Apprentissage pour la Recherche d'Information

Massih-Reza Amini

Université Joseph Fourier
Laboratoire d'Informatique de Grenoble
Massih-Reza.Amini@imag.fr

UNIVERSITE JOSEPH FOURIER
SCIENCES. TECHNOLOGIE. SANTÉ

Apprentissage Automatique

Qu'est-ce que l'apprentissage ?

On considère un espace d'entrée $\mathcal{X} \subseteq \mathbb{R}^d$ et un espace de sortie $\mathcal{Y}$.

Hypothèse : Les paires d'exemples $(x, y) \in \mathcal{X} \times \mathcal{Y}$ sont *identiquement* et *indépendamment* distribuées (`i.i.d`) suivant une distribution de probabilité inconnue $\mathcal{D}$.

Échantillons : On observe une séquence de m paires (x_i, y_i) générées `i.i.d` suivant $\mathcal{D}$.

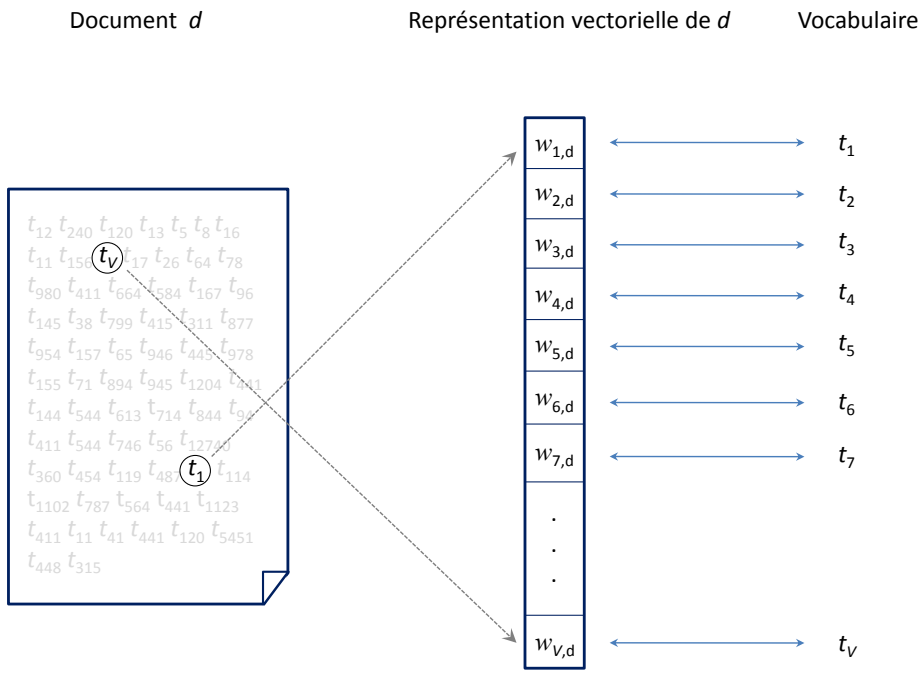
But : Construire une fonction $f : \mathcal{X} \to \mathcal{Y}$ qui prédit la sortie y d'une nouvelle observation x avec une probabilité d'erreur minimale.

Théorie de l'apprentissage [Vapnik88] : Borner cette probabilité d'erreur, $R(f)$

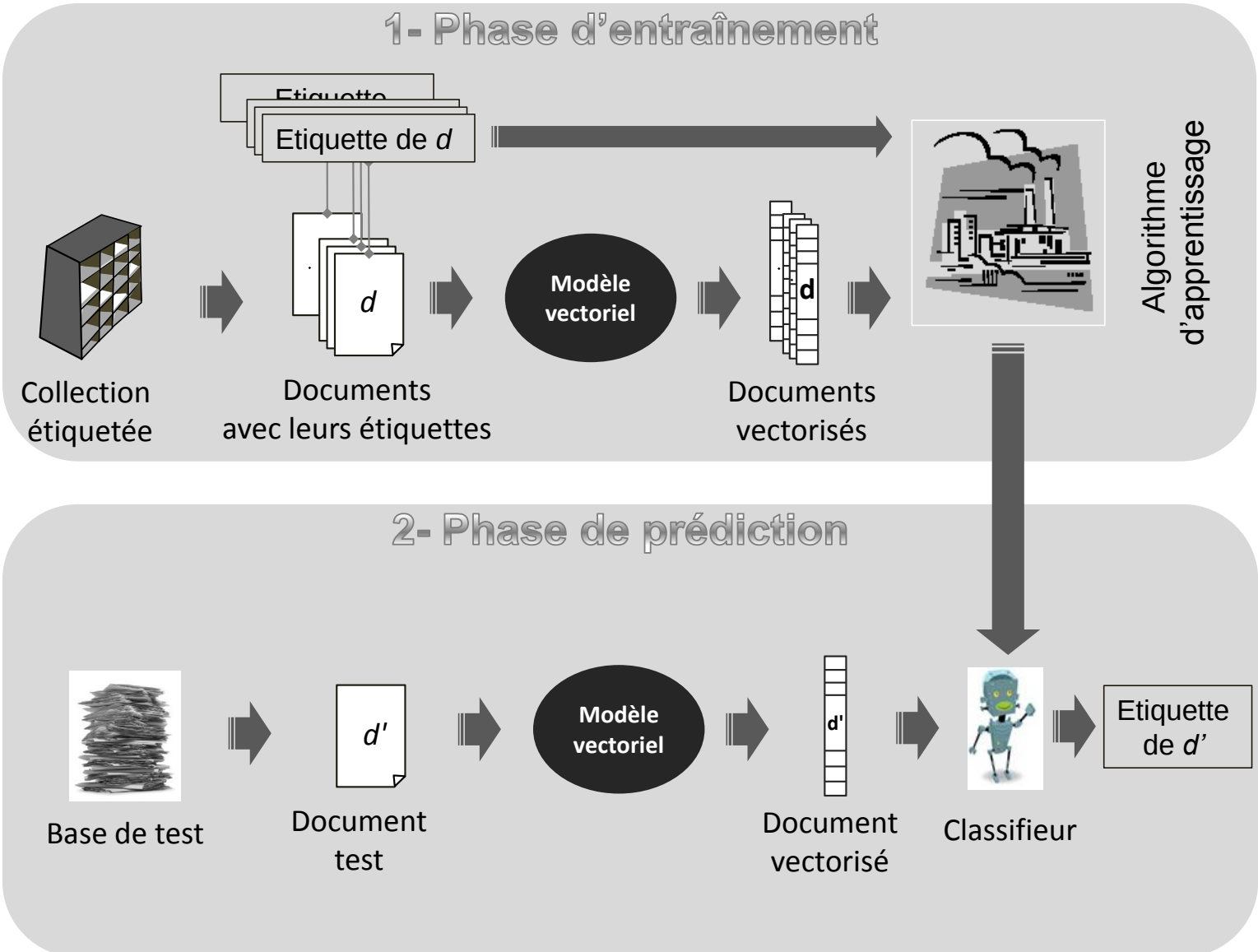
$R(f) \leq$ Erreur empirique de f + Complexité de la classe de fonctions + Terme résiduel

