Pré-traitements classiques ou par analyse distributionnelle : application aux méthodes de classification automatique déployées pour DEFT08

Eric Charton, Nathalie Camelin, Rodrigo Acuna-Agost,
Pierre Gotab, Remi Lavalley, Remy Kessler et Silvia Fernandez

Laboratoire Informatique d'Avignon

13 juin 2008

Classification en genre et en thème

Comment s'attaquer au problème ?

- 7 participants pour l'équipe jeunes chercheurs
- Choix communs :
  - Dissocier genre et thème
  - Choisir : 1 méthode, des pré-traitements

$\longrightarrow$ chacun implémente son propre système

$\longrightarrow$ proposition de différentes techniques de fusion

Analyse des corpus

- Tâche1 : Classification en genre et en thème

CORPUS1	*ECO*	*TEL*	*ART*	*SPO*	*LM*	*W*	Échantillons
APP1	30.41%	8.88%	37.88%	22.82%	57.97%	42.02%	15223
EVAL1	29.11%	12,75%	36.27%	21.84%	53.63%	46.36%	10596

Tab.: Répartition des volumes de documents de *CORPUS1* pour chaque classe

- Tâche2 : Classification en thème

CORPUS2	*SOC*	*FRA*	*INT*	*LIV*	*SCI*	Échantillons
APP2	16.04%	14.12%	22.52%	19.43%	27.87%	23550
EVAL2	16.03%	14.12%	22.53%	19,42%	27.87%	15693

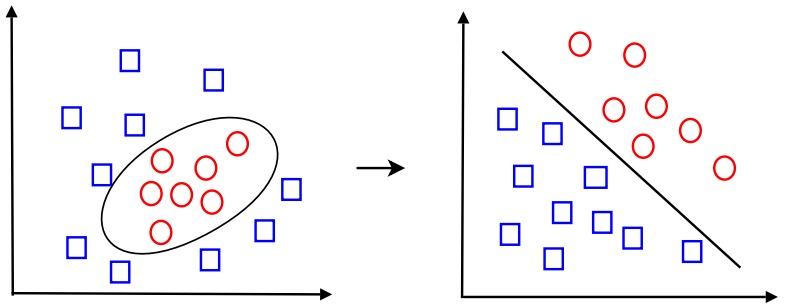
Tab.: Répartition des volumes de documents de *CORPUS2* pour chaque classe

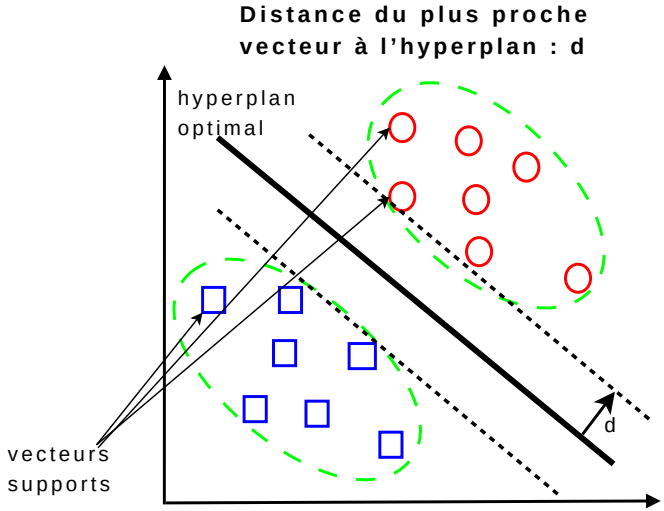
Les *SVM* : Les machines à supports vectoriels

