Modèle génératif et modèle discriminant pour l'étiquetage morpho-syntaxique

première partie

à rendre pour le 13 octobre 2015

L'étiquetage morpho-syntaxique consiste à associer chaque mot d'une séquence sa catégorie correcte (verbe, nom, adjectif, adverbe ...). La complexité du problème provient du fait que certains mots peuvent avoir plusieurs catégories selon leur position dans la phrase (... **la** maison ..., ... il **la** donne ..., ... le **la** est une note de musique ...).

L'objectif de ce projet est d'implémenter un étiqueteur morpho-syntaxique à l'aide d'un HMM. Dans sa version la plus simple, le HMM possède autant d'états qu'il existe de catégories différentes. Les observables sont constitués par les mots.

L'étiquetage proprement dit est réalisé à l'aide de l'algorithme de Viterbi, qui prend en entrée une séquence de mots $M = m_1 \dots m_n$ et calcule la séquences de catégories $\hat{C} = \hat{c}_1 \dots \hat{c}_n$ telle que :

$$\hat{C} = \arg \max_{C \in \mathcal{T}^n} S(C, M)$$

où $S(C, M)$ est le score associé à l'étiquetage de la séquence de mots M par la séquence de catégories C et $\mathcal{T} = \{t^1, \dots, t^k\}$ est l'ensemble des catégories.
Ce score se décompose de la façon suivante :

$$S(C, M) = \pi(c_1) + \sum_{i=1}^{n-1} T(c_i, c_{i+1}) + \sum_{i=1}^{n} E(c_i, m_i)$$

où π sont les scores initiaux ($\pi(t^i)$ est le score associé au fait que t^i soit la catégorie du premier mot de la phrase), T sont les scores de transition ($T(t^i, t^j)$ est le score associé au fait que la catégorie t^j suive directement la catégorie t^i) et E sont les scores d'émission ($T(t^i, m)$ est le score associé au fait que le mot m soit associé à la catégorie t^i).
Pour estimer les paramètres du HMM, on dispose de données complètes : une collection de phrases dont chaque mot a été associé à sa catégorie correcte.

1 Deux modèles

On testera deux fonctions de score, correspondant à deux types de modèles.

1.1 Modèle génératif

Le premier modèle permet de calculer, pour toute séquence de mots M et toute séquence de catégories C la probabilité jointe $P(C, M)$. C'est cette probabilité qui constitue le score $S(C, M)$ La séquence $\hat{C}$ calculée par l'algorithme de Viterbi est donc la séquence qui maximise cette probabilité.
Les paramètres du HMM sont :

— des probabilités initiales $\pi(t^i) = \log P(X_1 = t^i)$,
— des probabilités de transition $T(t^i, t^j) = \log P(X_t = t^j | X_{t-1} = t^i)$ et
— des probabilités d'émission $E(t^i, m) = \log P(O_t = m | X_t = t^i)$.

Ces probabilités peuvent etre estimées directement par les fréquences relatives dans le corpus d'apprentissage :

— Les probabilités initiales :

$$\pi(t^i) = P(X_1 = t^i) \approx \frac{\mathcal{I}(t^i)}{\mathcal{N}}$$

où $\mathcal{I}(t^i)$ est le nombre de phrases commençant par la catégorie t^i et $\mathcal{N}$ est le nombre de phrases.

— Les probabilités de transition :

$$T(t^i, t^j) = p(X_t = t^j | X_{t-1} = t^i) \approx \frac{\mathcal{C}(t^i, t^j)}{\mathcal{C}(t^i)}$$

où $\mathcal{C}(t^i, t^j)$ est le nombre d'occurrences du bigramme $\langle t^i t^j \rangle$ et $\mathcal{C}(t^i)$ le nombre d'occurrences de la catégorie t^i.

— Les probabilités d'émission :

$$E(t^i, m) = P(O_t = m | X_t = t^i) \approx \frac{\mathcal{C}(m, t^i)}{\mathcal{C}(t^i)}$$

où $\mathcal{C}(m, t^i)$ est le nombre de fois que m a été étiqueté t^i et $\mathcal{C}(t^i)$ le nombre d'occurrences de la catégorie t^i.