Remarques

1. Dans le cas de la catégorisation binaire, on considère généralement $\mathcal{Y} = \{-1, 1\}$.
2. En catégorisation de documents, un document est représenté par un vecteur dans l'espace vectoriel des termes (représentation sac de mots).



Catégorisation de documents



Modèles génératifs

- Les modèles génératifs sont les premiers modèles d'apprentissage développés pour la catégorisation de documents.
- L'hypothèse fondamentale est que chaque vecteur représentatif d'un document, **d**, est la réalisation d'une variable aléatoire multidimensionnelle, généré par le mélange de K densités de probabilités avec des proportions $(\pi_k)_{k=1}^{K}$:

$$\sum_{k=1} \pi_k = 1 \quad \text{et} \quad \pi_k \geq 0 \ (k = 1, ..., K).$$

- Chaque fonction de densité est une fonction paramétrique modélisant la distribution de probabilité conditionnelle :

$$\forall k \in \mathcal{Y}; P(\mathbf{d} \mid y = k) = f_k(\mathbf{d}, \theta_k)$$

- La densité de mélange modélise ainsi la génération de **d** par ces K densités de probabilité :

$$P(\mathbf{d}, \Theta) = \sum_{k=1}^{K} \pi_k f_k(\mathbf{d}, \theta_k)$$

Modèles génératifs (2)

- Où Θ est l'ensemble des proportions π_k ainsi que tous les paramètres définissant les fonctions de densité f_k :

$$\Theta = \{\theta_k, \pi_k : k \in \{1, ..., K\}\}$$

- Le but de l'apprentissage est alors d'estimer l'ensemble des paramètres Θ pour qu'au sens du maximum de vraisemblance le modèle de mélange explique au mieux les exemples de la base d'entraînement.
- Une fois l'estimation de ces paramètres terminée, un nouveau document d' est affecté à une classe de l'ensemble $\mathcal{Y}$ d'après la règle de la décision bayésienne :

$$d' \text{ appartient à la classe } k \text{ ssi } k = \underset{h \in \mathcal{Y}}{argmax}\, P(y = h \mid \mathbf{d}')$$

Modèle Naïve Bayes (multivarié de Bernoulli)

- Le modèle Naïve Bayes est un des modèles les plus populaires en catégorisation de documents
- Avec ce modèle on suppose que les termes apparaissant dans un document sont indépendants les uns des autres.
- Dans le cas où, chaque document d a une représentation vectorielle binaire $\mathbf{d} = (w_{id})_{i \in \{1,...,V\}}$ où une caractéristique w_{id} est soit égal à 1 ou 0, indiquant si le terme d'indice i du vocabulaire est présent ou pas dans le document d, les densités conditionnelles pour un document d donné s'écrivent :

$$\forall k \in \mathcal{Y}; f_k(\mathbf{d}, \theta_k) = P(\mathbf{d} = (w_{1d}, ..., w_{Vd}) \mid y = k) = \prod_{i=1}^{V} P(w_{id} \mid y = k)$$

Modèle Naïve Bayes (multivarié de Bernoulli)

- En posant $\theta_{t_i|k} = P(w_{id} = 1 \mid C_k = 1)$, la probabilité de présence du terme d'indice i du vocabulaire dans la classe k, ces densités conditionnelles s'écrivent :

$$\forall d; \forall k \in \mathcal{Y}; f_k(\mathbf{d}, \theta_k) = \prod_{i=1}^{V} \theta_{t_i|k}^{w_{id}} (1 - \theta_{t_i|k})^{1 - w_{id}}$$

- Les estimés au sens du maximum de vraisemblance des paramètres du modèle de Bernoulli sur une base d'entraînement, S, de taille m :

$$\forall i \in \{1, ..., V\}, \forall k \in \mathcal{Y}, \quad \hat{\theta}_{t_i|k} = \frac{\mathrm{df}_{t_i}(k)}{N_k(S)}$$

$$\forall k \in \mathcal{Y}, \quad \hat{\pi}_k = \frac{N_k(S)}{m}$$

Où $N_k(S)$ est le cardinale de la classe k et $df_t(k)$ est le nombre de documents de la classe k contenant le terme t.