Un classifieur à large marge

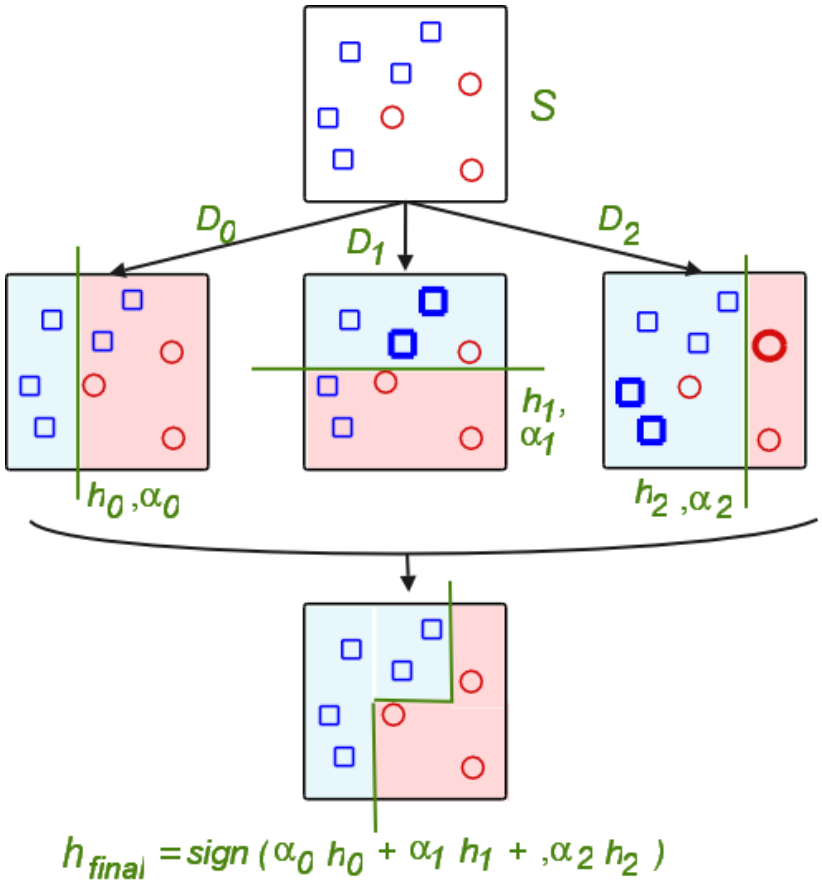
- trouver un espace où les données soient linéairement séparables ;
- séparer les données par un hyperplan optimal.

$\longrightarrow$ L'hyperplan optimal implique une marge maximale de séparation des données





Algorithme du boosting



un classifieur à large marge

- exécution répétitive d'un apprenant *faible* ;
- re-pondération des données à chaque tour d'exécution.

→ Améliorer la précision des règles de classification en combinant plusieurs hypothèses *faibles*

Classification Bayésienne Naïve

Formule générique

La probabilité qu'un document D contenant des mots m appartienne à une classe k est égale à $P(k|D) = \frac{p(D|k)(p(k)}{p(D)}$.

Estimation d'appartenance à une classe

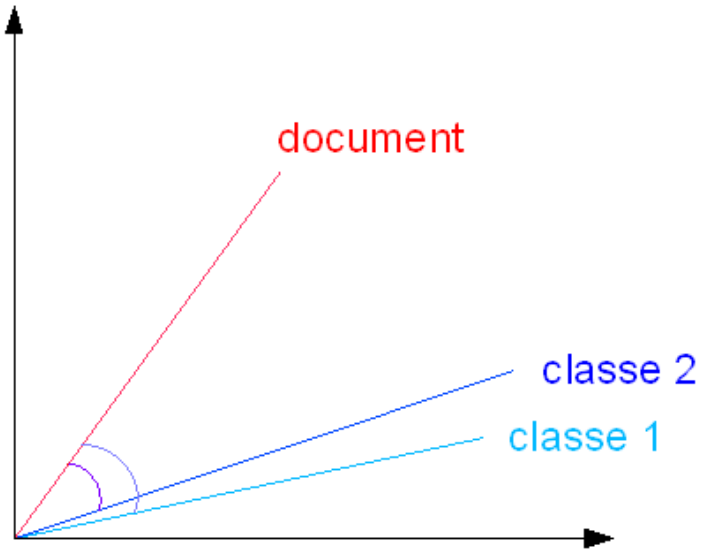
Les classifieurs bayésiens naïfs appliquent la méthode de l'estimateur de maximum de vraisemblance pour décider de l'attribution d'une classe

- $P(k|D) = p(m_1, m_2, ..., m_n|k) = \prod_{i=1}^{n} p(m_i|k)$.

