MAT 4506
Méthodes statistiques pour la segmentation dans les chaînes de Markov Cachées

Erwan Le Blévec

November 12, 2021

1 Introduction

Ce TP est consiste en l'étude de la méthode de segmentation par maximum de vraisemblance à postériori. La principale différence de cette approche, par rapport au maximum de vraisemblance, tient en la considération de la loi dite *a priori* de X.
Cherchant à maximiser $\mathbb{P}(X|Y)$ la formule des probabilités conditionnelles nous donne :

$$\mathbb{P}(X|Y) = \mathbb{P}(Y|X)\mathbb{P}(X)$$

On observe toujours la présence du terme de maximum de vraisemblance mais cette fois couplé à la loi *a priori* de X.
Dans notre cas la v.a X ne peut prendre que deux valeurs : cl1 ou cl2. La règle de décision correspondante est énoncée dans l'équation (4) de l'énoncé. Cette approche implique donc d'estimer, en plus, la loi *a priori* de X sur une des classes. Il existe de nombreuses approches pour cela, dans ce TP nous nous concentrerons sur deux.

2 Apport des méthodes bayésiennes de segmentation

2.1 Question 2

Il est question ici de comparer la précision de la segmentation par maximum de vraisemblance à postériori en considérant l'estimateur empirique pour la loi *a priori* de X.

Le code présenté en annexe permet d'obtenir les résultats suivants :

Signal	Bruit 1	Bruit 2	Bruit 3	Bruit 4	Bruit 5
Signal 1	0.00	0.20	0.28	0.0	0.36
Signal 2	0.00	0.18	0.31	0.0	0.38
Signal 3	0.00	0.18	0.31	0.0	0.38
Signal 4	0.00	0.18	0.31	0.0	0.38
Signal 5	0.00	0.18	0.31	0.0	0.38
Signal 6	0.00	0.18	0.31	0.0	0.38

Table 1: Erreur moyenne (500 réalisations) avec la méthode de MAP

Les valeurs affichées dans ce tableau ont été arrondies au centième près. Il est cependant à noter que 0.0 correspond à une valeur exacte contrairement à 0.00 qui est une valeur approchée. Les valeurs du paramètre "p1" de l'estimateur empirique étaient respectivement :

$$0.26; 0.49; 0.5; 0.5; 0.5; 0.5; 0.5$$

Soit une (quasi) équiprobabilité des classes pour tous les signaux sauf le premier. Ce point explique la similarité entre les valeurs de ce tableau et celles du TP précédent pour les 5 derniers signaux : lorsque la loi *a priori* de X est équiprobable on peut simplifier les termes en $\mathbb{P}(X)$ dans l'équation (4). On retrouve alors l'expression du classificateur par maximum de vraisemblance.
Remarques sur le choix de l'estimateur empirique : Solutions possibles pour l'estimation

- Estimateur empirique (estimateur avec le maximum de vraisemblance)
- Chaîne de Markov (étudiée dans la suite du TP)
- Réseaux de neuronnes, Kernel : Trop fin pour ce type de simulation dont les critères restent simples, le rapport complexité/performance n'est pas intéressant.

2.2 Question 4

Objectif : Comparer les segmentations par maximum de vraisemblance et par maximum de vraisemblance à postériori sur des signaux générés aléatoirement via l'estimateur empirique

Le nombre de signaux générés est de 5, sur ces derniers nous avons appliqués les 5 bruits définis dans le TP précédent, soit un total de 25 signaux considérés.
L'erreur moyenne est calculée sur 500 réalisations (les variations proviennent des bruits gaussiens).

Bruit	Bruit 1	Bruit 2	Bruit 3	Bruit 4	Bruit 5
Segmentation	MAP, MV	MAP, MV	MAP, MV	MAP, MV	MAP, MV
p1 = 0.18	0.00 | 0.00	0.10 | 0.12	0.20 | 0.31	0.0 | 0.0	0.20 | 0.29
p1 = 0.34	0.00 | 0.00	0.12 | 0.13	0.24 | 0.31	0.0 | 0.0	0.25 | 0.31
p1 = 0.47	0.00 | 0.00	0.14 | 0.15	0.28 | 0.31	0.0 | 0.0	0.31 | 0.33
p1 = 0.66	0.00 | 0.00	0.17 | 0.17	0.31 | 0.32	0.0 | 0.0	0.37 | 0.38
p1 = 0.87	0.00 | 0.00	0.18 | 0.24	0.28 | 0.31	0.0 | 0.0	0.36 | 0.50

