Éléments de processus stochastiques

Chaînes de Markov cachées pour la reconnaissance vocale

3ème BAC IC, Majeures en biomédical, électricité, informatique et physique

Profs. Louis Wehenkel et Pierre Geurts
Assistante : Laurine Duchesne

Année académique 2016-2017

Ce document décrit le projet du cours d'Eléments de Processus Stochastiques pour les étudiants de troisième année en Bac ingénieur civil qui ont choisi la majeure en biomédical, électricité, informatique, ou physique.

Le travail sera réalisé par groupe de trois étudiants. Les consignes générales pour réaliser le travail, et celles relatives à la manière de rédiger les rapports et préparer la présentation orale, sont les mêmes pour toutes les sections et sont communiquées par le Professeur V. Denoël, coordinateur du cours.

Des éléments complémentaires de théorie nécessaires pour réaliser ce projet seront expliqués lors des séances d'encadrement du travail. Nous encourageons vivement les étudiants à se documenter eux-mêmes sur les sujets abordés dans ce projet. Des pointeurs vers des sources de théorie/pratique seront fournis au fur et à mesure de la réalisation du travail.

Le travail sera encadré au moyen de séances de questions/réponses qui seront programmées au fur et à mesure de l'avancement du projet. Ces séances auront lieu les mardis de 10h45 à 12h45 au local S33 (B37, Mathématique). Seules les questions posées lors de ces séances donneront lieu à des réponses de la part des encadrants.

Contexte général et objectifs

L'objectif général du projet est de développer un algorithme permettant de reconnaître une séquence de chiffres émis vocalement. Ce système pourrait par exemple être utilisé pour débloquer par la voix un téléphone portable ou encore pour récupérer un numéro de carte de visa dans un système d'achat par téléphone. Pour construire ce système, on se basera sur une modélisation des signaux acoustiques à l'aide de modèles particuliers de processus stochastiques appelés 'modèles de Markov cachés'. Ces derniers sont une extension des chaînes de Markov et sont très fréquemment utilisés pour la reconnaissance vocale mais également dans beaucoup d'autres applications.

Le projet est divisé en deux parties. La première partie du projet, relativement théorique et dirigée, a pour but de vous familiariser avec les modèles de Markov cachés et les techniques d'analyse associées. La deuxième partie, qui constitue le cœur du projet, vise à mettre en œuvre concrètement les modèles de Markov cachés pour implémenter le système de reconnaissance de chiffres.

1 Première partie : modèles de Markov cachés

Dans cette première partie du projet, on se propose d'aborder les différentes utilisations possibles des modèles de Markov cachés, en se focalisant sur celles qui sont en lien direct avec l'application visée en reconnaissance vocale.

Définitions et notations

Dans cette section, nous fournissons les définitions et notations minimales nécessaires pour la première partie du projet. Nous vous encourageons néanmoins à consulter d'autres références pour obtenir plus de détails à propos des modèles de Markov cachés (p.ex. [Rab90, Mur12]). Des références supplémentaires seront données sur le site web du projet.

Soit deux suites de variables aléatoires $\{X_1, X_2, \ldots, X_t, \ldots\}$ et $\{Y_1, Y_2, \ldots, Y_t, \ldots\}$. Ces suites définissent un modèle (ou une chaîne) de Markov caché ssi, pour tout $t \geq 1$, la distribution conjointe de t premières variables peut se factoriser comme suit :

$$P(X_1, X_2, \ldots, X_t, Y_1, \ldots, Y_t) = P(X_1) \prod_{l=2}^{t} P(X_l|X_{l-1}) \prod_{l=1}^{t} P(Y_l|X_l).$$

Les variables $\{X_1, X_2, \ldots\}$ forment une chaîne de Markov (d'ordre 1) et sont supposées cachées, c'est-à-dire non observées. Les variables $\{Y_1, Y_2, \ldots\}$ sont les observations qui sont indépendantes entre elles conditionnellement à la séquence d'états et sont telles que l'observation au temps t ne dépend que de l'état au même temps. Dans cette partie, on supposera que le modèle de Markov caché est discret et on notera sans restriction $\{1, \ldots, N\}$ l'ensemble des valeurs possibles des variables X_i, appelées les états cachés, et $\{1, \ldots, M\}$ l'ensemble des valeurs possibles des variables Y_i, appelées les observations.