Mesure de similarité cosine

Un espace de dimension égal à |mots du vocabulaire|

- représentation des documents dans cet espace ;
- calcul du cosinus de l'angle entre le document à classer et les différentes classes possibles.

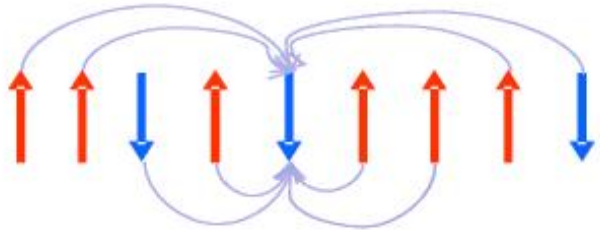


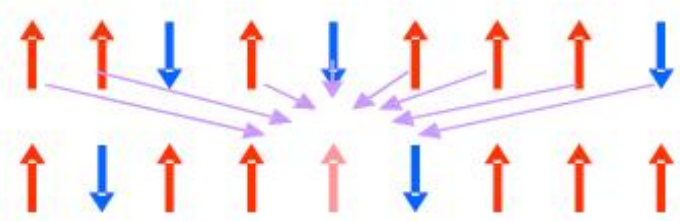
Énergie textuelle : le texte comme système magnétique

$$E = S \times J \times S^T$$

S :terme-segment

J :co-ocurrence des termes





Apprentissage

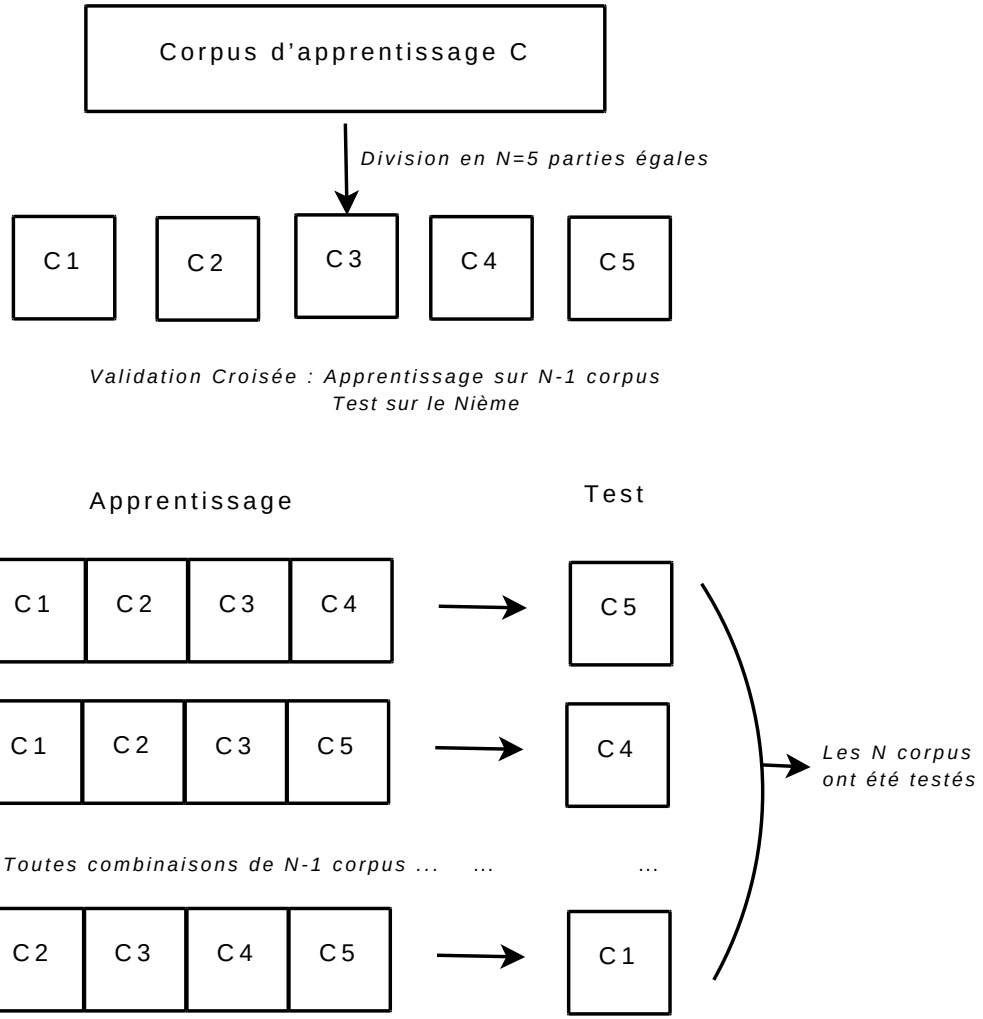
- Catégorie=matériau
- Purifier les matériaux (vocabulaire exclusif)
- Calcul des J_{cat}