Modèles discriminants

- Les modèles discriminants trouvent directement une fonction de classification $f : \mathbb{R}^V \rightarrow \mathcal{Y}$ qui résout le problème de catégorisation, sans faire d'hypothèses sur la génération des exemples.
- Le classifieur recherché est cependant supposé appartenir à une classe de fonction donnée $\mathcal{F}$ et sa forme analytique est trouvée en minimisant une certaine *fonction d'erreur* (appelée aussi *risque* ou *fonction de perte*)

$$L : \mathcal{Y} \times \mathcal{Y} \rightarrow \mathbb{R}^+$$

La fonction de risque usuelle généralement considérée en catégorisation est l'erreur de classification :

$$\forall (d, y); L(f(\mathbf{d}), y) = [\![f(\mathbf{d}) \neq y]\!]$$

Où $[\![\pi]\!]$ vaut 1 si le prédicat π est vrai et 0 sinon.

Modèles discriminants (2)

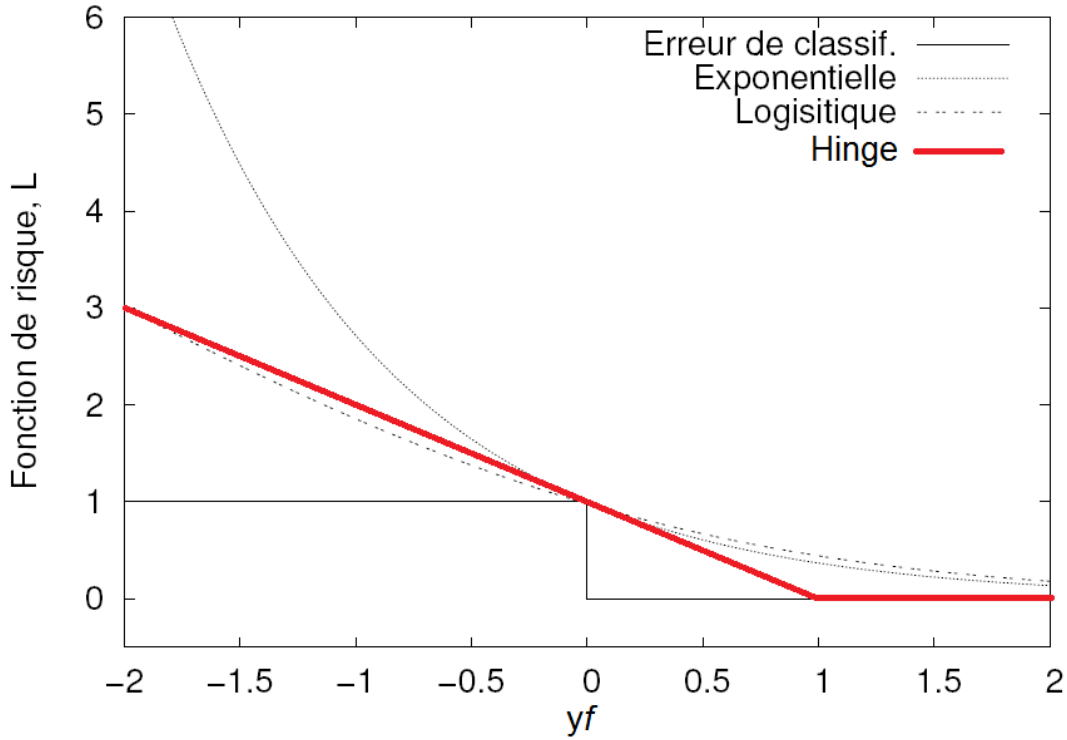
- Rappel : le classifieur qu'on apprend doit être capable de faire de bonnes prédictions sur de nouveaux exemples, où avoir une faible erreur de généralisation, qui avec l'hypothèse de génération i.i.d des exemples s'écrit :

$$R(f) = \mathbb{E}_{(\mathbf{d},y)\sim\mathcal{D}} L(\mathbf{d}, y) = \int_{\mathcal{X}\times\mathcal{Y}} L(f(\mathbf{d}), y) d\mathcal{D}(\mathbf{d}, y)$$

- Principe de la minimisation du risque empirique : Trouver f en minimisant l'estimateur non-biaisé de R sur une base d'entraînement $S = (\mathbf{d}_i, y_i)_{i=1}^m$:

$$\hat{R}_m(f, S) = \frac{1}{m} \sum_{i=1}^{m} L(f(\mathbf{d}_i), c_i)$$

Modèles discriminants (3)



Les fonctions loss considérées :

