

---

# Apprentissage par Renforcement dans un système de filtrage adaptatif

M. Boughanem\* , H.Tebri\* , M.Tmar\*\*

\* IRT-SIG

Université Paul Sabatier de Toulouse  
118, route de Narbonne  
F-31062 Toulouse Cedex 4  
{bougha,tebri}@irit.fr

\*\* ISIMS, Université de Sfax  
M'Harza 3000 Sfax (Tunisie)  
mohamedtmar@yahoo.fr

---

*RÉSUMÉ.* Cet article présente une méthode incrémentale d'apprentissage des profils dans les systèmes de filtrage d'information. Cette méthode est basée sur le principe de renforcement. L'idée de base consiste à construire, à chaque arrivée d'un document pertinent, un profil " provisoire " permettant de sélectionner le document en question avec un score " fort ", puis intégrer ce profil, grâce à une descente de gradient, dans le profil global. Cette méthode est comparée à une version incrémentale de l'algorithme de Rocchio adapté au filtrage d'information. Des expérimentations effectuées sur la collection de test de référence TREC<sup>1</sup>, montrent que les résultats obtenus par la méthode de renforcement sont plus intéressants que ceux obtenus par l'algorithme de Rocchio.

*ABSTRACT.* This paper presents a profile learning method in the information filtering systems. This method is based on a reinforcement process. The basic idea consists to build, when a document is judged as relevant, the provisional profile which makes it possible to find this document with a strong score, then integrate this profile by using the gradient propagation formula in the global profile. The reinforcement method is compared to Rocchio learning algorithm which is adapted to information filtering. Experiments carried out on TREC-2002 collection showed the effectiveness of the reinforcement process.

*MOTS-CLÉS :* filtrage adaptatif, profil, apprentissage par renforcement, algorithme de Rocchio.

*KEYWORDS:* Adaptive filtering, profile, reinforcement learning, Rocchio algorithm.

---

## 1. Introduction

Le filtrage d'information est un processus permettant, à partir d'un flot de documents provenant d'une source dynamique d'information (Internet, News, E-mail, etc.), de sélectionner ceux qui satisfont les besoins relativement stables des utilisateurs, appelés *profils* [BEL 92].

Compte tenu de la dualité entre la Recherche d'information (RI) et le Filtrage d'Information (FI), comme le montre Belkin dans [BEL 92], bon nombre de modèles de filtrage d'information sont basés sur des modèles de recherche d'information augmentés par une fonction de décision, le plus souvent de type seuil. D'une façon générale, les documents et les profils sont représentés par des listes de mots pondérés. Le filtrage d'information revient à comparer chaque document, qui arrive dans le système aux différents profils. Ceci consiste à mesurer un score de similarité entre le document et le profil. Si le score est supérieur au seuil le document est accepté sinon il est rejeté. Or, la problématique majeure en FI vient du fait qu'en l'absence de collection de référence, la détermination de ce seuil et des pondérations adéquates associées aux profils et aux documents est pratiquement impossible. Car, dans un système de filtrage d'information (SFI), au démarrage du processus de filtrage on ne dispose d'aucune connaissance sur les documents à filtrer.

La solution adoptée, dans la majorité des travaux actuels, consiste à démarrer le processus de filtrage en initialisant le profil avec une liste de mots clés extraits du texte du profil et le seuil à une valeur arbitraire, puis adapter et apprendre le seuil et le profil au fur et à mesure que les documents arrivent. Cette approche est appelée filtrage incrémental ou *Adaptive filtering* dans la terminologie TREC [VOO 01].

La majorité des techniques d'adaptation de profil proposées, sont inspirées du principe de reformulation de requêtes. Les techniques utilisées sont principalement basées sur une version incrémentale de l'algorithme de Rocchio [ROC 71], on y trouve notamment les travaux de [CAL 98] [SCH 98], ou des techniques basées sur les classifieurs Bayésiens [KIM 00], les réseaux de neurones [KWO 01] et des techniques génétiques [BOU 99].

Concernant le seuillage, les différentes méthodes proposées tentent de définir un seuil qui permet d'optimiser une fonction d'utilité. Une fonction d'utilité permet de mesurer la capacité d'un système de FI à ne sélectionner que des documents pertinents [VOO 01].

Nous nous intéressons dans cet article à l'apprentissage du profil dans un cadre purement incrémental. Contrairement à ce qui se fait dans les autres travaux, qui souvent utilisent des collections d'apprentissage ou effectuent l'apprentissage sur un lot de documents, nous proposons une méthode incrémentale pour l'apprentissage de profils, appelée apprentissage par renforcement. Pour évaluer l'intérêt de cette méthode, nous la comparons à l'un des algorithmes les plus utilisés dans le domaine de la RI, l'algorithme de Rocchio [ROC 71].

