Séparateurs à Vaste Marge pondérés en norme l2 pour la sélection de variables en apprentissage d'ordonnancement

Léa Laporte, Sébastien Déjean, Josiane Mothe

Séparateurs à Vaste Marge pondérés en norme ℓ_2 pour la sélection de variables en apprentissage d'ordonnancement

Léa Laporte* — **Sébastien Déjean**** — **Josiane Mothe *****

* *IRIT, UMR 5505, INSA Toulouse, Université de Toulouse, France*
** *IMT, UMR 5219, Université Paul Sabatier, Université de Toulouse, France*
*** *IRIT, UMR 5505, ESPE Midi-Pyrénées, Université de Toulouse, France*

RÉSUMÉ. *Les algorithmes d'apprentissage d'ordonnancement utilisent un très grand nombre de caractéristiques pour apprendre les fonctions d'ordonnancement, entraînant une augmentation des temps d'exécution et du nombre de caractéristiques redondantes ou bruitées. La sélection de variables est une méthode prometteuse pour résoudre ces enjeux. Dans cet article, nous proposons de nouvelles méthodes de sélection de variables en apprentissage d'ordonnancement basées sur des approches de pondération des SVM en norme ℓ_2. Nous proposons une adaptation d'une méthode ℓ_2-AROM pour la résolution des SVM en norme ℓ_0 et un algorithme générique de pondération de la norme ℓ_2 qui résout les problèmes en norme ℓ_0 et ℓ_1. Nos expérimentations montrent que les méthodes proposées sont jusqu'à 7 fois plus rapides et 10 fois plus parcimonieuses que l'état de l'art, pour des qualités d'ordonnancement équivalentes.*

ABSTRACT. *Learning to rank algorithms are dealing with a very large amount of features to automatically learn ranking functions, which leads to an increase of both the computational cost and the number of noisy redundant features. Feature selection is seen as a promising way to address these issues. In this paper, we propose new feature selection algorithms for learning to rank based on reweighted ℓ_2 SVM approaches. We investigate a ℓ_2-AROM algorithm to solve the ℓ_0 norm optimization problem and a generic ℓ_2-reweighted algorithm to approximate ℓ_0 et ℓ_1 norm SVM problems with ℓ_2 norm SVM. Experiments show that our algorithms are up to 10 times faster and use up to 7 times less features than state-of-the-art methods, without lowering the ranking performance.*

MOTS-CLÉS : *Apprentissage d'ordonnancement, Sélection de variables, SVM parcimonieux, Algorithmes de pondération.*

KEYWORDS: *Learning to rank, Feature selection, SVM, Reweighted algorithms*

1. Introduction

L'apprentissage d'ordonnancement, ou *learning-to-rank*, a pour objectif l'optimisation des fonctions d'ordonnancement utilisées par les systèmes de recherche d'information (RI) pour classer les documents suite à une requête. Les algorithmes d'apprentissage d'ordonnancement utilisent des jeux de données composés de couples requête-document associés à des jugements de pertinence pour apprendre la fonction qui optimise le classement. Cette dernière prédit l'ordre optimal des documents à restituer lors de requêtes ultérieures.

Les travaux existants se concentrent majoritairement sur la proposition d'algorithmes d'apprentissage d'ordonnancement (Burges *et al.*, 2005 ; Cao *et al.*, 2007 ; Chapelle et Keerthi, 2010 ; Freund *et al.*, 2003 ; Joachims, 2002 ; Xu et Li, 2007). Alors que le nombre de caractéristiques combinées par les fonctions d'ordonnancement ne cesse d'augmenter (de quelques dizaines à plusieurs milliers), deux problématiques majeures apparaissent. Premièrement, l'augmentation du nombre de caractéristiques affecte les temps d'exécution des algorithmes et de réponse des systèmes, ceux-ci devenant chronophages. Deuxièmement, des caractéristiques non pertinentes ou bruitées peuvent être présentes dans les données, rendant les fonctions apprises non optimales. La sélection de variables, qui contrôle les caractéristiques utilisées par les modèles, est un moyen prometteur de résoudre ces problèmes (Geng *et al.*, 2007).

Trois types d'approches ont été proposés en sélection de variables pour l'apprentissage d'ordonnancement. Les méthodes par filtre considèrent la sélection comme une étape de pré-traitement. Le sous-ensemble de caractéristiques pertinentes est déterminé avant la phase d'apprentissage, indépendamment de cette dernière. Les approches encapsulantes sont aussi des étapes de pré-traitement. Elles dépendent de l'algorithme d'apprentissage, qu'elles utilisent pour déterminer le meilleur sous-ensemble de caractéristiques. Les méthodes embarquées effectuent simultanément apprentissage et sélection, cette dernière étant spécifique de l'algorithme d'apprentissage considéré.