Coût logistique $L_\ell(f(\mathbf{d}), y) = \ln(1 + \exp(-yf(\mathbf{d})))$

Coût exponentiel $L_e(f(\mathbf{d}), y) = e^{-yf(\mathbf{d})}$

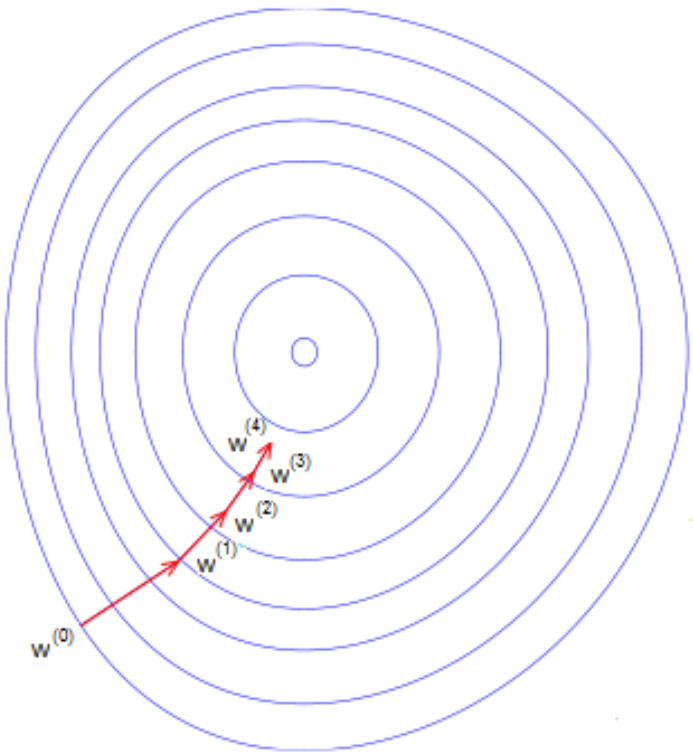
Fonction hinge $L_h(f(\mathbf{d}), c) = [\![(1 - yf(\mathbf{d})) > 0]\!]$

Descente de gradient

- La descente de gradient est un algorithme d'optimisation de première ordre largement utilisé pour trouver le minimum d'une fonction de coût convexe ou dérivable.

Algorithm 1 Descente de gradient

1: Initialiser les poids $w^{(0)}$
2: $t \leftarrow 0$
3: Pas d'apprentissage $\lambda > 0$
4: Précision $\epsilon > 0$
5: **repeat**
6: $\quad w^{(t+1)} \leftarrow w^{(t)} - \lambda \nabla_{w^{(t)}} \mathcal{L}(w^{(t)})$
7: $\quad t \leftarrow t + 1$
8: **until** $|\mathcal{L}(w^{(t)}) - \mathcal{L}(w^{(t-1)})| < \epsilon$



Quelques algorithmes standards

- Perceptron
- Les séparateurs à vaste marge (SVM)

  Mais aussi ....

- Boosting
- Les k plus proches voisins (k-PPV / *k-NN*)
- La régression logistique

Le Perceptron

- Un des premières algorithms d'apprentissage proposé par Rosenblatt dans les années cinquante
- La fonction de prédiction est de la forme

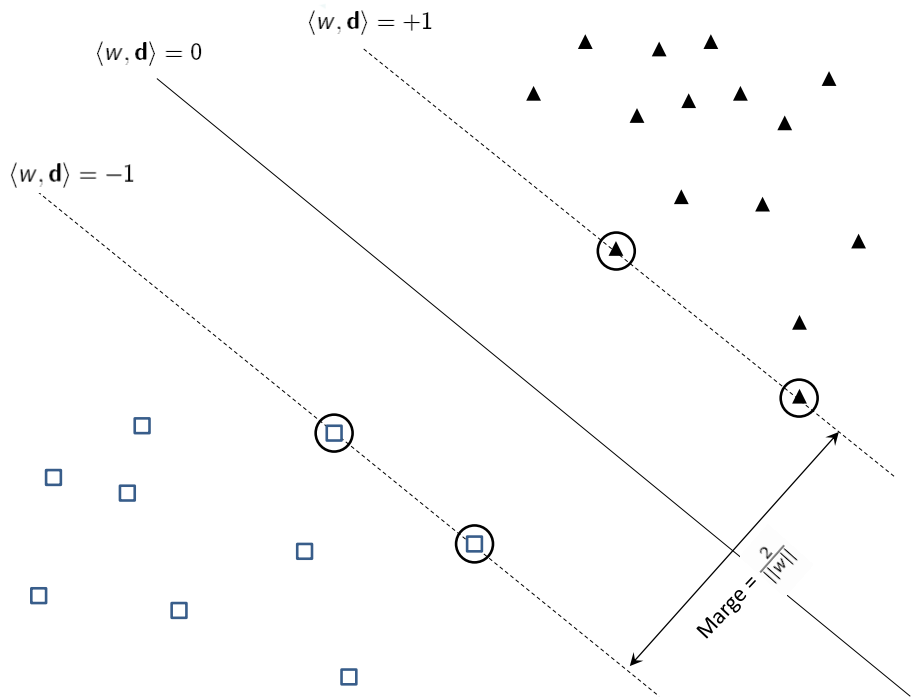
$$\begin{array}{rcl} f_w : \mathbb{R}^V & \mapsto & \mathbb{R} \\ \mathbf{d} & \rightarrow & \langle w, \mathbf{d} \rangle \end{array}$$

Algorithm 2 L'algorithme de perceptron

Base d'apprentissage $S = \{(\mathbf{d}_i, y_i) \mid i \in \{1, \ldots, m\}\}$
Initialiser les poids $w^{(0)}$
$t \leftarrow 0$
Le pas d'apprentissage $\lambda > 0$
Nombre toléré d'exemples mal-classés K
repeat
    Choisir un exemple $(\mathbf{d}, y) \in S$
    if $y \langle w^{(t)}, \mathbf{d} \rangle < 0$ **then**
        $w^{(t+1)} \leftarrow w^{(t)} + \lambda \times y \times \mathbf{d}$
    end if
    $t \leftarrow t + 1$
until Le nombre d'exemples mal-classés est moins que K

Les SVMs (1)

