir de l'état initial 1 (espace).

Ecrire une fonction qui génère un état (à $t+1$) à partir de l'état courant (à t) et à l'aide de la matrice de transitions et de la fonction de répartition.

Afficher sur un graphique la fonction de répartition pour une ligne de la matrice de transition et expliquer son rôle pour la génération de l'état à $t+1$.

Utiliser cette fonction pour générer les autres lettres du mot, état par état jusqu'à aboutir à l'état final (28). Donner des exemples de mots générés.

2.c Générer une phrase

On veut générer une suite de mots (phrase). Créer un état final de phrase (état 29) dont la probabilité de transition vers cet état depuis un état final de mot est 0.1. Modifier la matrice de transition en conséquence. Donner des exemples de phrases générées.

3. Reconnaissance de la langue

Calculer la vraisemblance de la phrase « to be or not to be » (multiplier les probabilités de transition) en fonction du modèle de langue, anglaise et française. De même calculer la vraisemblance de "etre ou ne pas etre" suivant les deux modèles. Charger la matrice des transitions entre caractères pour le français.