Test : pour chaque document inconnu s

- Calcul des énergies $E_{cat} = s \times J_{cat} \times s^T$

$\rightarrow$ s sera un échantillon du matériau pour lequel la plus grande quantité d'énergie E est présente

Protocole expérimental



Pré-traitements classiques

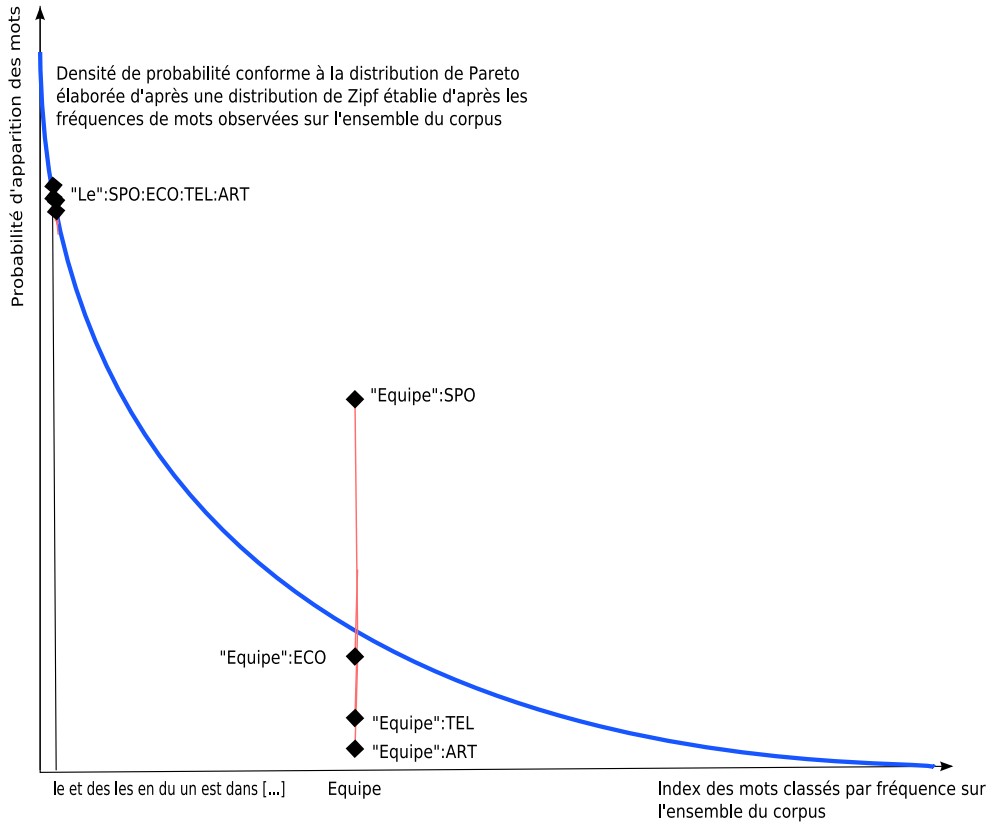
- suppression des marques typographiques
- minusculisation
- POS : représentation morpho-syntaxique
- LEMME : forme canonique
- N-GRAM : agglutination de mots
- antidico : suppression de mots fonctionnels, d'expressions courantes, de chiffres, de symboles . . .