1.2 Modèle discriminant

Le second modèle appartient à la famille des modèles discriminants. On ne cherche pas ici à retrouver la séquence de catégories qui maximise une probabilité, mais la séquence qui permet de minimiser le nombre d'erreurs.
La fonction de coût est construite à l'aide de l'algorithme du perceptron :

1. Un paramètre I est choisi qui définit le nombre d'itérations de l'algorithme
2. Tous les poids ($T(t^i, t^j)$,$E(t^i, m)$ et $\pi(t^i)$) sont initialisés à zéro.
3. Pour chaque itération et pour chaque phrase $m_1 \dots m_n$ du corpus d'apprentissage ayant pour séquence d'étiquettes $c_1 \dots c_n$, la meilleure séquence de catégories, notée $\hat{c}_1 \dots \hat{c}_n$ est construite à l'aide de l'algorithme de Viterbi. Les poids sont alors mis à jour de la manière suivante :
   — $T(t^i, t^j) = T(t^i, t^j) + \sum_{k=1}^{n-1} \phi_{T(t^i,t^j)}(k, C) - \sum_{k=1}^{n-1} \phi_{T(t^i,t^j)}(k, \hat{C})$
   — $E(t^i, m) = E(t^i, m) + \sum_{k=1}^{n} \phi_{E(t^i,m)}(k, C, M) - \sum_{k=1}^{n} \phi_{E(t^i,m)}(k, \hat{C}, M)$
   — $\pi(t^i) = \pi(t^i) + \phi_{\pi(t^i)}(C) - \phi_{\pi(t^i)}(\hat{C})$

où les fonctions caractéristiques $\phi_{T(t^i,t^j)}(k,C)$, $\phi_{E(t^i,m)}(k,C,M)$ et $\phi_{\pi(t^i)}(C)$ sont définies de la façon suivante :

— $\phi_{T(t^i,t^j)}(k,C) = \begin{cases} 1 & si\ c_k = t^i\ et\ c_{k+1} = t^j \\ 0 & sinon \end{cases}$

— $\phi_{E(t^i,m)}(k,C,M) = \begin{cases} 1 & si\ c_k = t^i\ et\ m_k = m \\ 0 & sinon \end{cases}$

— $\phi_{\pi(t^i)}(C) = \begin{cases} 1 & si\ c_1 = t^i \\ 0 & sinon \end{cases}$

La règle de mise à jour des poids a pour effet d'augmenter les poids des caractéristiques absentes dans la séquence $\hat{C}$ et diminuer le poids des caractéristiques erronées de la séquence $\hat{C}$.

2 Implémentation

Vous pourrez utiliser l'implémentation de HMM qui se trouve dans le fichier `hmm.c`. Elle permet de représenter un HMM sous la forme de trois tableaux, un tableaux pour les paramètres initiaux un tableau pour les paramètres de transitions et un tableau pour les paramètres d'émission.

Dans l'implémentation proposée, un HMM peut être stocké dans un fichier texte au format suivant [1] :

```
#nb etats
2
#nb observables
2
#parametres initiaux
0.362146 # I(0)
0.637854 # I(1)
#parametres de transition
0.479827 # T(0,0)
0.520173 # T(0,1)
0.819147 # T(1,0)
0.180853 # T(1,1)
#parametres d'emission
0.257264 # E(0,0)
0.742736 # E(0,1)
0.742653 # E(1,0)
0.257347 # E(1,1)
```

L'implémentation qui vous est donnée ne manipule que des entiers (les états et les observables sont identifiés par des entiers successifs à partir de zéro). C'est la raison pour laquelle les données sont encodées .

1. Les caractères qui suivent un dièse sont des commentaires, ils peuvent être omis.

3 Données

Les données d'apprentissage se trouvent dans le fichier `train`. Dans le fichier `test`, on trouvera d'autres phrases étiquetées qui serviront à calculer les performances de l'étiqueteur. Les phrases de test ne seront pas utilisées pour estimer les paramètres, elles sont sensées représenter des nouvelles phrases. Les fichiers fournis sont encodés, toute catégorie et tout mot est représenté par un entier.

La mesure utilisée pour l'évaluation de l'étiqueteur est la précision, qui mesure tout simplement la proportion de mots du corpus de test auxquelles l'étiqueteur a attribué la bonne catégorie. Vous trouverez le programme d'évaluation `eval.pl` sur la page du cours.

4 Ce qui vous est demandé

1. Programmer l'algorithme de Viterbi.
2. Estimer les paramètres T, E et π par fréquence relative sur le corpus d'apprentissage.
3. Estimer les paramètres T, E et π à l'aide de la règle de mise à jour des poids du perceptron.
4. Calculer les performances des différents modèles sur le corpus de test en faisant varier la taille du corpus d'apprentissage.

Bon courage !