Classification de Documents Texte

Pr T.RACHAD

QFM – 2024/2025

Plan

1. Notion de classification
2. Types de classification
3. Applications de la classification
4. Processus de classification
5. Modèles de classification

Notion de Classification

- La classification (catégorisation) est le processus qui permet d'attribuer à chaque document dans un corpus la classe (ou les classes) le(s) plus adéquat(s), sachant que la liste des classes possibles est prédéfinie.

- Formellement, la tâche de la classification est une approximation d'une fonction inconnue $F : D \times C \rightarrow \{0, 1\}$, qui pour chaque couple $(d, c) \in D \times C$ retournera 1 si le document ***d*** appartient à la classe ***c***, sinon elle retournera 0.

- La fonction $M : D \times C \rightarrow \{0, 1\}$ qui est une fonction approchée de la fonction ***F*** est appelée un classificateur ou modèle de classification qui va apprendre la classification des nouveaux documents à partir de ceux qui sont déjà classifiés.

Types de Classification

- Deux approches de classification:
  - **Mono-label** :chaque document appartient exactement à une seule classe. En plus, si le nombre de classes possibles est exactement 2 alors on parlera de la **classification binaire**.
  - **Multi-label**: Un document peut être attribué à plusieurs classes simultanément. Ce genre de classification peut être réalisé en se basant sur n classificateurs binaires (un classificateur binaire par classe) tel que chaque classificateur décide sur l'affectation d'un document à une classe indépendamment des autres.

Applications de la Classification

- Indexation de texte.
- Tri de documents texte.
  - Analyse de topics
  - Analyse de sentiments
  - Analyse d'intentions
  - Analyse d'opinions
- Filtrage de documents texte.
  - Recommandation
- …

Indexation de Documents

- Extraire à partir d'un document texte **les termes clés** qui décrivent mieux son contenu et qui appartiennent à un **vocabulaire contrôlé**.

- L'indexation de texte est utilisée principalement dans la recherche d'information (**information retrival**) pour extraire à partir d'un corpus de documents texte ceux qui répondent mieux à la requête de l'utilisateur qui est même fondée sur la base des termes du vocabulaire.

- Les termes clés peuvent être vus comme des classes à attribuer aux différents documents d'un corpus.

Tri de Documents

- Classifier une collection de documents en fonction de plusieurs topics. chaque document appartient exactement à un topic.

- Exemples:
  - Classification des articles de presse (Sport, Economie, Politique, Social…)
  - Classification des emails (Professionnel, Publicité, Evènement…)

Filtrage de Documents

- Il s'agit d'un tri de documents de texte selon un critère pour lequel chaque document est considéré comme **pertinent ou non pertinent**.

- Exemples :
  - Détection des emails spam
  - Système de recommandation (publicité, e-Commerce,…)
  - Traitement des feedbacks des clients
  - …

Processus de Classification de Documents

1. Préparation du dataset.
2. Appliquer les différents prétraitements nécessaires.
3. Extraction et sélection des éléments clés pertinents.
4. Choisir un modèle de classification.
5. Evaluation du modèle obtenu.

Préparation du Dataset

1. Collecte de données
2. Déterminer la liste des classes.
3. Fournir un dataset d'apprentissage pour chaque classe.
4. Résoudre les petits datasets. ➔ Data Augmentation
5. Résoudre les Dataset déséquilibrés ➔ SMOTE
6. Eliminer les outliers. ➔ (distance-based, density-based, Subspace methods)
7. Eliminer le bruit qui peut causer un surapprentissage ➔ Cross validation

Prétraitements

- Un document texte ne peut pas être traité dans son format brute.

- Nécessité d'avoir un format plus structuré et plus représentatif du contenu d'un document.

- Les représentations les plus courantes sont:
  - Tokenized Document (segmentation, racinisation, lemmatisation)
  - bag-of-words (bow)
  - TF-IDF (Term Frequency–Inverse Document Frequency)
  - **Word2vec (https://arxiv.org/pdf/1301.3781.pdf)**
  - **Doc2Vec (https://cs.stanford.edu/~quocle/paragraph_vector.pdf)**
  - **GloVe (https://nlp.stanford.edu/projects/glove/)**
  - **BERT (https://github.com/google-research/bert)**

Sélection des éléments clés

- Les dimensions des vecteurs obtenus sont énormes. Cela aura un impact direct sur la performance du modèle (overfitting, accuracy, training Time).

- Il est nécessaire de réduire la taille de ces vecteurs en éliminant les redondances et en sélectionnant juste les features qui sont les plus significatifs ou en combinant les features similaires ou appartenant à la même famille .

  - Ça peut commencer par supprimer les stop words, connecteurs grammaticaux, mots courants

  - Calculer la pertinence (fréquence) de chaque feature et garder les 10% les plus pertinent(Univariate Selection, Feature Importance, correclation Heatmap).

  - Appliquer une LSA, PCA pour réduire la dimensionnalité.

  - ….

Choisir un Modèle de Classification

- Modèles traditionnels (NB, BLR, DT,…)
- Modèles à base des réseaux de neurones.
- Modèles à base des réseaux de neurones convolutif.
- Modèles à base des réseaux de nuerons récurrents.
- Modèles à base de Transfer Learning.

Modèles traditionnels : Naïve Bayes

- Calcul la probabilité qu'un document **d** appartient à une classe **c** en appliquant le théorème de Bayes:

$$P(c|d) = P(d|c) * P(c) / P(d)$$

  - $P(d)$: la probabilité marginale de d.
  - $P(c)$: la probabilité marginale de c.
  - $P(d \mid c) =) = \prod P(w_i \mid c)$. avec **d** = ($w1$, $w2$, . . .).