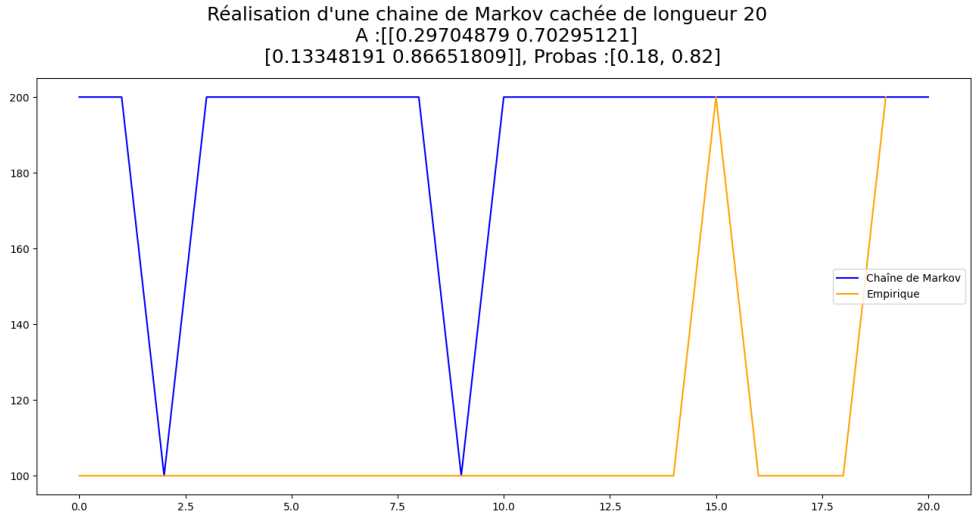
Table 2: Comparaison des erreurs moyennes des méthodes MAP et MV

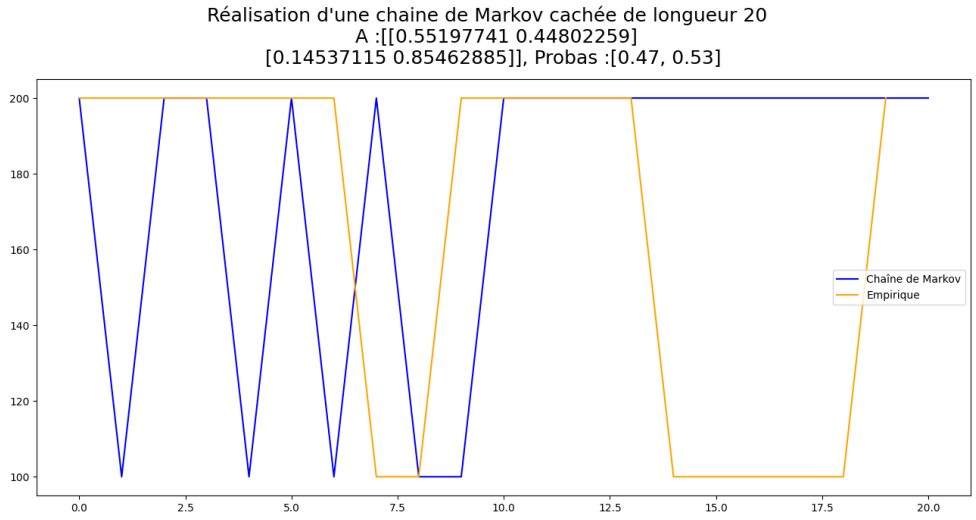
On observe à nouveau que les erreurs sont d'autant plus proches que la loi *a priori* de X tant vers une loi équiprobable (ligne 3). À l'inverse lorsque les probabilités des classes diffèrent fortement la segmentation par maximum de vraisemblance à postériori est bien plus intéressante que celle par maximum de vraisemblance seulement.