On cherche une fonction de décision de la forme :
$f(\mathbf{d}) = \text{sgn}(\langle w, \mathbf{d} \rangle)$
L'équation $\langle w, \mathbf{d} \rangle = 0$ définit un hyperplan dont la *marge* vaut $\frac{2}{||w||}$



Les SVMs (2)

Trouver l'hyperplan *séparateur* de marge maximale revient donc à résoudre le problème d'optimisation quadratique suivant :

$$\begin{cases} \text{Minimize} & \frac{1}{2} w^T w \\ \text{sous les contraites} & y_i \langle w, \mathbf{d}_i \rangle \geq 1, \ i = 1, \cdots, m \end{cases}$$

Cas non séparable

$$\begin{cases} \text{Minimiser} & \frac{1}{2} w^T w + C \sum_i \xi_i \\ \text{sous les contraintes} & \xi_i \geq 0, \ y_i \langle w, \mathbf{d}_i \rangle \geq 1 - \xi_i, \ i = 1, \cdots, m \end{cases}$$

Evaluation des méthodes de catégorisation

Classe k		Jugement de l'expert	
		Dans la classe	Pas dans la classe
Prédiction du classifieur	Dans la classe	VP_k	FP_k
	Pas dans la classe	FN_k	VN_k

Taux de bonne classification $$TBC_k = \frac{VP_k + VN_k}{VP_k + VN_k + FP_k + FN_k}$$

Précision $$P_k = \frac{VP_k}{VP_k + FP_k}$$

Rappel $$R_k = \frac{VP_k}{VP_k + FN_k}$$

Mesure F_1 $$F_k = \frac{2P_kR_k}{P_k + R_k}$$

$$P_{macro} = \frac{1}{|\mathcal{Y}|}\sum_{k\in|\mathcal{Y}|} P_k, \qquad P_{micro} = \frac{\sum_{k\in\mathcal{Y}} VP_k}{\sum_{k\in\mathcal{Y}} VP_k + FP_k}$$

Catégorisation de documents : collection Reuters-RCV2 (français)

Variables	Symboles	Valeurs
# de documents de la collection		85 167
# de classes initiales		3 586
# de documents considérés	N	70 703
# de termes du vocabulaire	V	141 146
# moyen de termes par document		136.7
# de classes considérées	K	29
# Taille de la base d'entraînement	m	35 000
# Taille de la base de test		35 703

Catégorisation de documents : collection Reuters-RCV2 (français)

Modèle	Type	F_1 Micro	F_1 Macro
Multivarié de Bernoulli	Génératif	0.531	0.502
3-PP		0.561	0.519
Multinomial	Génératif	0.694	0.669
Perceptron	Discriminant	0.715	0.679
Logistique	Discriminant	0.734	0.704
SVM	Discriminant	0.759	0.723
AdaBoost	Discriminant	0.766	0.728

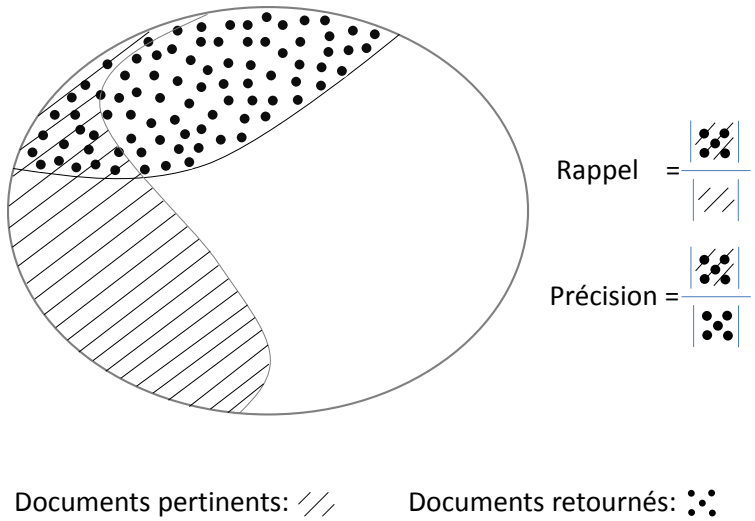
TABLE: Comparaison entre les *micro* et *macro* moyennes des mesures F_1 entre les différents modèles génératifs et discriminants sur la base Reuters RCV-2 français. Représentation binaire pour la méthode multivarié de Bernoulli et fréquentiste (pour le reste). Chaque expérience est répétée 10 fois en séléctionnant aléatoirement les bases d'entraînement et de test de la collection initiale avec les proportions mentionnées dans le tableau d'au-dessus. Chaque performance reportée dans ce tableau est la moyenne des performances obtenues sur les bases tests ainsi échantillonnés.

Evaluation en RI

Les jugements/annotations les plus fréquents

- Jugements binaires : ce doument est pertinent (1) ou non (0) pour cette requête
- Jugements multi-valués :
  Parfait $>$ *Excellent* $>$ *Bon* $>$ *Correct* $>$ *Mauvais*
- Paires de préférence : document d_A plus pertinent que document d_B pour cette requête

Les mesures d'évaluation les plus courantes (jugements binaires)



- Précision moyenne à un rang k donné : $\mathrm{P@}k(q) = \frac{1}{k} \sum_{rg=1}^{k} R_{d_{rg},q}$
- *Average Precision* : $AveP(q) = \frac{1}{n_+^q} \sum_{k=1}^{N} \mathbf{1}_{Per(q)}(k) \times \mathrm{P@}k(q)$
- *Mean Average Precision* :

$$MAP = \frac{1}{|Q|} \sum_{j=1}^{|Q|} AveP(q_j) = \frac{1}{|Q|} \sum_{j=1}^{|Q|} \frac{1}{n_+^{q_j}} \sum_{k=1}^{N} \mathbf{1}_{Per(q_j)}(k) \times \mathrm{P@}k(q_j)$$

