Projet HMM

La communauté de l'info

Anthony GIRAUDO, Hichma KARI, Kilian MAC DONALD, Louise MARINHO

Nous étudierons dans ce rapport la démarche réalisée pour construire un HMM adapté à la reconnaissance de l'anglais à l'aide du fichier anglais2000 qui a été mis à la disposition de l'ensemble de la classe.

1 Étude du nombre d'états optimal

Nous avons commencé par nous intéresser au nombre d'états optimal que devait contenir notre HMM. Pour cela nous nous avons utilisé la structure de la fonction *xval* vue en cours pour étudier l'efficacité d'un HMM pour reconnaître l'anglais en fonction de son nombre d'états. Nous avons pour cela supposé que 100 itérations de BW1 pour entraîner un HMM suffiraient. Cette hypothèse se révélera vérifiée par la suite. Nous avons fixé le nombre d'initialisations aléatoires à une par appel de la fonction BW3 au risque d'avoir un certain bruit sur nos courbes : nous ne voulions que voir l'évolution générale de la log-vraisemblance avec le nombre d'états. Nous avons partitionné la liste des mots en 10.

Finalement, nous avons obtenu les résultats suivants.

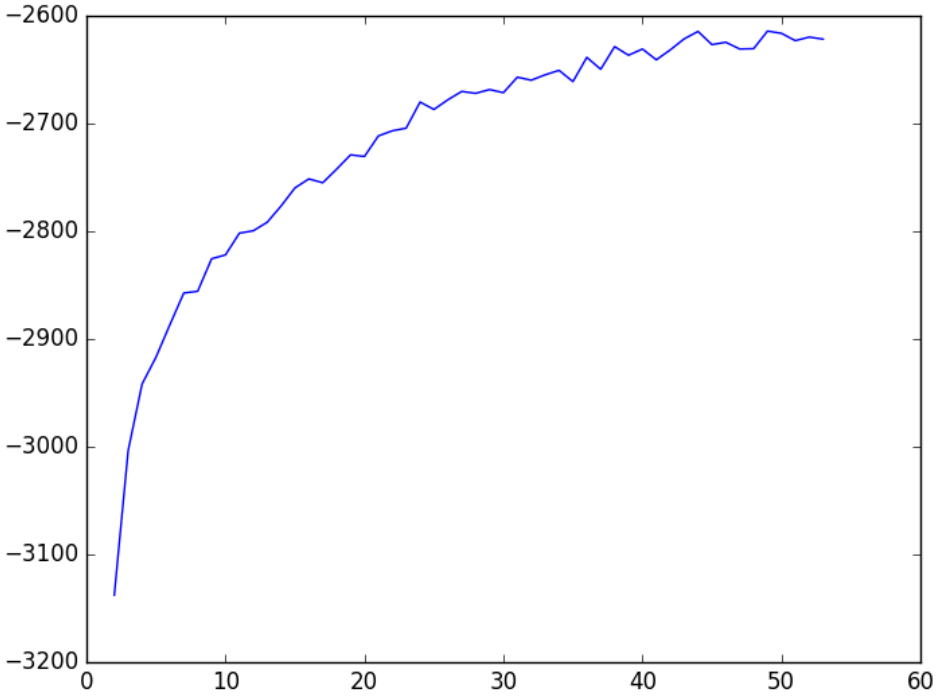

FIGURE 1 – Log-vraisemblance moyenne sur un échantillon ne faisant pas partie des exemples en fonction du nombre d'états du HMM. Les paramètres de xval utilisés sont nbFolds = 10, nbIter = 100 et nbInit = 1

