Le Naïve Bayes

Un autre modèle de classification

Rafik LACHAAL

Sommaire

- Le Naïve Bayésien
- Représentation vectorielle des textes (classification de texte).
- Pondération TF-IDF : Fréquence des Terme et Fréquence Inverse de Documents

Le Naïve Bayes

Aussi appelé Gaussien Naïve Bayes ou la Classification naïve

- Algorithme de classification
- Repose sur l'hypothèse naïve d'indépendance totale des variables.
  - On dit que l'évènement A et indépendant de B si : $P(A|B) = P(A)$
- Algorithme polyvalent

Théorème de Bayes :

$$P(A|B) = \frac{P(B|A).P(A)}{P(B)}$$

On peut réécrire $P(B)$ comme suit :

$$P(B) = P(A \cap B) + P(\bar{A} \cap B) \ = P(B|A).P(A) + P(B|\bar{A}).P(\bar{A})$$

On peut donc réécrire le théorème de Bayes :

$$P(A|B) = \frac{P(B|A).P(A)}{P(B|A).P(A) + P(B|\bar{A}).P(\bar{A})}$$

Le Naïve Bayes

Exemple du filtre anti-spam :

Quel est la probabilité qu'un email sois un spam s'il contient le mot « gratuit ».

$$P(S|"gratuit") = \frac{P("gratuit"|S).P(S)}{P("gratuit"|S).P(S) + P("gratuit"|\bar{S}).P(\bar{S})}$$

La probabilité d'avoir le mot gratuit dans un spam

La probabilité d'avoir le mot gratuit dans un non spam

$P("gratuit"|S).P(S)$
$P("gratuit"|\bar{S}).P(\bar{S})$

Ces probabilités seront calculé par python à partir des fréquences d'apparition dans un jeux d'entrainement

Le Naïve Bayes

Exemple du filtre anti-spam :

$$P(S) = 0.86$$
$$P(\bar{S}) = 0.14$$
$$P("gratuit"|S).P(S) = 0{,}96$$
$$P("gratuit"|\bar{S}).P(\bar{S}) = 0{,}02$$

$$P(S|"gratuit") = \frac{(0{,}96).(0{,}86)}{(0{,}96).(0{,}86) + (0{,}02).(0{,}14)} = 0{,}9966$$

Il donc donc une forte probabilité que cet email soit un spam.

Le Naïve Bayes

On peut généraliser l'exemple précédent à une séquence de mot :

$$P(S|mot1, mot2, \dots, motn) = \frac{P(mot1, mot2, \dots, motn|S).P(S)}{P(mot1, mot2, \dots, motn|S).P(S) + P(mot1, mot2, \dots, motn|\bar{S}).P(\bar{S})}$$

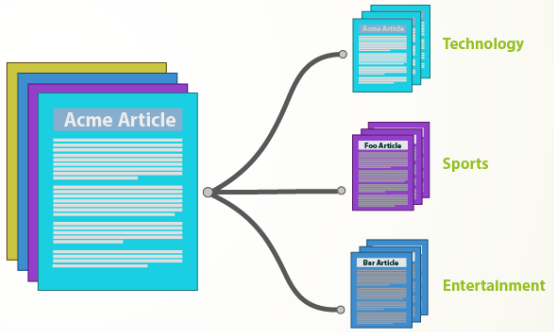
Avec :

$$P(mot1, mot2, \dots, motn|S) = P(mot1|S)P(mot2|S) \dots P(motn|S)$$

Représentation vectorielle des textes

Problématique :

Je dispose d'un ensemble de document (des articles de presse, ou des emails par exemple) et je souhaite pouvoir déterminer de manière automatisé le thème qu'il traite ou à quel catégorie de document appartient (spam, non spam).



On comprends assez bien que se problème est un problème d'apprentissage supervisé et plus particulièrement de classification.

Le hic, c'est qu'on ne sais traiter les problèmes de classifications qu'avec des données numériques et des variables catégorielles et nom un bloque de texte.