Pour les jugements multi-variés : NDCG

- NDCG à la position k :

$$N(k) = \overbrace{Z_k}^{\text{normalisation}} \underbrace{\sum_{j=1}^{k}}_{\text{cumul}} \overbrace{(2^{p(j)} - 1)}^{\text{gain}} \; / \; \underbrace{\log_2(j+1)}_{\text{position discount}}$$

- Score moyenné sur toutes les requêtes

G : Gain

Pertinence	Valeur (gain)
Parfait (5)	$31 = 2^5 - 1$
Excellent (4)	$15 = 2^4 - 1$
Bon (3)	$7 = 2^3 - 1$
Correct (2)	$3 = 2^2 - 1$
Mauvais (0)	$0 = 2^1 - 1$

DCG : Discounted CG

Discounting factor : $\frac{\ln(2)}{\ln(j+1)}$

Doc. (rang)	Pert.	Gain	CG	DCG
1	*Parf. (5)*	31	31	31
2	*Corr. (2)*	3	$34 = 31 + 3$	$32,9 = 31 + 3 \times 0,63$
3	*Exc. (4)*	15	49	$40,4$
4	*Exc. (4)*	15	64	$46,9$
...	...	...	...	...

Ordre idéal : max DCG

Document (rang)	Pertinence	Gain	max DCG
1	*Parfait (5)*	31	31
3	*Excellent (4)*	15	40,5
4	*Excellent (4)*	15	48
...	...	...	...

Normalized DCG

Document (rang)	Pertinence	Gain	DCG	max DCG	NDCG
1	*Parfait (5)*	31	31	31	1
2	*Correct (2)*	3	32, 9	40, 5	0, 81
3	*Excellent (4)*	15	40, 4	48	0.84
4	*Excellent (4)*	15	46, 9	54, 5	0.86
. . .	. . .	. . .	. . .	. . .	

Mesures d'évaluation : remarques

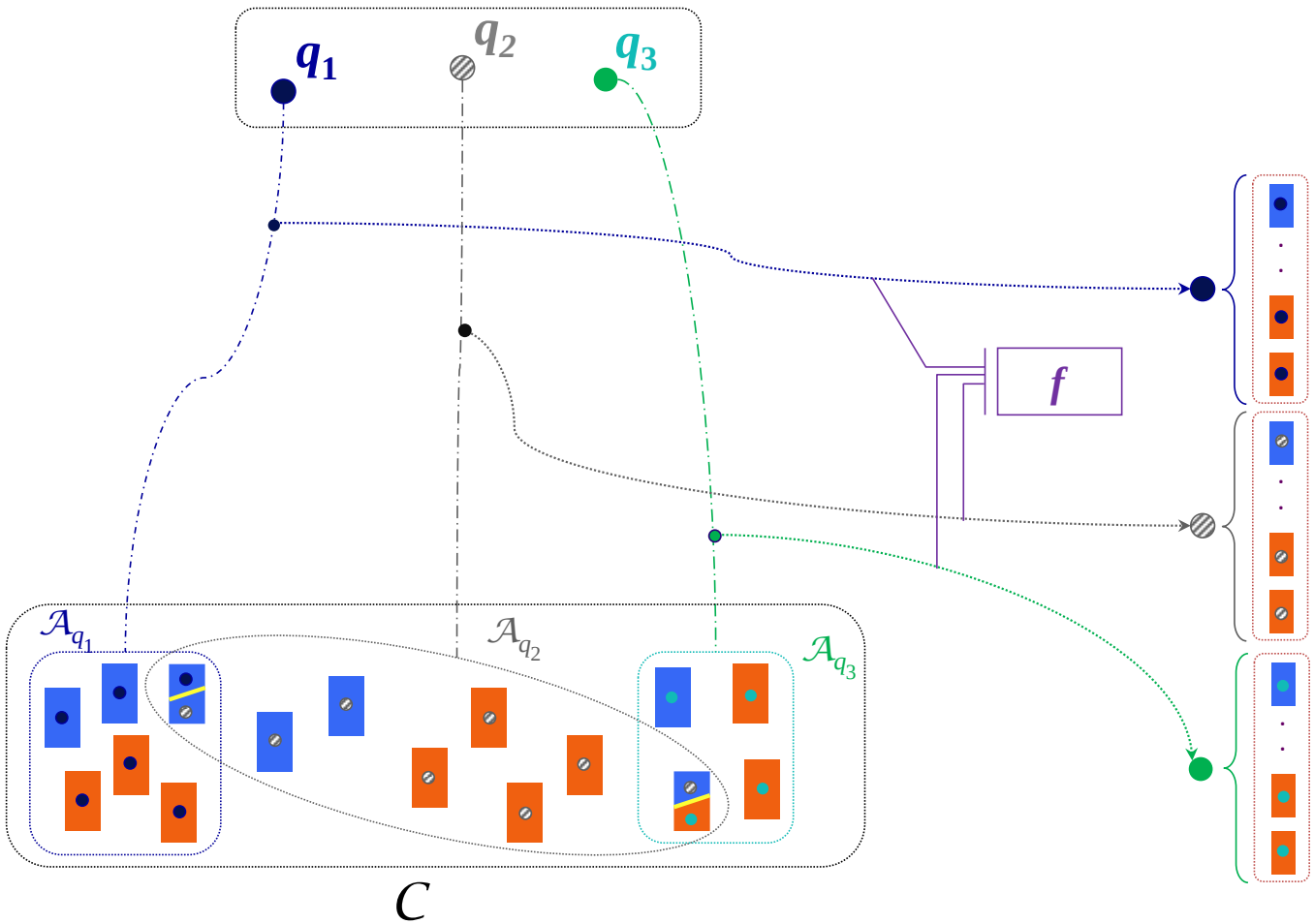
- Mesures pour une position donnée (liste de 10 documents retournés par exemple)
- Mesures au niveau "requêtes"
- NDCG est une mesure plus générale que la MAP (pertinence multi-valuée vs pertinence binaire)
- Mesures non continues (et non dérivables)