Hypothèse distributionnelle et distributions de Zipf-Mandelbrot

- La loi de Zipf-Mandelbrot est une distribution de probabilité discrète (index de mots).
- Cette loi prédit que si dans un texte de longueur N on range les mots dans l'ordre de leur fréquence d'apparition, alors la fréquence $f(r)$ du mot de rang r est approximativement de forme : $f(r) = \frac{K}{r}$
- L'idée : générer une distribution de Zipf-Mandelbrot pour chaque corpus et retirer les intersections pour accentuer leur caractère discriminant.
- On utilise le critère de Gini pour mesurer les intersections
- Distributions sur des tri-grammes avec suppression des marques typographiques (pas d'antidictionnaire, de POS ou de lemme)

Hypothèse distributionnelle et distributions de Zipf-Mandelbrot

Principe



Présentation des systèmes

Sept systèmes

- **SVM_Baseline** : filtrage par antidico, noyau linéaire
- **SVM_Extended** : pré-traitement par analyse distributionnelle
- **N_Bayes_extended** : pré-traitement par analyse distributionnelle
- **BoosTexter** : lemmes, tri-grammes, 1500 tours
- **isciboost** : lemmes et pos, tri-grammes, 2500 tours
- **Cosine_Discriminant** : lemmes, filtrage par antidico, pas de minusculisation
- **Enertex** : pré-traitement par analyse distributionnelle

Stratégies de fusion (1/2)

Fusion par vote majoritaire

- Utilisation des résultats des trois meilleurs systèmes (SVM_Extended, N_Bayes_extended, isciboost)
  - le vote majoritaire l'emporte (2/3)
- Si pas de classe majoritaire
  - repli sur le système le plus performant (SVM_Extended sur les thèmes et isciboost sur le genre)

Stratégies de fusion (2/2)

Fusion probabiliste avec BoosTexter

- 4 systèmes : SVM_Extended, N_Bayes_extended, isciboost et BoosTexter
- représenter une phrase par l'ensemble des résultats fournis par ces systèmes

Exemple :

SPO_{N_Bayes} 0.98140 ECO_{N_Bayes} 0.00431 TEL_{N_Bayes} 0.00984 ART_{N_Bayes} 0.00444, ART_{BoosT} 0.4051887 ECO_{BoosT} 0.406487 SPO_{BoosT} 0.7468423 TEL_{BoosT} 0.3704997, ART_{isci} 0.3422 ECO_{isci} 0.2983 SPO_{isci} 0.7131 TEL_{isci} 0.3311, SPO_{SVM} 1.00

Phase d'apprentissage

Fscores APP	tâche1-catégorie	tâche1-genre	tâche2-catégorie
SVM_baseline	0.8391	0.93	0.78
SVM_extended	**0.9150**	0.9594	**0.8445**
N_Bayes_extended	0.8629	0.9353	0.8271
BoosTexter	0.8958	**0.9869**	0.8316
icsiboost	0.9051	**0.9858**	**0.8409**
Cosine_Discriminant	0.8508	0.9222	0.8244
Enertex	0.8328	0.8390	0.7561
Fusion ternaire	0.9192	0.9870	**0.8676**
Fusion probabiliste	**0.9292**	**0.9880**	0.8599

Performances des systèmes lors de la phase d'apprentissage et fusion

Phase de test

	Précision	Rappel	Fscore
Fusion par vote ternaire majoritaire			
tâche1-genre	0.9795	0.9800	0.9798
tâche1-catégorie	0.9082	0.8448	0.8754
tâche2-catégorie	0.8814	0.8759	0.8786
Fusion probabiliste			
tâche1-genre	0.9584	0.9600	0.9592
tâche1-catégorie	0.7919	0.8267	0.8089
tâche2-catégorie	0.8124	0.5584	0.6618
SVM_Extended : tâche1-catégorie et isciboost :tâche1-genre,tâche2-catégorie			
tâche1-genre	0.9795	0.9800	0.9798
tâche1-catégorie	0.8972	0.8006	0.8442
tâche2-catégorie	0.8553	0.8497	0.8525

Évaluation : Performances des 3 exécutions

Conclusion

- Découplage de la tâche 1
- Pré-traitements classiques et par analyse distributionnelle
- Utilisation de diverses méthodes de classification qui se complètent
- La méthode de fusion la plus simple est la plus efficace
- Faible différence entre apprentissage et test
  - tâche1-genre : performances similaires
  - tâche1-catégorie : perte de 5 points
  - tâche2-catégorie : meilleures performances ! !