

Introduction au Machine learning

M1MINT

Text mining: comment transformer un texte en un vecteur numérique ?

Agathe Guilloux

Hashing

Hashing

- ▶ **Idée:** réduire le nombre de valeurs d'une variable nominale avec des valeurs dans un grand ensemble \mathcal{D}

Hashing

- ▶ Construction d'une fonction de *hashage* $H: \mathcal{D} \rightarrow \{1, \dots, V\}$ et on utilise les valeurs hashées au lieu des valeurs originales.
- ▶ La fonction de hashage doit être la *plus injective possible...*, du moins au sens probabiliste.

Construire une telle fonction est un art !

Bag of Words

Bag of Words

- ▶ Comment transformer un **texte** en vecteur numérique de features ?

La stratégie “Bag of Words strategy”

- ▶ Créer un *dictionnaire* de mots,
- ▶ Calculer un *poids* pour chaque mot.

Construction d'une liste

- ▶ Faire une liste de tous les mots avec le nombre d'occurrence
- ▶ Réunir les mots qui ont la même racine (stemming)
- ▶ Hash les racines avec une fonction de hashage (MurmurHash avec 32bits par exemple)
- ▶ Calculer l'histogramme $h_w(d)$

Calcul des poids

- ▶ Calculer l'histogramme $h_w(d)$
- ▶ Re-normaliser :
 - ▶ tf transformation (profil du mot):

$$\text{tf}_w(d) = \frac{h_w(d)}{\sum_w h_w(d)}$$

de telle sorte que $\text{tf}_w(d)$ est la fréquence dans le document d .

- ▶ tf-idf transformation (profil du mot re-pondéré par sa rareté):

$$\text{tf-idf}_w(d) = \text{idf}_w \times \text{tf}_w(d)$$

avec idf un poids dépendant du corpus

$$\text{idf}_w = \log \frac{n}{\sum_{i=1}^n \mathbf{1}_{h_w(d_i) \neq 0}}$$

- ▶ Utiliser le vecteur $\text{tf}(d)$ (or $\text{tf-idf}(d)$) pour décrire un document.
- ▶ C'est le pré-processing le plus classique en textmining.

Clustering de textes

Probabilistic latent semantic analysis (PLSA)

- Modèle:

$$\mathbb{P}(\text{tf}) = \sum_{k=1}^K \mathbb{P}(k) \mathbb{P}(\text{tf}|k)$$

avec k le thème caché, $\mathbb{P}(k)$ la probabilité du thème et $\mathbb{P}(\text{tf}|k)$ une loi multinomiale pour le thème.

- Clustering avec un modèle de mélange

$$\mathbb{P}(k|\text{tf}) = \frac{\widehat{\mathbb{P}(k)} \widehat{\mathbb{P}(\text{tf}|k)}}{\sum_{k'} \widehat{\mathbb{P}(k')} \widehat{\mathbb{P}(\text{tf}|k')}}}$$

- Modèle de mélange
- Il existe une variante bayésienne appelée Latent Dirichlet Allocation.

Mots et Word Vectors

Word Vectors

Word Embedding

- ▶ On construit une représentation des mots dans \mathbb{R}^d .
- ▶ en espérant que la relation entre 2 vecteurs est liée à la relation entre les 2 mots dont ils sont issus.

Word And Context

Look ! A single word and its context

Le mot et son contexte

- ▶ **Idée:** caractériser un mot w par son contexte c ...
- ▶ **Description probabiliste:**
 - ▶ Loi jointe : $f(w, c) = \mathbb{P}(w, c)$
 - ▶ Lois conditionnelles: $f(w, c) = \mathbb{P}(w|c)$ or $f(w, c) = \mathbb{P}(c|w)$.
 - ▶ Information mutuelle : $f(w, c) = \mathbb{P}(w, c) / (\mathbb{P}(w) \mathbb{P}(c))$
- ▶ Le mot w est caractérisé par le vecteur $C_w = (f(w, c))_c$ ou $C_w = (\log f(w, c))_c$.
- ▶ En pratique, on estime C sur un large corpus
- ▶ Attention : c'est un modèle de très grande dimension !
- ▶ GloVe (Global Vectors) via les moindres carrés
- ▶ Word2vec via la régression logistique
- ▶ Singular value decomposition