RI et catégorisation

UNIVERSITE JOSEPH FOURIER
SCIENCES.TECHNOLOGIE.SANTÉ

Problématique

Requêtes pour lesquelles
on dispose des jugements de pertinence dans C



Modéliser la RI comme un problème de catégorisation

Quelle est l'intuition ?

1. Choisir une représentation des exemples
2. Choisir le nombre de classes
3. Choisir le principe d'apprentissage et l'algorithme associé

Représentation des exemples

Problème crucial : quels types d'exemples considérer ? Docs ?

...

Représentation standard, $x = (q, d) \in \mathbb{R}^m$. Les coordonnées $(f_i(q, d), i = 1, \cdots, p)$ sont très générales. On essaye de se reposer sur un maximum d'information :

- $f_1(q, d) = \sum_{t \in q \cap d} \log(t^d)$, $f_2(q, d) = \sum_{t \in q} \log(1 + \frac{t^d}{|\mathcal{C}|})$
- $f_3(q, d) = \sum_{t \in q \cap d} \log(\text{idf}(t))$, $f_4(q, d) = \sum_{t \in q \cap d} \log(\frac{|\mathcal{C}|}{t^{\mathcal{C}}})$
- $f_5(q, d) = \sum_{t \in q} \log(1 + \frac{t^d}{|\mathcal{C}|} \text{idf}(t))$,
  $f_6(q, d) = \sum_{t \in q} \log(1 + \frac{t^d}{|\mathcal{C}|} \frac{|\mathcal{C}|}{t^{\mathcal{C}}})$
- $f_7(q, d) = \text{RSV}_{\text{vect}}(q, d)$, ...

Choix du nombre de classes

Le choix du nombre de classes dépend *a priori* :

- Des valeurs de pertinence à disposition (binaires ou multi-valuées)
- Des préférences des concepteurs et développeurs du système

Dans le cas le plus simple, on se contente d'un ensemble de deux classes, correspondant aux documents pertinents et non pertinents
$\rightarrow$ Catégorisation binaire

Modèle SVM

- Une fois les données représentées comme ci-dessus, toutes les techniques de catégorisation peuvent *a priori* être utilisées.
- L'application de la méthode vue précédemment est directe ici. Chaque $x(=(q,d))$ contenant un document pertinent pour q est associé à la classe $+1$, les exemples avec des documents non pertinents à la classe -1
- On obtient alors un hyper-plan séparateur, associé à la fonction de décision :

$$g(R|d,q) = \langle w, x \rangle$$

- Remarque : On utilise ici directement la valeur de sortie et non le signe de façon à obtenir un ordre sur les documents

D'autres remarques sur cette approche

1. Cas d'une pertinence multi-valuée : catégorisation multi-classes
2. Méthode qui permet d'attribuer un score, pour une requête donnée, à un document, indépendamment des autres (méthode dite *pointwise*)
3. Résultats comparables à ceux des modèles probabilistes dans le cas de collections "classiques", meilleurs dans le cas du Web (espace d'attributs plus riches)
4. Méthode qui repose sur une notion "absolue" de pertinence
5. La fonction objectif est "éloignée" de la fonction d'évaluation
6. Disponibilité des annotations ?

RI et ordonnancement

Les paires de préférence

- La notion de pertinence n'est pas une notion absolue. Il est souvent plus facile de juger de la pertinence relative de deux documents

- Les jugements par paires constituent en fait les jugements les plus généraux

Représentation des données

Comme précédemment, pour une requête donnée, l'élément fondamental est un couple $x_i = (d_i, q)$ $(1 \leq i \leq n)$. A partir de ces éléments et des jugements de pertinence, on peut former l'ensemble des exemples d'apprentissage sous la forme de paires étiquetées :

$$\{(x_1^{(1)} - x_2^{(1)}, z^{(1)}), \cdots, (x_1^{(p)} - x_2^{(p)}, z^{(p)})\}$$

avec :

$$z^{(i)} = \begin{cases} +1 & \text{si} \quad d_1^{(j)} \succ_{\text{pert}} d_2^{(j)} \\ -1 & \text{sinon} \end{cases}$$

Ranking SVM et RI

On peut alors directement utiliser la méthode *Ranking SVM* vue précédemment pour trouver un hyper-plan séparateur des données. Pour rappel, le problème d'optimisation associé est :

$$\begin{cases} \text{Minimize} & \frac{1}{2} w^T w + C \sum_i \xi_i \\ \text{subject to} & \xi_i \geq 0,\ y^{(i)}(w.(x_1^{(i)} - x_2^{(i)})) \geq 1 - \xi_i,\ i = 1, \cdots, p \end{cases}$$

et fournit une solution optimale w^*

Remarques sur Ranking SVM

- Propriété : $d \succ_{pert-q} d'$ ssi $\text{sgn}(w^*, \overrightarrow{(d,q)} - \overrightarrow{(d',q)})$ positif

