ENSTA ParisTech

MA201

TP numéro 4 : Chaines de Markov

Année académique 2018-2019

1 Jeu de pile ou face

Deux joueurs A et B jouent à un jeu de pile ou face, de façon itérative. La probabilité de face est $p \in [0, 1]$. On suppose que les tirages successifs sont indépendants et identiquement distribués. On note X_n la fortune du joueur A à la n-ième itération. A chaque itération, chaque joueur mise 1 euro : A mise sur face à chaque lancer, B mise sur pile à chaque lancer. Le jeu se termine lorsqu'un des deux joueurs est ruiné, on note T l'itération où le jeu se termine. La fortune initiale du joueur A est notée a, celle de B est notée b, on note $c = a + b$. On cherche à calculer la probabilité que A gagne, qu'on note
$f(a) = P(X_T = c | X_0 = a)$.

1. Ecrire une équation de récurrence sur la suite $f(n)$, pour n compris entre 0 et c.

2. A partir de cette équation, déterminer $f(a)$ en fonction de a, c, et p ; dans le cas où p est différent de $1/2$.

3. Même question si $p = 1/2$.

2 Exercice 2 : Traitement du langage naturel

Nous allons travailler sur une base de données de critiques de films d'IMDB (Internet Movie Database : https ://www.imdb.com/). Chacune de ces 1959 critiques a été cataloguée comme une revue positive ou négative du film qu'elle traite. Nous allons entrainer un système automatique pour classifier automatiquement si une critique est positive ou négative.

prétraitement Une analyse de texte doit systématiquement comporter une première partie de prétraitement. Nous allons pour cela utiliser le Natural Language Processing Toolkit (NLTK) de Python. La base de données de critiques de film "movie-reviews.csv" est tout d'abord chargée (dans la tranche de code notée 'Partie 1') et stockée dans une liste "data", telle que data[i] est la $i$ème critique, et labal[i] contient si la critique est positive (+1) ou négative (-1). Les mots trop fréquents (qui sont listés dans le code) sont tout d'abord supprimés. On effectue ensuite une partie de "stemming", qui consiste à ne garder que le radical d'un mot, en supprimant les conjugaisons, accords, etc. On enlève également les ponctuations et les problèmes de casse (majuscules et minuscules). Ce prétraitement est effectué dans la partie notée "partie 2" dans le fichier

Reconnaissance par mots isolés

1. Nous allons tout d'abord observer une loi appelée Loi de Zipf (https ://fr.wikipedia.org/wiki/Loi de Zipf), qui est une loi d'observation empirique sur les mots d'un texte. Cette loi stipule que, en ordonnant les mots du plus fréquent au moins fréquent et en notant donc n le n-ième mot le plus fréquent, la fréquence des mots d'un texte suit une loi de type $f = K/n$.

   Dans la tranche de code notée "partie 3", on utilise une outil fourni par la librairie NLTK appelé "fdist". Il s'agit d'une table de hash qui stocke les fréquences des mots et comporte des fonctions utiles pour analyser le contenu d'un texte. Dans le tableau "sorted-lis", les fréquences d'apparition des mots, ordonnés du plus fréquent au moins fréquent, sont fournis. Visualisez les et commentez. Estimez ensuite les paramètres K et α en estimant la distribution des mots selon une loi $f = K/n^{\alpha}$. Commentez.

2. On suppose que tous les mots au sein d'une critique sont indépendants. En notant p_i^1 la probabilité d'apparition du mot i dans une critique positive, et p_j^{-1} la probabilité d'apparition du mot j dans une critique négative, estimer ces probabilités pour tous les mots en utilisant *fdist* et en conservant en apprentissage 800 critiques positives et 800 critiques négatives.

3. Classifiez les critiques de l'ensemble de test. On considère que la critique est positive si sa vraisemblance est supérieure en utilisant les probabilités p_i^1 et négative sinon. Commentez.

4. On essaie à présent de classifier les critiques en utilisant des bigrammes. Pour pouvoir réutiliser le travail précédent, on traitera les bigrammes comme des entités discrètes (comme des "mots"). On notera q_i^1 la probabilité d'apparition du bigramme i dans une critique positive, et q_j^{-1} la probabilité d'apparition du bigramme j dans une critique négative, estimer ces probabilités pour tous les mots en utilisant *fdist* et en conservant en apprentissage 800 critiques positives et 800 critiques négatives.Classifiez les critiques de l'ensemble de test. On considère que la critique est positive si sa vraisemblance en supérieure en utilisant les probabilités q_i^1 et négative sinon. Commentez.