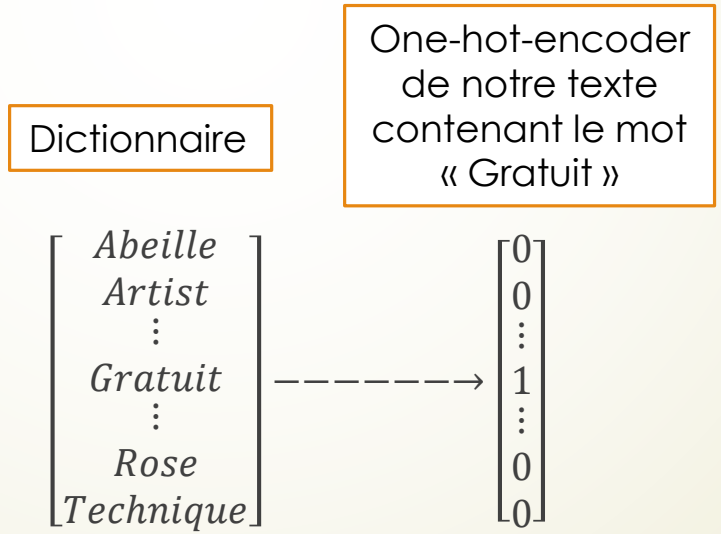
Représentation vectorielle des textes

Pour vectoriser nos documents il nous faudra tout d'abord **un dictionnaire contenant tout les mots possibles que l'on pourrait avoir à traiter**. Idéalement ce dictionnaire sera construit à partir de notre jeux d'entrainement.

On aurait donc une liste **ordonnée** constituée de tout les mot uniques apparaissant dans notre jeu d'entraînement.

Pour chaque mot dans le texte on peut donc construire à partir de ce dictionnaire ce que l'on appel le **one-hot-vector**. Ce vecteur contiendra :

- un **1** à la ième position si le texte contient le ième mot du dictionnaire.
- Sinon il contiendra **0**.



Représentation vectorielle des textes

En sommant l'ensemble des one-hot-vector on obtient un vecteur qui nous donne le nombre d'apparition des mots dans le texte. Et c'est avec ce type de vecteur que je peux déterminer des probabilités.

$$\begin{bmatrix}0\\0\\\vdots\\1\\\vdots\\0\\0\end{bmatrix}+\begin{bmatrix}0\\1\\\vdots\\0\\\vdots\\0\\0\end{bmatrix}+\cdots+\begin{bmatrix}0\\0\\\vdots\\0\\\vdots\\1\\0\end{bmatrix}=\begin{bmatrix}10\\4\\\vdots\\9\\\vdots\\21\\7\end{bmatrix}\longleftrightarrow\begin{bmatrix}Abeille\\Artist\\\vdots\\Gratuit\\\vdots\\Rose\\Technique\end{bmatrix}$$

Représentation vectorielle des textes

Problème :

- Lorsque j'ai un long document mon vecteur sera bien plus gros avec de forte valeur et plus difficile à gérer. On aura un coût de calcul plus élevé.
- Certain mot qui apparaissent moins souvent parce qu'il apparaissent dans un texte plus court mais qui peuvent être considéré comme discriminant se voit attribuer de petite valeur.

Solution :

- On divise par le nombre totale de mots pour obtenir la Fréquence des Termes, ou **Term Frequncy** en anglais (**TF**).
- On multiplie par la fréquence inverse des document, **Inverse Document Frequency** en anglais (**IDF**) :

$$idf = log \frac{Nombre\ total\ de\ textes}{Le\ nombre\ de\ texte\ ou\ le\ mot\ apparait}$$

Cette valeur permet de donner un poids plus important aux termes les moins fréquents, considérés comme plus discriminants.

Au finale pour chaque mot on calculera le produit : $\boldsymbol{tf.idf}$.On peut voire ça comme une forme de normalisation.