  Cette utilisation est toutefois coûteuse. On utilise en fait en pratique directement le score « svm » :

$$RSV(q,d) = (w^*.\overrightarrow{(q,d)})$$

- Pas de différence entre des erreurs faites en tête et en milieu de liste
- Les requêtes avec plus de documents pertinents ont un plus grand impact sur w^*

RSVM-IR (1)

Idée : modifier le problème d'optimisation à la base de *Ranking SVM* (RSVM) pour tenir compte des rangs des documents considérés ($\tau_{k(i)}$) et du type de la requête ($\mu_{q(i)}$)

$$\begin{cases} \text{Minimize} & \frac{1}{2} w^T w + C \sum_i \tau_{k(i)} \mu_{q(i)} \xi_i \\ \text{subject to} & \xi_i \geq 0,\ y^{(i)}(w.(x_1^{(i)} - x_2^{(i)})) \geq 1 - \xi_i,\ i = 1, \cdots, p \end{cases}$$

où $q(i)$ est la requête associée au $i$ième exemple, et $k(i)$ est le type de rangs associés aux documents du $i$ième exemple

RSVM-IR (2)

En pratique :

- $\mu_{q(i)} = \frac{\max_j \#\{\text{instance pairs associated with } q(j)\}}{\#\{\text{instance pairs associated with } q(i)\}}$
- Pour chaque requête, on établit son ordre optimal (ensemble d'apprentissage) ; on sélectionne ensuite aléatoirement un document pour chaque rang, et on inverse leur position ; ce nouveau ordonnancement induit une baisse de NDCG (ou d'une autre mesure), que l'on moyenne sur toutes les requêtes pour obtenir $\tau_{k(i)}$

RSVM-IR (3)

- En pratique, on utilise directement le *w* appris (tout comme dans RSVM)
- Les résultats obtenus par RSVM-IR sur des collections standard sont très prometteurs (entraînement à partir des données “campagne d’évaluation”), pour l’instant meilleurs que ceux des modèles probabilistes standard (ce qui n’est pas forcément le cas de RSVM ou des approches par catégorisation binaire)
- Approche *pairwise* vs *pointwise* : retour sur la notion de valeur absolue de pertinence

Extensions des approches précédentes

Approche *listwise*

- Traiter directement les listes triées comme des exemples d'apprentissage
- Deux grands types d'approche
  - Fonction objectif liée aux mesures d'évaluation
  - Fonction objectif définie sur des listes de documents
- Mais les mesures d'évaluation sont en général non continues

Constituer des données d'apprentissage

- On dispose de données annotées pour plusieurs collections
  - TREC (TREC-vidéo)
  - CLEF
  - NTCIR
  - Letor
  - Challenge Yahoo !

- Pour les entreprises (intranets), de telles données n'exitent pas en général $\rightarrow$ modèles standard, parfois faiblement supervisés

- Qu'en est-il du web ?

Données d'apprentissages sur le web

- Une source importante d'information : les clics des utilisateurs
  - Utiliser les clics pour inférer des préférences entre documents (paires de préférence)
  - Compléter éventuellement par le temps passé sur le résumé d'un document (*eye-tracking*)
- Que peut-on déduire des clics ?

Exploiter les clics (1)

Les clics ne fournissent pas des jugements de pertinence absolus, mais relatifs. Soit un ordre $(d_1, d_2, d_3, \cdots)$ et C l'ensemble des documents cliqués. Les stratégies suivantes peuvent être utilisées pour construre un ordre de pertinence entre documents :

1. Si $d_i \in C$ et $d_j \notin C$, $d_i \succ_{pert-q} d_j$
2. Si d_i est le dernier doc cliqué, $\forall j < i$, $d_j \notin C$, $d_i \succ_{pert-q} d_j$
3. $\forall i \geq 2, d_i \in C, d_{i-1} \notin C, d_i \succ_{pert-q} d_{i-1}$
4. $\forall i, d_i \in C, d_{i+1} \notin C, d_i \succ_{pert-q} d_{i+1}$

Exploiter les clics (2)

- Ces différentes stratégies permettent d'inférer un ordre partiel entre documents
- La collecte de ces données fournit un ensemble d'apprentissage très large, sur lequel on peut déployer les techniques vues précédemment
- La RI sur le web est en partie caractérisée par une course aux données :
  - Indexer le maximum de pages
  - Récupérer le maximum de données de clics

Conclusion - Apprentissage et RI

- Des approches qui tentent d'exploiter toutes les informations à disposition (700 attributs pour le challenge Yahoo ! par exemple, y compris les scores des modèles *ad hoc*)
- Des approches qui s'intéressent directement à ordonner les documents (*pairwise*, *listwise*)
- Beaucoup de propositions (réseaux neuronaux (Bing), *boosting*, méthodes à ensemble)
- Recherche actuelle se concentre sur *listwise* (et l'optimisation de fonctions proches des mesures d'évaluation - *NDCG_Boost*)

Quelques Références

M. Amini et E. Gaussier
Modèles et Algorithmes en Recherche d'Information et ses Applications
À paraître chez Eyrolles, 2013

Manning et al.
Introduction to Information Retrieval
Cambridge University Press 2008
www-csli.stanford.edu/~hinrich/information-retrieval-book.html

Yue et al.
A Support Vector Method for Optimizing Average Precision,
SIGIR 2007

V. Vapnik.
The nature of statistical learning theory.
Springer, Verlag, 1998.