TD : K-NN appliqué à la classification de textes

Suite du premier TD KNN

Il s'agit de prolonger le TD KNN de deux manières :

1.1 Utilisation de poids TF.IDF

Tout d'abord vous tenterez d'améliorer les résultats en utilisant d'autres valeurs pour les vecteurs représentant les documents :

- au lieu d'utiliser le nombre d'occurrences du mot dans le document (normalisé par le nb total d'occurrences ds le document)
- vous utiliserez un score dit « TF.IDF », classiquement utilisé en recherche d'information
  - TF.IDF signifie « term frequency inverse document frequency »
  - et est défini pour un couple mot / document :
  - **TF.IDF(m, d) = TF(m,d)*IDF(m)**
  - ici TF vaut les valeurs fournies ds le TD1 :
    - **TF(mot m, document d)** = nb d'occ de m dans d / nb d'occ total dans d)
    - Rem : Fréquence signifie ici « proportion d'occurrences ».
  - et IDF vaut « le log de l'inverse de la fréquence en documents » :
    - **IDF(mot m)=** log(nb total de documents / nb de documents contenant m)

Pour simplifier, vous calculerez une fois pour toutes les scores IDF des mots présents dans le corpus reuters.modapte.train.sgm (programme in=fichier au format .sgm, out=fichier au format de votre choix, qui associe l'IDF à chaque mot du vocabulaire du fichier d'entrée.
Utilisez ensuite ce fichier pour modifier les vecteurs de documents, et testez si cela améliore les résultats (sur medium). Vous pouvez également tester un seuil de filtrage des mots : les mots ayant un IDF trop bas pourraient être ignorés.

1.2 Script de réglage des hyper-paramètres

Ecrivez un script (sh) qui lance l'évaluation de votre KNN avec différentes valeurs de k, et avec ou sans pondération par l'inverse.

Le but est de chercher par vous-mêmes la syntaxe d'une boucle (cherchez « bash boucle for » par exemple).

Données : sont fournis

- reuters.modapte.train.sgm : pour le calcul des IDF(mot)
- reuteurs.train.examples et reuters.test.examples : pour tester votre KNN sur plus de données (mais c'est long...)