Dans le cas d'un modèle invariant dans le temps, on supposera que les probabilités $P(X_{t+1} = j|X_t = i)$ et $P(Y_t = j|X_t = i)$ ne dépendent pas du temps t. Dans ce cas, un modèle de Markov caché est défini par un triplet $\lambda = \langle \pi, A, B \rangle$ où :

— π est un vecteur ligne de dimension N dont la $i$ème composante vaut $P(X_1 = i)$,
— A est la matrice de transition $N \times N$ dont l'élément i, j vaut $P(X_{t+1} = j|X_t = i)$.
— B est la matrice d'émission $N \times M$ dont l'élément i, j vaut $P(Y_t = j|X_t = i)$.

On définit généralement trois problèmes classiques liés à l'utilisation d'un modèle de Markov caché [Rab90] :

— Problème 1 : Etant donné une séquence d'observations $y_1, y_2, \ldots, y_t$ et un modèle λ, calculer la probabilité de cette séquence $P(Y_1 = y_1, \ldots, Y_t = y_t|\lambda)$ étant donné le modèle.
— Problème 2 : Etant donné une séquence d'observations $y_1, y_2, \ldots, y_t$, trouver la séquence d'états correspondante la plus probable étant donné le modèle, c'est-à-dire calculer :

$$\{x_1^*, \ldots, x_t^*\} = \arg \max_{x_1, \ldots, x_t} P(X_1 = x_1, \ldots, X_t = x_t|Y_1 = y_1, \ldots, Y_t = y_t).$$

— Problème 3 : Etant donné une séquence d'observations $y_1, y_2, \ldots, y_t$, trouver un modèle $\lambda = \langle \pi, A, B \rangle$ maximisant la probabilité $P(Y_1 = y_1, \ldots, Y_t = y_t|\lambda)$

Les premier et troisième problèmes sont directement pertinents pour la reconnaissance vocale et sont donc explorés dans les sections 1.1 et 1.2 ci-dessous. Le deuxième problème est résolu par l'algorithme de Viterbi, qui sera vu au cours.

Pour les questions des sections suivantes, on vous fournit dans le fichier matlab `hmm.m`, trois modèles de Markov cachés `hmm1`, `hmm2`, et `hmm3`. Pour répondre aux questions ci-dessous, vous pouvez utiliser soit les fonctions internes MATLAB ou de manière équivalente l'implémentation de ces fonctions dans la toolbox HMM MATLAB de Kevin Murphy [1]. Le texte ci-dessous fait référence à ces deux types de fonctions et suppose donc que la toolbox HMM de Kevin Murphy a été préalablement installée.

1.1 Vraisemblance d'une séquence d'observations

Le calcul de la probabilité (ou *vraisemblance*) d'une séquence d'observations étant donné un modèle (Problème 1 ci-dessus) est utile notamment pour déterminer parmi plusieurs modèles lequel a la plus vraisemblablement généré la séquence observée. Les questions ci-dessous explorent cette utilisation.

Questions :

1. Ecrivez une fonction permettant de générer une séquence d'états et d'observations étant donné les matrices π, A, B définissant une chaîne de Markov cachée. Utilisez cette fonction pour générer 5 séquences d'observations de longueur 15 à partir de chacun des deux modèles `hmm1` et `hmm2` fournis. Sur base de ces séquences et des propriétés des modèles, décrivez la nature des séquences d'observations générées par chacun de ces deux modèles.
2. Montrez que la probabilité $P(Y_1 = y_1, \ldots, Y_t = y_t|\lambda)$ peut être calculée par l'expression suivante :

$$\sum_{x_1,\ldots,x_t} P(Y_1 = y_1, \ldots, Y_t = y_t|X_1 = x_1, \ldots, X_t = x_t, \lambda)P(X_1 = x_1, \ldots, X_t = x_t|\lambda), \quad (1)$$