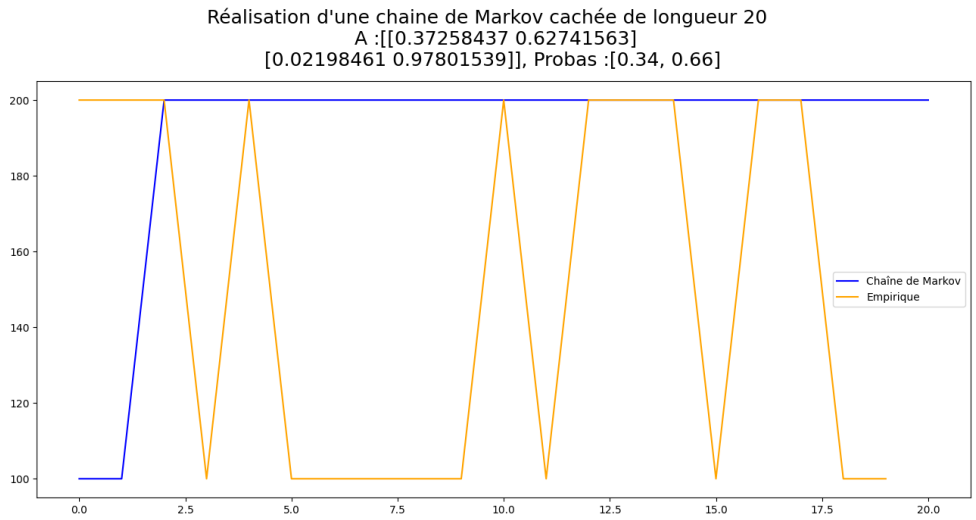
3 Les chaînes de Markov

3.1 Question 3

Quelques générations de signaux de longueurs 20 pour une représentation par des chaînes de Markov avec A et [p1, p2] définis aléatoirement :

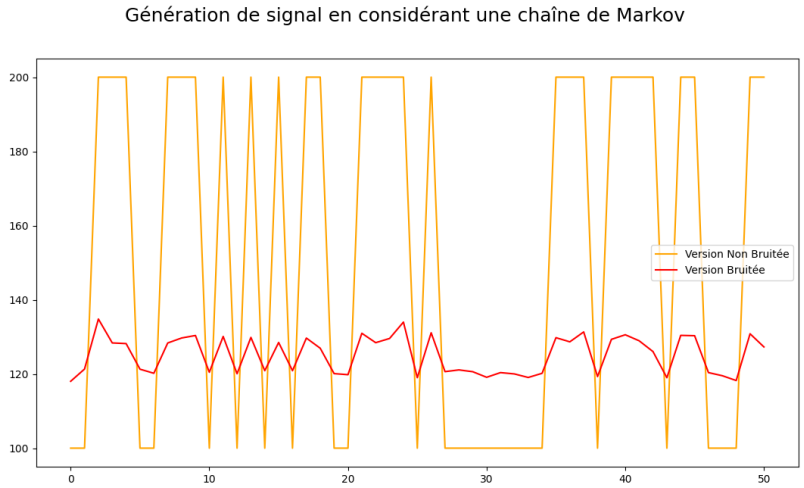






3.2 Question 4

De même :



C'est ce signal bruité qui sera ensuite traité par les algorithmes de segmentation mis en oeuvre dans la partie *IV* (voir prochain TP).

4 Annexe

Fonctions utilisés dans les différentes questions.

Partie 2

```python
def calc_probaprio2(X):  # Question 1
    return [np.unique(X, return_counts=True)[1][0]/len(X), np.unique(X, return_counts=True)[1][1]/len(X)]


def MAP_MPM(Y, params, p1, p2): #Question 1
    return ((stats.norm.pdf(Y, params['m1'], params['sig1'])*p1 >= stats.norm.pdf(Y, params['m2'], params['sig2'])*p2)*params['cl1'] +
            (stats.norm.pdf(Y, params['m1'], params['sig1'])*p1 < stats.norm.pdf(Y, params['m2'], params['sig2'])*p2)*params['cl2']
            )


def simul2(p1, p2, n, cl1, cl2): # Question 3
    generated_signal = []
    for i in range(n):
        if np.random.rand() >= p1:
            generated_signal.append(cl1)
        else:
            generated_signal.append(cl2)
    return np.array(generated_signal)
```

Partie 3

```python
def tirage_classe2(p1, p2, cl1, cl2): # Question 1
    a = np.random.rand()
    return cl1*(a <= p1) + cl2*(a > p1)


def genere_chaine2(n, cl1, cl2, A, p): # Question 2
    chain = []
    chain.append(tirage_classe2(p[0], p[1], cl1, cl2))

    for i in range(0, n):  # Idem pour cette boucle
        chain.append((chain[i-1] == cl1)*tirage_classe2(A[0][0], 0, cl1, cl2) + (chain[i-1] == cl2)*tirage_classe2(A[1][0], 0, cl1, cl2))
    return np.array(chain)


def rand_matrix_gen(): # Pour générer une matrice A de manière aléatoire
    (a, b) = (np.random.rand(), np.random.rand())
    return np.array([[a, 1-a], [b, 1-b]])
```