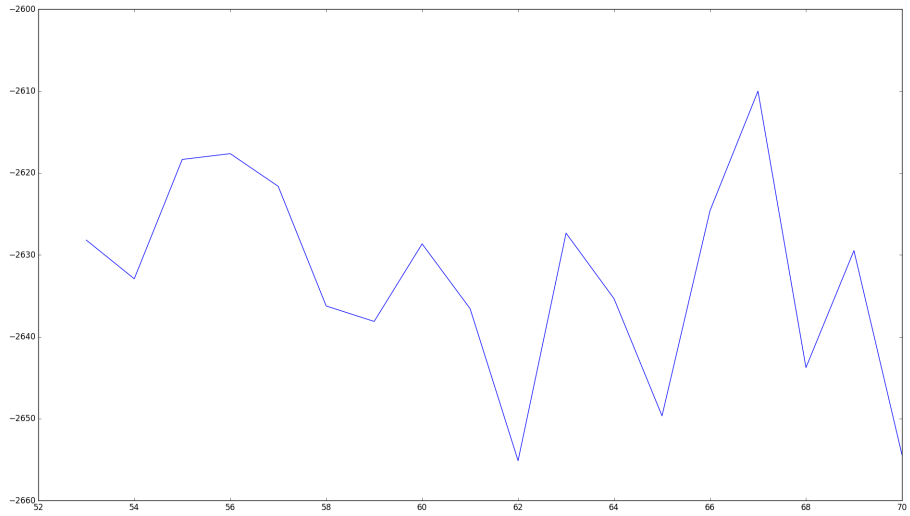

FIGURE 2 – Log-vraisemblance moyenne sur un échantillon ne faisant pas partie des exemples en fonction du nombre d'états du HMM. Les paramètres de xval utilisés sont nbFolds = 10, nbIter = 100 et nbInit = 1

Nous voyons que la log-vraisemblance ne semble plus augmenter à partir de 45 états environ. Nous travaillerons donc pour l'anglais avec des HMM possédant 45 états.

2 Étude de la variation de la log-vraisemblance en fonction du nombre d'itérations dans BW2

Nous avons cherché à déterminer le nombre d'itérations de BW1 nécessaire pour pouvoir considérer qu'un HMM est bien entraîné. Pour cela, nous avons tracé la variation de la log-vraisemblance en fonction du nombre d'itérations de BW1. Cette opération a été répétée plusieurs fois, donnant des résultats reproductibles. Pour nous aider à analyser les résultats nous avons pris l'initiative d'utiliser le module scipy.optimize pour faire correspondre à nos courbes l'opposée d'une exponentielle décroissante.

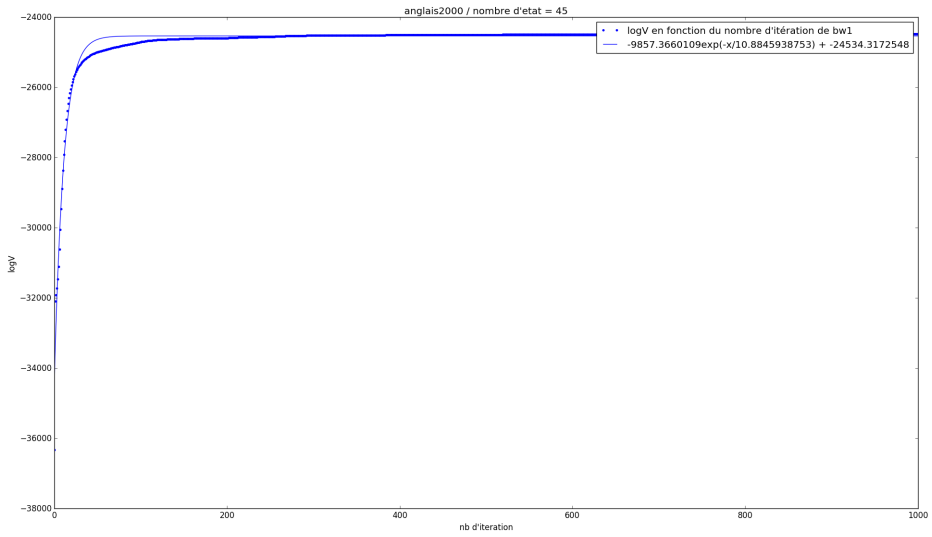

FIGURE 3 – Log-vraisemblance en fonction du nombre d'itérations de BW1 (HMM entraîné sur anglais2000)

La valeur qui divise $-x$ dans l'exponentielle est le nombre d'itérations caractéristiques, que nous noterons τ. Pour un nombre d'itérations de 3τ la log-vraisemblance aura atteint 95% de sa valeur à l'infini. Pour un nombre d'itérations de 5τ, la log-vraisemblance aura atteint 99.3% de sa valeur à l'infini. D'après le fitting, nous avons $\tau = 11$. Donc en faisant 55 itérations, nous aurons atteint 99.3% de la valeur maximale que nous pouvons atteindre. Cependant, nous voyons sur le graphique que le fitting n'est pas très bon à 55 itérations : les valeurs expérimentales sont plus faibles que la courbe de l'exponentielle. Il vaut donc mieux aller jusqu'à 200 voir 400 itérations où les deux courbes se rejoignent et où l'on est alors sûr d'avoir approché la limite à plus de 99.3%. Nous avons par la suite entraîné plusieurs HMM avec plus ou moins d'itérations suivant les cas et le temps que nous souhaitions y consacrer, mais alors en connaissance précise des approximations que nous faisions, grâce à ces études.

