Bases de données documentaires et distribuées, http://b3d.bdpedia.fr

Recherche plein texte

Pondération des termes

Le classement s'appuie sur l'idée qu'il est possible d'identifier **l'importance des termes** dans un document. Deux idées essentielles :

Plus un terme est **fréquent dans un document**, plus il est représentatif du contenu du document.

> Ex. : *contrepoint* apparaît 10 fois dans le document d, $\Rightarrow$ d est un document important pour une recherche sur "contrepoint".
>
> **Normalisation** : si d est très long, il est normal de trouver plus d'occurrences des termes ; on peut décider de *normaliser* cet indicateur pour éviter ce biais.

Plus un terme est **rare dans la collection**, plus sa présence dans un document est importante.

> Ex. : *musicologie* apparaît 100 fois dans une grande collection, 4 fois dans le document d $\Rightarrow$ d est un document important pour une recherche sur "musicologie".

Premier indicateur : la fréquence du terme

Soit un document d, t un terme.

La **fréquence** du terme t dans d, noté $n_{t,d}$, est simplement le nombre d'occurrences de t dans d.

Attention

On parle bien de **termes**, résultats de la normalisation lexicale (racinisation, élimination des mots inutiles, etc.)

Exemple : soit d le document suivant :

```
Spider Cochon Spider Cochon, il peut marcher au plafond,
Est ce qu'il peut faire une toile ? Bien sûr que non,
c'est un cochon. Prends garde ! Spider Cochon est là !
```

Donc $tf(\text{cochon}, \text{d}) = 4$

Deuxième indicateur : fréquence inverse des documents

Soit t un terme, une collection D. On mesure sa **rareté** de t par l'inverse de sa fréquence dans D.

Le nombre total de documents est $|D|$
le nombre de documents avec t est $|\{d' \in D, |, n_{t,d'} > 0\}|$

La rareté de t est donc mesurée par :

$$\frac{|D|}{|\{d' \in D \,|\, n_{t,d'} > 0\}|}$$

Ajustement. La valeur obtenu par la formule ci-dessus croît très vite avec la taille de la collection. On ajuste en prenant le logarithme et on obtient la **fréquence inverse des documents** (*inverse document frequency*, idf)

$$\mathrm{idf}(t) = \log \frac{|D|}{\left|\left\{d' \in D \,|\, n_{t,d'} > 0\right\}\right|}.$$

Pondération par TF-IDF

Le poids d'un terme dans un document est représenté par l'indicateur **Term Frequency—Inverse Document Frequency** (tf-idf)

$$\text{tfidf}(t, d) = n_{t,d} \cdot \log \frac{|D|}{\left|\left\{d' \in D \mid n_{t,d'} > 0\right\}\right|}$$

$n_{t,d}$	nombre d'occurrences de t dans d
D	ensemble de tous les documents

le tf-idf décroît quand un terme est présent dans beaucoup de documents ;
il décroît également quand il est peu présent dans un document ;
il est maximal pour les termes peu fréquents apparaissant beaucoup dans un document particulier.

Descripteurs

Le tf.idf remplace l'indicateur 0/1 dans la matrice d'incidence.

Pondérons nos cochons

Voici des documents A, B et C.

```
Spider Cochon Spider Cochon, il peut marcher au plafond,
Est ce qu'il peut faire une toile ? Bien sûr que non,
c'est un cochon. Prends garde ! Spider Cochon est là !

Un petit cochon, pendu au plafond

Les Trois Petits Cochons est un conte traditionnel européen
mettant en scène trois jeunes cochons et un loup.
```

Test

Calculer le tf et l'idf pour les termes "cochon", "loup" et "plafond" et pour chaque document. En déduire la matrice d'incidence.

Calcul de la similarité

À ce stade, on peut **décrire** un document par un **vecteur** composé des **poids** de tous ses termes.

$$\text{Descr(C)} = [\text{'cochon'} : 2,\ \text{'plafond'} : 0,\ \text{'loup'} : 4]$$

Le poids est à 0 pour les termes qui n'apparaissent pas dans le document. On a un espace vectoriel $E = \mathbb{N}^{|V|}$, V étant le vocabulaire.

['cochon' : 0, ..., 'jaguar' : 4, ..., 'loup' : 0, ..., 'python' : 2, ..., 'mouton' : 0]

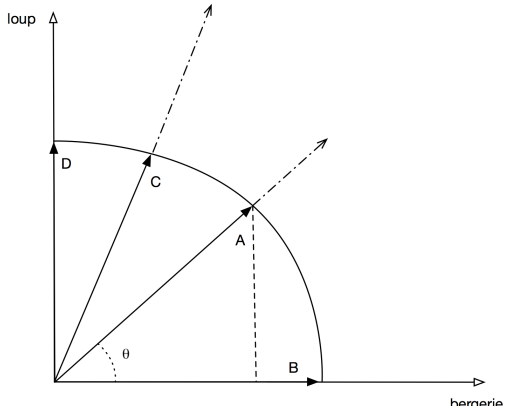
L'espace a **beaucoup** de dimensions (des millions d'axes / termes).
Chaque vecteur est principalement constitué de 0.

Distance Euclidienne ?

Potentiellement très coûteuse à calculer, et introduit un biais lié à la longueur des documents.

Classement par cosinus

Plus deux documents sont “proches” l’un de l’autre, plus l’angle de leurs vecteurs descripteurs est petit.



Rappel : le cosinus est une fonction décroisante sur l’intervalle $[0, 90]$.

La similarité cosinus

La **similarité cosinus** est un bon candidat pour mesurer la proximité des vecteurs dans $E\mathbb{N}^{|V|}$

- Indifférent la la **longueur** (norme) des vecteurs.
- Maximal si même direction (angle $= 0$, cosinus $= 1$)
- Minimal si directions "orthogonales" (pas de terme en commun)
- Varie continuement entre 0 et 1.

En pratique

Calcul efficace car nécessite seulement les coordonnées non nulles.

Calcul du cosinus de deux vecteurs

Produit scalaire de deux vecteurs (niveau 3ème !)

$$v_1.v_2 = ||v_1|| \times ||v_2|| \times cos\theta = \sum_{i=1}^{n} v_1[i] \times v_2[i]$$

où θ désigne l'angle entre les deux vecteurs et $||v||$ la norme d'un vecteur.

Donc :

$$cos\theta = \frac{\sum_{i=1}^{n} v_1[i] \times v_2[i]}{||v_1|| \times ||v_2||}$$

Normalisation

La division par la norme revient à éliminer le biais lié à la longueur des documents.

Calcul du cosinus, en pratique

Première étape : on calcule la **norme** du vecteur représentant chaque document, on la stocke.

Norme du vecteur $\vec{d}$:

$$||\vec{d}|| = \sqrt{\Sigma_i d_i^2}$$

Seconde étape : d le document, q la requête ; on prend les vecteurs $\vec{q}$ (calculé à la volée) et $\vec{d}$ (stocké dans un **index**).

$$\frac{1}{||\vec{d}||} \times \frac{1}{||\vec{q}||} \times \Sigma_i d_i q_i$$

Approximation

On peut ignorer la norme de la requête (qui est constante), et la racine carré pour la calcul des normes : c'est le classement qui nous intéresse.