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

Rendu première partie : 18 octobre 2016
Deadline : 15 novembre 2016

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. Cet algorithme qui prend en entrée une séquence de mots $O = O_1 \dots O_T$ et calcule la séquences de catégories $\hat{Q} = \hat{q}_1 \dots \hat{q}_T$ telle que :

$$\hat{Q} = \arg\max_{Q \in S^T} P(O, Q) = \arg\max_{Q \in S^T} P(O|Q)P(Q)$$

où $P(O, Q)$ est le score (souvent dérivé de la probabilité) associé à l'étiquetage de la séquence de mots O par la séquence de catégories Q et $S = \{S_1, \dots, S_N\}$ est l'ensemble de catégories possibles.

Ce score se décompose de la façon suivante :

$$P(O, Q) = \pi_{q_1} + \sum_{t=2}^{T} a_{q_{t-1} q_t} + \sum_{t=2}^{T} b_{q_t}(O_t)$$

où π sont les scores initiaux (π_i est le score associé au fait que S_i soit la catégorie du premier mot de la phrase), a sont les scores de transition (a_{ij} est le score associé au fait que la catégorie $q_t = S_j$ suive directement la catégorie $q_{t-1} = S_i$) et $b(\cdot)$ sont les scores d'émission ($b_t(j)$ est le score associé au fait que le mot q_t soit associé à la catégorie S_j).

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 O et toute séquence de catégories Q la probabilité jointe $P(O, Q)$. C'est cette probabilité qui constitue le score $P(O, Q)$ La séquence $\hat{Q}$ calculée par l'algorithme de Viterbi est donc la séquence qui maximise cette probabilité. [1]

Les paramètres du HMM sont :

— des probabilités initiales $\pi_i = -\log P(q_1 = S_i)$,

— des probabilités de transition $a_{ij} = -\log P(q_t = S_j | q_{t-1} = S_i)$ et

— des probabilités d'émission $b_j(k) = -\log P(O_t = V_k | q_t = S_j)$.

Ces probabilités peuvent etre estimées directement par les fréquences relatives dans le corpus d'apprentissage en fonction des nombres d'occurrences $C(\cdot)$:

— Les probabilités initiales :

$$\pi_i = -\log P(q_1 = S_i) \approx -\log \frac{C_{ini}(S_i)}{\mathcal{N}}$$

où $C_{ini}(S_i)$ est le nombre de phrases commençant par la catégorie S_i et $\mathcal{N}$ est le nombre de phrases.

— Les probabilités de transition :

$$a_{ij} = -\log P(q_t = S_j | q_{t-1} = S_i) \approx -\log \frac{C(S_i, S_j)}{C(S_i)}$$

où $C(S_i, S_j)$ est le nombre d'occurrences du bigramme $\langle S_i S_j \rangle$ et $C(S_i)$ le nombre d'occurrences de la catégorie S_i.

— Les probabilités d'émission :

$$b_j(k) = -\log P(O_t = V_k | q_t = S_j) \approx -\log \frac{C(V_k, S_i)}{C(S_i)}$$

où $C(V_k, S_i)$ est le nombre de fois qu'un mot V_k a été étiqueté S_i et $C(S_i)$ le nombre d'occurrences de la catégorie S_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 (a_{ij},$b_i(k)$ et π_i) sont initialisés à zéro.
3. Pour chaque itération et pour chaque phrase $O = O_1 \dots O_T$ du corpus d'apprentissage ayant pour séquence d'étiquettes $Q = q_1 \dots q_T$, la meilleure séquence de catégories, notée $\hat{Q} = \hat{q}_1 \dots \hat{q}_n$ est construite à l'aide de l'algorithme de Viterbi.

   Si $Q \neq \hat{Q}$, alors poids sont alors mis à jour de la manière suivante :

1. Les scores ci-dessous sont définis en domaine $-\log P$. Il faut donc les *minimiser* dans l'algorithme de Viterbi.

— $a_{ij} = a_{ij} + \sum_{t=1}^{T-1} \phi_{a_{ij}}(t, Q) - \sum_{t=1}^{T-1} \phi_{a_{ij}}(t, \hat{Q})$
— $b_j(k) = b_j(k) + \sum_{t=1}^{T-1} \phi_{b_j(k)}(t, Q) - \sum_{t=1}^{T-1} \phi_{b_j(i)}(t, \hat{Q})$
— $\pi_i = \pi_i + \phi_{\pi_i}(Q) - \phi_{\pi_i}(\hat{Q})$

où les fonctions caractéristiques $\phi_{a_{ij}}(t, Q)$, $\phi_{b_j(k)}(t, Q)$ et $\phi_{\pi_i}(Q)$ sont définies de la façon suivante :

— $\phi_{a_{ij}}(t, Q) = \begin{cases} 1 & si\ q_t = S_i\ et\ q_{t+1} = S_j \\ 0 & sinon \end{cases}$
— $\phi_{b_j(k)}(t, Q) = \begin{cases} 1 & si\ q_t = S_i\ et\ O_t = V_k \\ 0 & sinon \end{cases}$
— $\phi_{\pi_i}(Q) = \begin{cases} 1 & si\ q_1 = S_i \\ 0 & sinon \end{cases}$

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

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 [2] :

```
#nb etats
2
#nb observables
2
#parametres initiaux
0.362146 # pi(0)
0.637854 # pi(1)
#parametres de transition
0.479827 # a(0,0)
0.520173 # a(0,1)
0.819147 # a(1,0)
0.180853 # a(1,1)
#parametres d'emission
0.257264 # b(0,0)
0.742736 # b(0,1)
0.742653 # b(1,0)
0.257347 # b(1,1)
```

L'implémentation qui vous est donnée ne manipule que des entiers pour représenter les états S_i et les observables V_k, qui sont identifiés et indexés par des entiers suc-

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

cessifs à partir de zéro. C'est la raison pour laquelle les données sont encodées .

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 $\lambda = (A, B, \pi)$ par fréquence relative sur le corpus d'apprentissage.
3. Estimer les paramètres $\lambda = (A, B, \pi)$ 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.

5 Extensions

Vous pouvez améliorer votre modèle de plusieurs façons :
— Prise en compte des mots inconnus avec lissage
— Utilisation de l'algorithme EM pour estimer les paramètres avec des données non annotées
— Ajustement de la mise à jour des poids (moyenne, taux d'apprentissage)
— Initialisations du modèle

6 Modalités d'évaluation

Le projet peut être fait en binôme et sera évalué sur deux aspects :

1. Un compte-rendu d'environ 10-15 pages qui décrit le modèle, les choix d'implémentation, les extensions et améliorations, ainsi que les performances selon la taille du corpus d'entraînement
2. Un exposé oral de ∼10min pour décrire les résultats obtenus, suivi de questions/discussion