Cet article est organisé comme suit : la section 2 décrit l'algorithme de reformulation de requêtes développé par Rocchio. Dans la section 3, nous présentons notre modèle de base du système de filtrage d'information et la méthode d'apprentissage des profils basée sur le principe de renforcement. Enfin la section 4, est consacrée aux expérimentations et aux résultats obtenus par les deux méthodes. Les expérimentations sont effectuées sur une collection de test de référence TREC-2002 [VOO 02].

## 2. Algorithme de Rocchio

L'algorithme de reformulation de requêtes développé par Rocchio au milieu des années 60 [ROC 66] [ROC 71], est l'un des algorithmes d'apprentissage les plus utilisés dans le domaine de la RI. Il permet de construire une requête performante à partir de la requête initiale et d'un ensemble de documents jugés pertinents et non pertinents. La forme standard de l'algorithme de Rocchio est donnée comme suit :

$$\vec{Q}_{nllc} = \alpha \vec{Q}_{init} + \beta \frac{1}{R} \sum_{d \in E_r} \vec{d} - \gamma \frac{1}{N - R} \sum_{d \notin E_{nr}} \vec{d} \quad (1)$$

Où :

$\vec{Q}_{init}$  : le vecteur de la requête initiale, représenté par une liste de termes pondérés,

$\vec{Q}_{nllc}$  : le vecteur de la nouvelle requête,

$E_r$  (resp.  $E_{nr}$ ) : l'ensemble des documents jugés pertinents (resp. non pertinents),

$R = |E_r|$  : le nombre de documents pertinents,

$N$  : le nombre de documents total (avec  $N = |E_r| + |E_{nr}|$ ),

$\alpha, \beta, \gamma$  : des paramètres constants, ils permettent de contrôler respectivement la contribution de la requête initiale, des documents pertinents et non pertinents dans le processus de construction de la nouvelle requête.

Une version incrémentale de l'algorithme de Rocchio a été utilisée pour l'apprentissage de profils. On y trouve notamment les travaux de [CAL 98] [SCH 98], Query Zoning (ou QZ) développé par Singhal et al. [SIN 97] et Dynamic Query Zoning (ou DQZ) [BUC 95].

Dans nos différentes expérimentations, nous comparons notre méthode d'apprentissage à une version incrémentale de l'algorithme de Rocchio tel qu'il est utilisé dans " Query Zoning ". La version incrémentale de QZ consiste à construire les deux ensembles,  $E_r$  et  $E_{nr}$ , à partir des documents sélectionnés et jugés pertinents par l'utilisateur à l'instant  $t$ . Le processus de d'apprentissage (construction du nouveau profil) est déclenché à chaque arrivée d'un document pertinent.

## 3. Apprentissage par renforcement

Avant de commencer à présenter la méthode d'apprentissage par renforcement, nous décrirons tout d'abord le modèle de filtrage de base utilisé dans cet article.

### 3.1. le modèle de filtrage de base

Le modèle de filtrage que nous utilisons est basé sur une approche vectorielle. Les documents et les profils sont représentés sous forme d'une liste de termes pondérés.

Un profil  $p^{(t)}$  est représenté par un ensemble de termes sans les mots vides. Il est donné sous une forme vectorielle, où à chaque terme  $tp_i$  lui est associé un poids  $w_i^{(t)}$ ,  $t$  représente l'instant où le système reçoit un document. Ce profil est appelé le profil global de l'utilisateur.

Initialement, les termes du profil peuvent être saisis par un utilisateur ou extraits à partir d'un ensemble de documents représentant le centre d'intérêt de l'utilisateur. Le poids du terme dans le profil à l'étape initiale est calculé comme suit :  $w_i^{(0)} = tfp_i * (\max_j(tfp_j))^{-1}$ , où  $tfp_i$  est la fréquence du terme  $tp_i$  dans le profil. Ce poids sera ajusté par apprentissage à chaque fois un document est sélectionné et jugé pertinent.