Geng *et al.* (Geng *et al.*, 2007) ont été les premiers à proposer une méthode de sélection de variables spécifique à l'apprentissage d'ordonnancement. Leur approche de type filtre, nommée GAS, utilise un algorithme glouton pour sélectionner les caractéristiques les plus pertinentes, en minimisant leur similarité et en maximisant leur importance, calculées via des mesures définies par les auteurs. D'autres travaux ont proposé des versions par filtre et encapsulantes d'un même algorithme. Hua *et al.* (Hua *et al.*, 2010) utilisent l'algorithme des k-moyennes pour créer des groupes de caractéristiques similaires. Les auteurs sélectionnent alors k variables représentatives, une par groupe, qu'ils utilisent pour apprendre la fonction d'ordonnancement. Yu *et al.* (Yu *et al.*, 2009) ont proposé deux méthodes basées sur une adaptation de l'algorithme Relief (Kira et Rendell, 1992).

Certaines études se sont intéressées spécifiquement aux approches encapsulantes. Pan *et al.* (Pan *et al.*, 2009) utilisent les arbres de décision et un algorithme glouton pour effectuer la sélection. Dang *et al.* (Dang et Croft, 2010) adaptent l'algorithme

Best First Search (Kohavi et John, 1997) à l'apprentissage d'ordonnancement, tandis que Pahikkala *et al.* (Pahikkala *et al.*, 2010) considèrent un algorithme glouton basé sur la méthode RankRLS (Pahikkala *et al.*, 2009). Enfin, certains travaux comme ceux de Lai *et al.* (Lai *et al.*, 2011) ont considéré des approches de sélection de type Forward-Backward. Il est globalement difficile de comparer ces différentes approches, car elles ont été évaluées sur des jeux de données différents, sans que le code source ne soit disponible en ligne.

Relativement peu de travaux se sont intéressés aux méthodes embarquées en apprentissage d'ordonnancement. Ces approches considèrent des problèmes d'optimisation régularisés. Ces problèmes cherchent à minimiser une fonction composée d'une part d'un terme mesurant la divergence entre valeur prédite et valeur effective et d'autre part, d'un terme de régularisation qui contrôle la capacité de généralisation du modèle.

Certaines régularisations, dites parcimonieuses, permettent, de part leurs propriétés mathématiques, de sélectionner les caractéristiques les plus pertinentes et d'en apprendre la combinaison optimale. Dans ce cadre, une approche naturelle pour effectuer la sélection est de minimiser la norme ℓ_0, qui représente le nombre d'éléments non nuls d'un vecteur. Néanmoins, cette régularisation étant non convexe, non différentiable et non continue, le problème d'optimisation ne peut être résolu par des algorithmes classiques. La régularisation ℓ_1, ou Lasso (Hastie *et al.*, 2003), proposée pour la sélection de variables en discrimination, lui est généralement préférée. Sun *et al.* (Sun *et al.*, 2009) ont ainsi proposé RSRank, un algorithme parcimonieux qui optimise une mesure de RI via un problème d'optimisation régularisée en norme ℓ_1. Lai *et al.* (Lai *et al.*, 2013a) ont considéré un algorithme primal-dual, nommé FenchelRank, pour la résolution de Séparateurs à Vaste Marge (SVM) régularisés en norme ℓ_1. L'utilisation de la régularisation mixte $\ell_1 - \ell_2$ a également été proposée dans l'algorithme FSMRank (Lai *et al.*, 2013b). Les auteurs de ces deux derniers algorithmes ont montré que ces approches surpassaient les méthodes de référence GAS et RSRank. D'autres travaux considèrent des régularisations non convexes (Laporte *et al.*, accepté).

Dans cet article, nous nous concentrons sur les SVM parcimonieux régularisés, qui sont des techniques de sélection efficaces. Nous proposons d'utiliser des approches de pondération de type moindres carrés pondérés (*Iteratively Reweighted Least Square* ou IRLS). Celles-ci résolvent des problèmes d'optimisation difficiles à résoudre par itérations successives de problèmes plus faciles à résoudre, comme des SVM en norme ℓ_2. Nous proposons trois algorithmes pour effectuer la sélection de variables via des SVM régularisés en norme ℓ_0 ou ℓ_1. Le premier, Rankℓ_2-AROM, considère un problème régularisé en norme ℓ_0 résolu par une approche de type ℓ_2-AROM (Weston *et al.*, 2003). Les deux derniers algorithmes sont des approches de type IRLS, adapté pour l'apprentissage d'ordonnancement. L'un, RankRWFS-ℓ_1, résout le problème en norme ℓ_1 tandis que l'autre, RankRWFS-ℓ_0, considère la régularisation ℓ_0. Tous deux utilisent la même structure, mais des règles de pondération de la norme ℓ_2 différentes. Nos expérimentations sur des jeux de données de référence en apprentissage d'ordonnancement montrent que nos approches utilisent de 2 à 7 fois moins de caractéristiques

que l'état de l'art, tout en conservant la même qualité d'ordonnancement et en étant plus rapides.