   où la somme est calculée sur toutes les séquences d'états cachés possible. Expliquez comment calculer cette expression à partir des matrices π, A, et B.
3. La calcul (1) est inefficace pour de longues séquences. Une façon plus efficace de calculer cette probabilité est de noter que $P(Y_1 = y_1, \ldots, Y_t = y_t|\lambda) = \sum_i \alpha_t(i)$ où $\alpha_t(i) \triangleq P(Y_1 = y_1, \ldots, Y_t = y_t, X_t = i|\lambda)$ et que les $\alpha_t(i)$ peuvent se calculer comme suit :

$$\alpha_t(i) = \sum_{j=1}^{N} \alpha_{t-1}(j)P(X_t = i|X_{t-1} = j, \lambda)P(Y_t|X_t = i, \lambda),$$

   pour tout $t > 1$. Prouvez que cette dernière relation est effectivement vérifiée et expliquez comment elle permet de calculer plus efficacement la probabilité recherchée.
4. La fonction matlab `dhmm_logprob` implémente ce calcul (pour des raisons de stabilité numérique, elle renvoit cependant le logarithme de la probabilité). Générez 500 séquences d'observations de longueurs 15 à partir des modèles `hmm1` et `hmm2` et pour chacune de ces séquences, déterminez leur vraisemblance sur base des deux modèles à l'aide de `dhmm_logprob`. Calculez ensuite le pourcentage des 1000 séquences dont la vraisemblance est plus grande pour le modèle dont elles sont effectivement issues. Ce pourcentage est appelé la *précision*. Commentez votre résultat.

1. `https://www.cs.ubc.ca/~murphyk/Software/HMM/hmm.html`

1.2 Estimation des paramètres

Dans beaucoup d'applications, il est difficile de mettre au point un modèle directement à partir de notre connaissance du problème à modéliser. C'est particulièrement le cas pour la reconnaissance vocale (voir la deuxième partie). Il est donc intéressant de pouvoir estimer les paramètres du modèle de Markov à partir de données (Problème 3 ci-dessus). On parlera dans ce cas aussi d'*apprentissage* du modèle.

Questions :

1. Expliquez comment estimer les paramètres de la chaîne de Markov cachée au maximum de vraisemblance dans le cas où on observe à la fois la séquence des états cachés et la séquence des observations correspondantes, toutes deux supposées avoir été générées à partir de la chaîne.
2. Expliquez comment étendre l'approche au cas où on observe plusieurs séquences.
3. Si on ne dispose que de séquences d'observations, imaginez une méthode itérative pour estimer les paramètres de la chaîne en vous basant sur l'algorithme de Viterbi permettant de calculer la séquence d'états la plus probable étant donné une séquence d'observations (Problème 2 ci-dessus). NB : Cette approche est implémentée dans la fonction MATLAB `hmmtrain`, qui implémente également un autre algorithme plus efficace : l'algorithme Baum-Welch (qui sera brièvement expliqué lors des séances).
4. Générez une séquence d'observations suffisamment longue (par exemple 500) à partir du modèle `hmm3` et appliquez la fonction `hmmtrain` sur cette séquence pour estimer les matrices de transition et d'émission de la chaîne `hmm3` à l'aide des deux algorithmes, Viterbi et Baum-Welch. Comparez les matrices obtenues avec les deux algorithmes entre elles et avec les matrices réelles. Répétez l'expérience pour différentes tailles de la séquence d'observations et pour différents choix des matrices initiales (essayez notamment en partant des matrices réelles). Commentez vos résultats.
5. Expliquez pourquoi les modèles `hmm1` et `hmm2` ne peuvent pas être appris à partir d'une seule séquence d'observations, même très longue, contrairement au modèle `hmm3`.
6. Générez un ensemble de N séquences d'observations de longueur 15 à partir des modèles `hmm1` et `hmm2` et utilisez l'algorithme Baum-Welch, cette fois via la fonction `dhmm_em`, pour estimer sur base de ces échantillons les paramètres des deux modèles. Estimez ensuite la précision comme à la question 4 de la section 1.1 en remplaçant les vrais modèles par ceux estimés. Calculez également la précision pour l'identification de l'origine des $2N$ séquences utilisées pour l'apprentissage des modèles. Répétez l'expérience pour des valeurs de N croissantes et en faisant varier également le nombre d'états cachés des modèles estimés (de 1 à 6 par exemple). Commentez vos résultats.

