M1–Master d'informatique

PJE – Analyse de comportement avec Twitter

Classification bayésienne (2)

Le but de cette séance est d'améliorer l'algorithme de *classification bayésienne* permettant de classifier un *nouveau tweet* comme `positif`, `négatif` ou `neutre`.

1 Représentation par présence

Lors de la séance précédente, nous avons défini un attribut comme un mot du "vocabulaire". On lui affecte une valeur booléenne qui indique la présence d'un mot particulier dans le tweet (ensemble de mots). Ainsi, on ne se préoccupe pas de la syntaxe du texte, c'est-à-dire ni de l'ordre des mots, ni de leur organisation dans le texte, ni de leur nombre d'occurrences.

Pour rappel, la probabilité pour un tweet t d'être `neutre` est estimée comme suit.

$$P(\texttt{neutral}|t) = \prod_{m \in t} P(m|\texttt{neutral}) \cdot P(\texttt{neutral}) \tag{1}$$

Il en va de même pour les classes `positif` et `négatif`. Nous retiendrons la classe pour laquelle la probabilité est la plus grande pour l'instance considérée.

2 Représentation par fréquence

Avec la représentation pour présence, on ne se préoccupe pas du nombre d'occurrences des mots. Nous allons étendre cette représentation en prenant en compte le nombre d'occurrences d'un mot du texte (sac de mots).

Ainsi, nous allons modifier l'équation (3) du TD1 comme suit :

$$P(t|c) = \prod_{m \in t} P(m|c)^{n_m} \tag{2}$$

où $P(m|c)$ est toujours la probabilité d'occurrence du mot m dans un texte de la classe c, et où n_m est le nombre d'occurrences du mot m dans le tweet t. Le reste du raisonnement est inchangé.

Question 2.1 : Étendez votre algorithme de classification bayésienne basé sur la *présence* des mots afin de prendre en compte la *fréquence* des mots dans un tweet.

Nous analyserons par la suite l'impact de la représentation par présence et de la représentation par fréquence sur les performances du classifieur.

3 Quels mots (ou ensembles de mots) ont de l'importance ?

Pour éviter certains calculs sans affecter les performances du classifieur, on peut encore simplifier le modèle en n'incluant pas dans le "vocabulaire" les mots tels que les articles ou les pronoms. En effet, ces mots n'ajoutent rien au sens du texte. Cela permet de réduire la taille de l'ensemble de mots considérés sans altérer les performances.
Pour faire simple, on peut se contenter d'ignorer les mots de moins de 3 lettres.

Question 3.1 : Modifiez votre algorithme de classification bayésienne afin de ne pas prendre en compte les mots de 3 lettres et moins.

Par ailleurs, nous remarquons que certaines combinaisons de mots (suite de mots) peuvent avoir de l'importance. Ainsi, afin de prendre en compte l'ordre des mots, plutôt que de considérer uniquement des mots simples et uniques, nous allons considéré des ensembles de mots contigus. Ceci peut être vu comme une manière de réintroduire une forme de dépendances entre les attributs (les mots). Par la suite, nous définissons un mot simple comme un *uni-gramme*. On peut également considérer des *n-grammes*, tels que les bi-grammes (deux mots consécutifs), tri-grammes (trois mots consécutifs), etc.
Pour illustrer l'hypothèse d'une telle approche, nous espérons que certains n-grammes soient caractéristiques d'un jugement favorable, défavorable, ou neutre. Par exemple, nous pouvons imaginer des n-grammes caractéristiques tels que : "raz de marrée" ou "chef d'oeuvre".

Question 3.2 : Jusqu'à présent, nous avons considérer les uni-grammes uniquement. Étendez votre algorithme de classification bayésienne afin de considérer des bi-grammes, ou la combinaison des uni-grammes + bi-grammes.

Nous analyserons par la suite l'impact du choix entre uni-grammes, bi-grammes, et uni-grammes + bi-grammes sur les performances du classifieur.