3INSA MARKOV

TP Modèles de Markov cachés

Introduction

Le but de ce TP est de mettre en œuvre les algorithmes de base liés au modèle de Markov cachés. Dans un premier temps, on pourra s'appuyer sur le modèle simple suivant :

$$\pi = \begin{pmatrix} 0.5 & 0.5 & 0 \end{pmatrix} \qquad A = \begin{pmatrix} 0.1 & 0.5 & 0.4 \\ 0 & 0.8 & 0.2 \\ 0.2 & 0.1 & 0.7 \end{pmatrix} \qquad B = \begin{pmatrix} 0.3 & 0.7 \\ 0.7 & 0.3 \\ 0.5 & 0.5 \end{pmatrix} .$$

Le modèle ci-dessus n'est donné qu'à titre d'exemple. Les fonctions demandées dans le TP doivent être génériques et prendre en argument les paramètres π, A et B définissant une chaîne de Markov cachée.

Vous êtes invité à utiliser python comme langage de programmation pour ce TP. À défaut, matlab ou R peuvent être utilisés. Dans tous les cas, vous ne bénéficierez que d'une aide minimale de votre encadrant en terme de programmation, sauf peut-être en python !

À la fin de la seconde séance, vous devez rendre un rapport contenant les fonctions programmées, avec leurs commentaires, ainsi que les résultats obtenus et les conclusions que vous pouvez en tirer. Des questions vous guident dans le texte du TP.

Tirage aléatoire

1. Dessinez la représentation sous forme d'automate correspond au modèle donné ci-dessus en exemple.

2. Écrire une fonction hmm_sample qui prend en entrée une chaîne de Markov caché et une longueur et qui réalise un tirage aléatoire d'un échantillon selon la loi définie par le modèle.

   On génerera quelques échantillons en calculant à chaque fois la (log-)probabilité de l'échantillon généré, en affichant de manière séparée les deux termes qui la composent, i.e., $P[X]$ et $P[Y|X]$.

Algorithme de Viterbi

1. Écrire une fonction treillis permettant de visualiser sous forme de *heatmap* les nœuds du treillis, *i.e.*, la matrice contenant les probabilités $\ln P[y_t|i] \quad \forall t, i$. En R, on pourra par exemple utiliser les fonctions heatmap3 ou image. En python, la fonction imshow de la librairie pyplot. Cette visualisation n'a que peu de sens pour des petites séquences et un faible nombre d'état. On cherchera donc à visualiser un treillis pour un modèle avec quelques dizaines d'état (une topologie gauche-droite, i.e., sans possibilité de revenir en arrière dans les états, est particulièrement intéressante pour cette partie du TP) et une séquence d'une cinquantaine d'observations.

2. Étendre la fonction treillis de manière à afficher en surimposition (par exemple en mettant une valeure arbitrairement élevée ou avec l'un des arguments highlightCell ou colorCell de la fonction heatmpa3 en R) pour chaque instant l'état i qui maximise $\ln P[y_t|i]$. L'intérêt est ici de visualiser l'appariement entre observations et états qui serait fait si l'on ne tenait compte que du terme $P[Y|X]$. En d'autres termes, on souhaite mettre en exergue $\hat{x}_t = \arg\max_i P[Y_t = y_t|X_t = i]$ à chaque instant.

3. Comme vu en cours, l'algorithme de Viterbi permet de trouver le meilleur chemin dans le treillis, i.e., celui qui maximise la probabilité conjointe $P[X, Y]$. Écrire la fonction viterbi qui retourne la séquence d'état X la plus probable étant donnée le processus observé $Y = y$ à l'aide de l'algorithme de Viterbi.

   Pour le modèle donné en exemple, la meilleure séquence d'état pour l'observation $Y = \{abba\}$ est $X = \{2222\}$ avec une log-probabilité $\ln P[X, Y] \simeq -4.5$.

4. Pour quelques séquences générées avec votre fonction hmm_sample, comparez l'alignement et les probabilités lors de l'échantillonage avec celles obtenues par Viterbi. Commentez.

5. Reprendre la fonction treillis de manière à permettre d'afficher en surimposition le meilleur chemin résultant de l'algorithme de Viterbi. Comme à la question 1, on pourra fixer une valeur arbitrairement élevée pour les cases corresonpdant au meilleur chemin ou simplement entouré les cases.

   Comment expliquez-vous les différences avec le chemin visualisé à la question 1 ?

Comparaison avec d'autres techniques d'estimation des états cachés

Estimer l'état caché correspondant à un instant connaissant la séquence observée ou une partie de la séquence observée peut se faire selon d'autres méthodes que Viterbi, *e.g.*, par la récursion *forward* donnant à chaque instant $P[Y_1, \ldots, Y_t, X_t]$ ou par le calcul exact des probabilités $\gamma_i(t) = P[X_t = i|Y]$ à l'aide des récursions *forward* et *backward*. Dans cette seconde partie du TP, nous souhaitons comparer ces différentes méthodes.

1. Écrivez une fonction forward qui retourne la matrice N×T (où N est le nombre d'état du modèle et T la longueur de la séquence) contenant l'ensemble des probabilités *forward* étant donnée une séquence d'observations Y. Vous pouvez vous inspirer largement de votre fonction viterbi pour ce faire.

2. Tester votre fonction forward. Qu'observe-t-on lorsque la longueur de la séquence devient très longue ? Discutez la solution dites de *scaling* décrite dans `http://people.irisa.fr/Guillaume.Gravier/tmp/INSA/scaling.pdf` pour pallier le problème observé ? Pour la suite du TP, on se contentera de travailler avec des séquences suffisamment courtes pour contourner le problème.

3. Grâce à la fonction forward, on peut estimer X_t en prenant pour chaque instant l'état qui maximise $\alpha_i(t)$, i.e., $\hat{x}_t = \arg\max_i \alpha_i(t)$. Comparez cette estimation à celle obtenue par l'algorithme de Viterbi. On pourra pour cela regarder sur quelques dizaines de séquences la proportion d'estimation qui diffèrent (chose que l'on pourrait d'ailleurs faire pour la question 3 de la partie précédente).

   Discutez les avantages et inconvénients respectifs de l'estimation des états cachés par l'algorithme de Viterbi et à l'aide de la variable *forward*. Par exemple, pensez-vous que l'estimation par les variables *forward* soit meilleure (plus fiable) que celle obtenue par Viterbi (et pourquoi) ? Dans quelle(s) situation(s) pensez-vous que l'approche par les variables *forward* est utilisée ?

Pour aller plus loin si vous avez du temps, vous pouvez étudiez l'esimateur donnée par $X_t = \arg\max_i \gamma_i(t)$. Écrivez la fonction gamma qui retourne les quantités $\gamma_i(t)$ étant donnée une séquence d'observation et comparer l'estimation de X_t à celles obtenues par Viterbi et par $\alpha_i(t)$.

Pour finir, et même si vous n'avez pas le temps de programmer la fonction gamma, élargissez la discussion de la question 3 précédente à l'estimation de l'état avec l'estimateur γ : intérêt de chacun au regard des critères sur lesquels ils sont fondés ; complexité (et donc temps) de calcul ; situations dans lesquels ils sont le plus adaptés ; *etc.* Quelles autres estimateurs de l'état caché peut-on envisager ?