AVIGNON UNIVERSITÉ

Rapport TP3 : Modèle de Markov Cachés

Fait par

Jordan FOUTÉ TAPÉ

1er mai 2023

Master d'informatique
parcours Intelligence artificielle

UE APPRENTISSAGE SUPERVISÉ
ECUE Modèles stochastiques

Responsable
Stéphane Huet

UFR
SCIENCES
TECHNOLOGIES
SANTÉ

AVIGNON UNIVERSITÉ
CERI

CENTRE
D'**ENSEIGNEMENT**
ET DE **RECHERCHE**
EN **INFORMATIQUE**
ceri.univ-avignon.fr

Sommaire

1 Introduction

Dans ce TP il est question de mettre en oeuvre les modèles de Markov cachés (HMM) et de les appliquer dans trois applications différentes.

2 Prédiction de la meilleure séquence d'états

Toute cette partie est pratique, elle consiste à compléter la fonction `Viterbi()` qui calcul la meilleure séquence d'états cachés pour une séquence d'observations donnée. Ensuite, tester le code sur les données data/weather/weather-test.csv en utilisant les paramètres de modèle data/weather/weather.true.model. Vérifer ensuite qu'on retrouve bien les prédictions données dans data/weather/weather-test.ref, ce qui est bien le cas.

3 Application en bio-informatique

Il s'agit ici d'appliquer le programme avec le code Viterbi développé plus haut sur le fichier data/genetics/CG-resgion.csv. En utilisant le script src/compErrors.py et les données de référence data/genetics/CG-region.ref, ensuite de donner la précision du HMM sur ces données. La figure suivante présente le résultat. Pour cela, en se servant de l'exemple précédent, nous avons défini a partir des données, le fichier **data/genetics/CG-region.true.model** et ensuite comparer les résultats avec la fonction fournie.

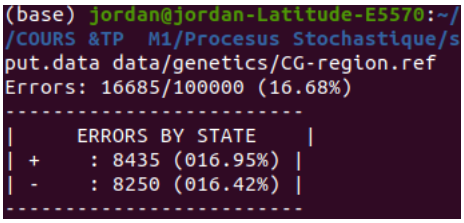

Figure 1. Précision du HMM

4 Apprentissage par maximum de vraisemblance

Dans cette partie, la première question a déjà faite et la réponse a été fournie par le professeur en ressource avec ce TP. La suite consiste a compléter la fonction **train-MLE()** calculant les paramètres du HMM à partir des observations et des états des données d'apprentissage. Nous avons utilisé cette fonction pour entraîner un HMM sur les données météo data/weather/weather500seq.csv et le résultat des paramètres se trouve dans le fichier weather.true.model.

Le taux de classification est présenté sur la figure suivante :

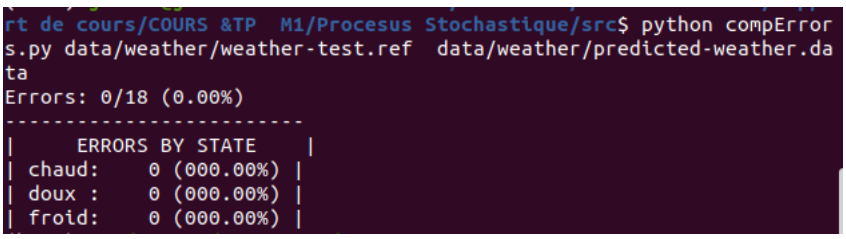

Figure 2. Taux de classification sur les données : Weather

5 Application en traitement automatique des langues

Explication sur comment le programme traite les mots inconnus Dans la fonction **read-vocab**, les mots inconnus sont identifiés en parcourant le fichier de vocabulaire ligne par ligne. Les mots et leurs occurrences sont stockés dans un dictionnaire **count-obs**. Ensuite, les mots qui ont moins d'occurrences que le seuil défini par la constante `UNK-THRESHOLD` sont associés à la clé "UNK" dans le dictionnaire count-obs. Cela signifie que ces mots sont considérés comme inconnus et seront remplacés par **"UNK"** lors de la conversion des mots en indices. Donc, les mots inconnus sont identifiés en vérifiant si les mots sont présents dans le dictionnaire de vocabulaire `model->dic-obs` pendant l'entraînement, et directement à partir du fichier de séquences d'observations pendant le test. Ils sont remplacés par "UNK" dans les séquences d'observations, et sont ajoutés à une file d'attente pour être traités ultérieurement.

Étiquetons avec l'algorithme de Viterbi chacun des corpus de test *test.word. L'algorithme de viterbi a été appliquée sur chaque fichier de test, et chaque résultat (fichier étiqueté) se trouve dans le dossier du corpus correspondant avec le préfixe **output***. Le tableau suivant résume l'erreur total sur chaque corpus en se servant des fichiers de références.

Corpus de test	**E**rreur	**P**ourcentage
fr-test.word	6602/81569	8.08
UD-French-FQB	2046/24135	8.48
UD-French-GSD	679/10019	6.78
UD-French-ParTUT	168/2603	6.45
UD-French-PUD	2200/24735	8.89
UD-French-Sequoia	622/10048	6.19
UD-French-Spoken	887/10029	8.84

Table 1. tableau synthétique donnant le pourcentage total d'étiquettes erronées

D'après les captures d'écrans se trouvant a la racine du dossier de Tp, correspondant aux résultats des test ; on constate que les étiquettes les plus difficiles à prédire sont **les noms (NOUN)**, **les adjectifs (ADJ)** et **les verbes (VERB)**.

6 Calcul de scores de confiance

Nous souhaitons ajouter ici une nouvelle fonctionnalité qui indique la confiance qu'a le modèle dans la prédiction de chaque étiquette.

Cette partie est purement pratique, les fonctions demandées ont été complété,le fichier fr-test.word a été étiqueté, et le fichier source C++ est joint avec ce rapport.