Modèles traditionnels : BLR

- Pour une classification binaire, l'application d'une régression logistique bayésienne est souhaitable:

$$P(c|\mathrm{d})=\varphi(\beta,d)=\varphi(\sum \beta_i w_i)$$

  - c = ±1 pour remplacer {0, 1}
  - **d** = (w_1 , w_2 , . . .) modèle représentatif du document d.
  - β = (β1 , β2 , . . .) est le vecteur des coefficients de régression.
  - ϕ la fonction logistique.

Modèles traditionnels : Decision Tree

- Un arbre de décision est une structure arborescente dans laquelle les nœuds internes sont étiquetés par les éléments clés du modèle représentatif des documents, les arcs sont étiquetés par des instruction conditionnelles sur les éléments clés et les feuilles sont étiquetées par les classes.

- La classification d'un nouveau document se fait par le parcours de l'arbre de décision depuis la racine et en passant par les branches qui vérifient les conditions jusqu'à qu'on arrive à une feuille dont l'étiquette représentera la classe du document.

- Techniques d'implémentation: ID3, C4.5,CART.

Modèles traditionnels : Decision Rule

- Permet d'induire les règles qui permettent de classifier correctement les documents à partir d'un dataset d'apprentissage et de l'ensemble de toutes les règles possibles.

- On démarre par la construction d'un classificateur admettant toutes les règles pour tous les documents du dataset d'apprentissage $w_1 \wedge w_2 \wedge \ldots \wedge w_n \rightarrow c$, avec d =($w_1$, w_2, . . . w_n) est un document du dataset d'apprentissage et c sa classe.

- L'algorithme d'apprentissage applique ensuite une série de généralisations (par exemple, en supprimant des termes et en fusionnant des règles) maximisant la compacité des règles tout en conservant la propriété de couverture.

Modèles traditionnels : KNN

- L'algorithme d'apprentissage kNN (*k*-nearest neighbor) se base sur les similarités entre les documents pour les classifier.

- Il permet de décider si un document appartient ou non à une classe c en vérifiant si les k documents les plus similaires appartiennent également à la classe c.

- Si la réponse est positive pour une proportion suffisamment importante des k documents, une décision positive est prise; dans le cas contraire, la décision est négative.

- La performance de l'algorithme d'apprentissage dépend étroitement de la valeur k.

Modèles traditionnels : SVM

- Dans le cas d'une classification binaire, le classificateur SVM peut être vue comme un **hyperplan** dans l'espace des features qui sépare les points appartenant à la classe de ceux qui n'y appartient pas.

- l'algorithme SVM semble être très efficace pour la classification de texte en la comparant avec les autres algorithmes:
  - Besoin d'un dataset d'apprentissage d'une petite taille.
  - Algorithme très rapide en la comparant avec les autres algorithmes.

Modèles traditionnels : Bagging and Boosting

- Deux techniques d'apprentissage ensembliste: le bagging et le boosting

  - le bagging démarre en parallèle k processus d'apprentissage avec k algorithmes. Par la suite, une classe est retenue pour un document si au minimum (k+1)/2 algorithmes décident d'attribuer au document la classe C .

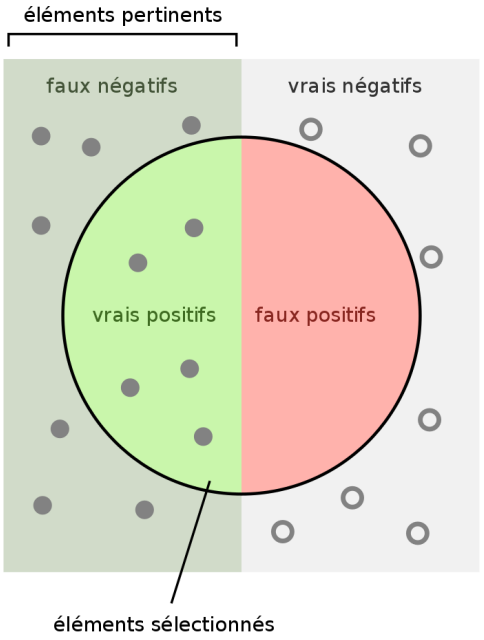
  - Le boosting exécute les algorithmes d'apprentissage séquentiellement l'un après l'autre. Avant de passer à l'algorithme d'apprentissage suivant les poids des différents documents du dataset d'apprentissage seront ajustés de telle sorte que les documents qui été bien classifiés vont perdre du poids et les autres documents vont gagner du poids. L'algorithme **AdaBoost** la plus utilisé pour le boosting.

Modèles traditionnels : Performances

- Les plus performants : SVM, AdaBoost, kNN, et RL.
- Peu performants: Neural Networks et Arbre de Décision.
- Les moins performants: Rocchio, Naïve Bayes (de préférence les utilisé comme baseline ou dans un apprentissage ensembliste).

Evaluation du Modèle

- Pour mesurer les performances de la classification le **rappel**(recall) et la **précision** sont les plus courantes.
  - Le rappel d'une classe est défini comme étant le pourcentage des documents qui sont classifiés correctement par rapport à tous les documents qui appartiennent à cette classe.
  - La précision d'une classe est définie comme étant le pourcentage des documents qui sont classifiés correctement parmi l'ensemble des documents attribués à cette classe.



Evaluation du Modèle

- Pour avoir un équilibre entre le rappel et la précision on peut se baser sur **le seuil de rentabilité** qui correspond au point de la courbe rappel/précision pour lequel la précision et le rappel sont égaux.

- On peut utiliser egalement le **F_mesure** qui représentent les deux mesures; F_Mesure=2x(rappel x précision)/((rappel + précision)