A chaque arrivée d'un document  $d_i^{(t)}$ , celui-ci est indexé. Le résultat de cette opération est une liste de termes pondérés. Le poids de chaque terme  $d_{i_k}^{(t)}$  dans le document  $d_i^{(t)}$  est calculé par une fonction de pondération utilisée dans le système de recherche d'information **Mercure** [BOU 00] :

$$d_{i_k}^{(t)} = \frac{tf_k^{(t)}}{h_3 + h_4 \frac{dl^{(t)}}{\Delta l^{(t)}} + tf_k^{(t)}} \log\left(\frac{N^{(t)}}{n_k^{(t)}} + 1\right) \quad (2)$$

Où :

$tf_k^{(t)}$  : fréquence du terme  $t_k$  dans le document  $d_i^{(t)}$ , à l'instant  $t$ ,

$h_3, h_4$  : paramètres constants, dans les expérimentations  $h_3 = 0.2$  et  $h_4 = 0.7$ ,

$dl^{(t)}$  : nombre de termes dans le document  $d_i^{(t)}$ ,

$\Delta l^{(t)}$  : longueur moyenne des documents,

$N^{(t)}$  : nombre de documents examinés jusqu'à l'instant  $t$ ,

$n_k^{(t)}$  : nombre de documents parmi  $N^{(t)}$  contenant le terme  $t_k$ .

Le filtrage consiste à mesurer un score, noté  $rsv(d_i^{(t)}, p^{(t)})$ , entre le document et le profil. Ce score est défini par le produit scalaire entre le document  $d_i^{(t)}$  et le profil  $p^{(t)}$ . Ensuite, ce score est comparé à un seuil ( $\theta$ ) de filtrage, pour décider si le document est accepté ou non : si  $rsv(d_i^{(t)}, p^{(t)}) \geq \theta^{(t)}$  alors le document  $d_i^{(t)}$  est sélectionné, sinon il est rejeté. Le profil et les statistiques liées à la pondération des termes des documents sont appris à chaque arrivée d'un document pertinent.

Nous présentons dans la section suivante la méthode d'apprentissage par renforcement. Nous ne détaillons pas la technique d'adaptation du seuil, les lecteurs intéressés peuvent se référer à [Tmar 02].

### 3.2. Apprentissage des profils par renforcement

L'apprentissage des profils que nous utilisons est basé sur un principe de renforcement [SUT 98]. A cet effet, nous considérons que quand un document  $d_i^{(t)}$  est sélectionné et jugé pertinent, il faut trouver une représentation du profil  $p_x^{(t)}$  qui permet de retrouver ce document avec un score " fort ", soit  $\lambda$ , puis intégrer cette solution pour apprendre le profil global de l'utilisateur. Ceci revient donc à trouver le profil tel que  $rsv(d_i^{(t)}, p_x^{(t)}) = \lambda$ . Autrement dit, il faut chercher les  $pw_j^{(t)}$ , poids du terme  $tp_j$  dans le profil  $p_x^{(t)}$ , qui satisfont l'équation suivante :

$$\sum_{t_k \in d_i^{(t)}, tp_j \in p_x^{(t)}, t_k = tp_j} d_{i_k}^{(t)} \cdot pw_j^{(t)} = \lambda \quad (3)$$

Cette équation admet évidemment une infinité de solutions. Pour pallier ce problème, nous proposons d'ajouter une contrainte pour réduire le nombre de solutions et donc arriver à une solution unique.

Avant de donner cette contrainte, nous précisons la notion du profil idéal et du poids idéal.

**Définition 1** Nous appelons profil idéal à l'instant  $t$ , le profil qui permet de sélectionner tous les documents pertinents et que les documents pertinents.

**Définition 2** Le poids idéal d'un terme est son poids dans le profil idéal.

La contrainte à intégrer est la suivante : compte tenu de l'aspect incrémental de l'apprentissage, nous considérons que les termes du profil solution de l'équation [3] doivent contribuer, à chaque instant, de manière proportionnelle à leur importance réelle dans le profil idéal. Ceci peut se traduire formellement comme suit : supposons que le poids idéal d'un terme  $tp_i$  est donné par  $f_i^{(t)}$ , la contrainte ci-dessus peut alors s'écrire :  $pw_i^{(t)} / f_i^{(t)}$  est une constante. Le système à résoudre devient alors :

$$\begin{cases} \sum_{t_k \in d_i^{(t)}, tp_j \in p_x^{(t)}, t_k = tp_j} d_{i_k}^{(t)} \cdot pw_j^{(t)} = \lambda \\ \forall (t_i, t_j) \in d_i^{(t)2}, \frac{pw_i^{(t)}}{f_i^{(t)}} = \frac{pw_j^{(t)}}{f_j^{(t)}} \end{cases} \quad (4)$$

La solution du système 4 est l'ensemble des poids du profil  $p_x^{(t)}$  qui permet de retrouver le document  $d_i^{(t)}$ .