1.3 Rapport et code

Pour cette première partie, vous devez nous fournir un rapport contenant vos réponses, concises mais précises, aux questions posées ainsi que les scripts matlab que vous avez utilisé pour y répondre. Dans le rapport final, cette partie ne devrait pas dépasser 6 pages.

2 Deuxième partie : reconnaissance de chiffres

L'objectif de cette seconde partie est de mettre au point un système de reconnaissance vocale de chiffres en se basant sur les techniques étudiées dans la première partie. On traitera le problème en deux étapes. La première consistera à mettre au point un système de reconnaissance de chiffres prononcés isolément. La deuxième partie consistera à traiter la reconnaissance de plusieurs chiffres prononcés les uns après les autres, en se basant sur les modèles générés pour la reconnaissance de chiffres isolés.

2.1 Reconnaissance de chiffres isolés

On vous demande d'écrire un programme prenant en entrée un fichier audio et fournissant en sortie le chiffre effectivement prononcé. Idéalement, le système devra être capable de fonctionner pour n'importe quel locuteur, homme ou femme.

Bien que d'autres techniques existent, l'approche que vous développerez devra se baser *sur des modèles de Markov cachés.* L'idée générale sera la suivante :

- Des fichiers audio seront collectés pour chacun des chiffres, idéalement obtenus de différents locuteurs pour assurer la robustesse du système.
- Les signaux d'onde sonore seront ensuite mis sous une forme facilitant la reconnaissance par l'extraction de ce qu'on appelle des "caractéristiques". L'extraction de ces caractéristiques ne faisant pas partie des compétences à acquérir dans le cadre de ce cours, des scripts vous seront fournis pour faire cette extraction. Libre à vous néanmoins de vous renseigner et de tester d'autres approches.
- Un modèle de Markov caché sera appris pour chaque chiffre en utilisant les techniques d'apprentissage des paramètres étudiées à la section 1.2.
- La reconnaissance d'un chiffre à partir d'un nouveau fichier audio se fera ensuite en calculant les caractéristiques à partir de ce fichier et en renvoyant le chiffre correspondant au modèle maximisant la vraisemblance de la séquence observée (calculée en utilisant l'algorithme étudié à la section 1.1).

Bien que l'approche soit relativement bien balisée, il y a beaucoup de problèmes à résoudre et de degrés de liberté qui peuvent être explorés pour arriver au système le plus robuste possible :

- Collecte de données : La collecte de données joue un rôle important. Pour que les modèles appris soient les plus robustes possibles, il faudra collecter un nombre suffisant d'échantillons sonores, obtenus de plusieurs locuteurs, pour chacun des chiffres. Nous vous invitons à collaborer entre groupes en vous échangeant les échantillons sonores que vous aurez générés.
- Observations numériques : Les caractéristiques extraites par les scripts qui vous seront fournis, qui donc constitueront les observations des modèles de Markov cachés, sont des vecteurs de $\mathbb{R}^n$, où n est typiquement de l'ordre $10 - 20$. Il existe deux solutions pour traiter des observations à valeurs numériques (et vectorielles) dans les modèles de Markov cachés : soit réduire ces observations à un ensemble discret de valeurs en utilisant ce qu'on appelle des techniques de *quantification vectorielle*, soit étendre les modèles de Markov cachés pour qu'ils puissent directement générer des observations sous la forme de vecteur de $\mathbb{R}^n$. L'extension la plus classique consiste à remplacer les distributions multinomiales discrète par des distributions gaussiennes multivariées, voire des mélanges de gaussiennes. Cette solution est implémentée dans la toolbox

HMM de Kevin Murphy [2].