3 Étude de la variation de la log-vraisemblance optimale après BW2 en fonction de l'initialisation

Pour voir l'influence de l'initialisation aléatoire, nous avons étudié la variation de la log-vraisemblance atteinte après entraînement d'HMM initialisés aléatoirement.

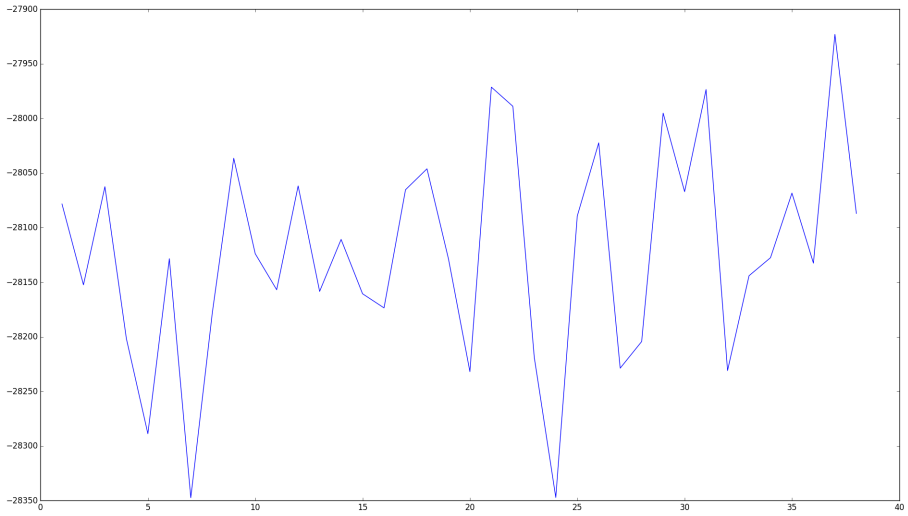

FIGURE 4 – Log-vraisemblance en fonction de l'initialisation aléatoire (numérotées de 1 à 40), après entraînement

Nous observons des variations non négligeables. Nous remarquons qu'avec 20 initialisations aléatoires différentes pour BW3, nous devrions obtenir au moins un HMM ayant une log-vraisemblance proche de la meilleure possible.

4 Prédire la langue d'un mot

Nous avons récupéré la liste des mots les plus fréquents en :
— Anglais
— Allemand
— Espagnol
— Néerlandais
— Suédois
— Elfique

Nous avons entraînés des HMM correspondants. Nous sommes capables de prévoir la langue d'un mot en calculant sa log-vraisemblance avec les HMM correspondant à chacune des langues étudiées. Ainsi, le mot devrait appartenir à la langue à laquelle correspond le HMM qui donnera la plus grande log-vraisemblance.

Résultats

Nous avons pu créer des mots en anglais tel que :
— prostion
— stl
— sader
— anervall
— kidentad
— sextions

— ablis
— modinar
— reakse
— comedon
— owalsh
— nelar
— sisty
— mewomsos
— soptrese
— sossomal
— sedstse
— berdery
— stor
— prentsbu

Ces mots ne semblent pas tous très anglais, mais parfois le HMM invente un mot déjà existant en anglais comme forest, pain ou brave alors que ces mots ne sont pas dans le fichier anglais2000. Nous avons réussi à atteindre une log-vraisemblance de -24495 en entraînant un HMM sur anglais2000.

Et en suédois :

— tarem
— gva
— gorr
— ostaj
— bas
— ittodaf
— seronat
— kalllals
— dar
— nangen
— gojjett
— honsken
— unt
— durgor
— antedara
— sju
— for
— nig
— ven
— tarftara

Pour la reconnaissance de langue, le modèle se trompe rarement pour les mots courants. Mais certains mots posent parfois problème comme understand qui est détecté comme étant allemand.