— Choix des modèles : Le choix du paramétrage des modèles de Markov cachés joue également un rôle important. Il s'agira entre autres de choisir le nombre d'états cachés, le nombre d'états observés dans le cas de la quantification vectorielle, le nombre de gaussiennes dans le modèle de mélange, etc. Il pourrait également être intéressant d'imposer une structure particulière aux modèles (en fixant certaines valeurs à 0 dans les matrices initiales pour l'apprentissage). Il est ainsi courant d'utiliser des modèles dit "left-right" en reconnaissance vocale, c'est-à-dire imposer que la matrice de transition soit triangulaire supérieure [Rab90] [3]. Le choix d'un modèle n'est pas trivial : choisir un modèle trop simple ne permettra pas de capturer la complexité des signaux acoustiques, alors que choisir un modèle trop complexe demandera un nombre plus important d'échantillons pour être appris efficacement.

— Evaluation des performances : Pour évaluer la performance de votre système, par exemple pour guider le choix du modèle, il vous faudra le tester sur des échantillons sonores. Comme illustré dans la section 1.2, l'utilisation pour cette évaluation des échantillons utilisés pour l'apprentissage des paramètres des modèles donnera généralement une évaluation biaisée (trop optimiste) des performances. L'approche classique pour obtenir une évaluation non biaisée est de diviser les échantillons disponibles en deux sous-ensembles disjoints, le premier utilisé pour l'apprentissage des paramètres des modèles et le second réservé au test de ces modèles. D'autres stratégies sont cependant possibles.

2.2 Reconnaissance d'une séquence de chiffres

Dans un second temps, on vous demande de réfléchir à la manière d'utiliser les modèles précédemment développés pour reconnaître une séquence de chiffres prononcés les uns à la suite des autres. Une solution à ce problème consiste à fusionner les modèles de Markov cachés appris pour les chiffres isolés en un modèle de Markov caché plus gros (voir par exemple [Rab90]). Des solutions plus simples peuvent cependant également être envisagées, basées sur des approches plus heuristiques de découpage du signal sonore en les différents chiffres qui le composent.

2.3 Bonus

Si vous êtes intéressés, deux questions supplémentaires vous sont proposées en bonus :

— l'implémentation d'un système de détection d'intrus (s'il s'avérait que le locuteur a en fait prononcé un mot qui ne correspond pas à un chiffre) ;

— l'implémentation de votre solution pour la reconnaissance d'une séquence de chiffres.

Ces deux questions sont facultatives.

2.4 Rapport et code

Vos solutions, pour la reconnaissances isolée et éventuellement la reconnaissance continue, devront être fournies sous la forme d'une fonction matlab que nous pourrons tester nous-mêmes sur de nouveaux échantillons sonores (le format des fonctions à fournir sera précisé

2. `https://www.cs.ubc.ca/~murphyk/Software/HMM/hmm.html`
3. les modèles `hmm1` et `hmm2` de la première partie sont des modèles "left-right"

sur le site web du cours). Cela nous permettra à la cloture du projet de tester vos solutions sur nos propres échantillons afin d'en évaluer les performances. Sur la base de ces tests, nous établirons un classement des groupes (par exemple en fonction de la précision obtenue sur nos échantillons) qui sera rendu public lors de la présentation finale [4].

Pour cette seconde partie, il n'y a pas de structure imposée au niveau du rapport. Vous ne devez cependant pas vous focaliser uniquement sur votre solution finale mais nous exposez le plus clairement possible la démarche que vous avez entreprise pour parvenir à cette solution, en ce compris la manière dont vous avez traité les points mentionnés dans la section 2.1. Idéalement, on s'attend également à ce que vous fournissiez une évaluation quantitative des performances de votre solution finale (voir le dernier point de la section 2.1).

3 Références

Toutes les références, les données, et les codes relatifs au projet seront collectés sur la page Web des projets (`http://www.montefiore.ulg.ac.be/~lduchesne/stocha/`). Une foire aux questions sera également ajoutée sur cette page et complétée au fur et à mesure de l'avancement du projet. Veillez à consulter régulièrement cette page.

Références

[Mur12] Kevin P. Murphy. *Machine Learning : A probabilistic Perspective*, chapter 17 : Markov and hidden markov models. The MIT Press, 2012.

[Rab90] Lawrence R. Rabiner. Readings in speech recognition. chapter A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition, pages 267–296. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1990.

4. Notez qu'un bon résultat à ce classement ne garantira pas que vous aurez une bonne cote au projet et inversement, vous pourrez avoir une bonne cote au projet sans être bien classé.