La section 2 décrit nos propositions. La section 3 détaille le protocole expérimental. La section 4 présente les résultats. Nous discutons des perspectives en section 5.

2. Sélection de variables via des SVM itérativement pondérés

2.1. *Cadre général*

En apprentissage d'ordonnancement, des algorithmes d'apprentissage sont utilisés pour optimiser le classement de documents ou de pages web. Nous nous plaçons spécifiquement dans le cadre d'approches d'apprentissage d'ordonnancement par paire, dont l'objectif est de prédire des préférences entre documents. Un document i est préféré au document j s'il doit être classé plus haut dans la liste de résultats.

Considérons Q requêtes, $\mathbb{D}$ l'ensemble des n documents associés à la requête q et m le nombre de caractéristiques. Pour une requête q fixée, chaque couple requête document (q, i) est représenté par le vecteur de caractéristiques $\mathbf{x}^i \in \mathbb{R}^m$. Chaque caractéristique représente le score de similarité entre requête et document obtenu grâce à un modèle de RI (BM25, modèle de langue, etc ...). A chaque paire $(\mathbf{x}^i, \mathbf{x}^j)$ est associé un jugement de préférence $y^{(i,j)}$ tel que $y^{(i,j)} = 1$ si i est préféré à j et $y^{(i,j)} = -1$ sinon. L'ensemble des préférences $\mathbb{P}$ associées à la requête est alors construit en utilisant ces jugements. Chaque préférence $p \in \mathbb{P}$ entre les documents i et j est représentée dans l'espace des caractéristiques par le vecteur $\tilde{\mathbf{x}}^p$ tel que $\tilde{\mathbf{x}}^p = (\mathbf{x}^i - \mathbf{x}^j)^\top$, vecteur ligne de $\mathbb{R}^m$.

Le problème d'optimisation des SVM linéaires non parcimonieux est alors défini de la façon suivante (Chapelle et Keerthi, 2010) :

$$\min_{\mathbf{w}} \frac{1}{2}\|\mathbf{w}\|_2^2 + C \sum_{p \in \mathbb{P}} L(\tilde{\mathbf{x}}^p \mathbf{w}) \qquad [1]$$

où $\mathbf{w} \in \mathbb{R}$ est le vecteur colonne des poids affectés aux caractéristiques, $\|\mathbf{w}\|_2^2 = \sum_{i=1}^m \mathbf{w}_i^2$ est le terme de régularisation en norme ℓ_2 et $\sum_{p \in \mathbb{P}} L(\tilde{\mathbf{x}}^p \mathbf{w})$ est le terme d'ajustement aux données, qui mesure l'écart entre valeur prédite et valeur effective. C est une constante positive qui permet de contrôler le compromis entre régularisation et ajustement.

Dans cet article, nous nous concentrons sur des formulations parcimonieuses des SVM. Nous nous intéressons aux SVM régularisés pour lesquels le terme de régularisation en norme ℓ_2 est remplacé soit par la régularisation ℓ_0 ($\|\mathbf{w}\|_0 = \sum_{\mathbf{w}_i \neq 0} 1$), soit par la régularisation ℓ_1 ($\|\mathbf{w}\|_1 = \sum_{i=1}^m |\mathbf{w}_i|$).

Minimiser la norme ℓ_0 revient à minimiser le nombre de caractéristiques intervenant dans le modèle linéaire. Bien qu'il s'agisse d'une façon naturelle de procéder à la sélection de variables, ce problème est en pratique NP-complet. Du fait du caractère

non continu, non différentiable et non convexe de la norme ℓ_0, il est particulièrement difficile à résoudre par les algorithmes usuels d'optimisation. Le problème régularisé en norme ℓ_1 constitue une relaxation convexe du précédent et lui est généralement préféré. La norme ℓ_1, bien que convexe, est non différentiable en zéro et des algorithmes spécifiques doivent être utilisés.

Les problèmes précédents en apprentissage d'ordonnancement par paire sont similaires à ceux rencontrés en discrimination. Chapelle et Keerthi justifient l'utilisation de méthodes spécifiques à l'apprentissage d'ordonnancement pour des raisons de temps de calcul (Chapelle et Keerthi, 2010). Le choix du modèle optimal s'effectue non pas en minimisant le risque quadratique, mais en maximisant une mesure de recherche d'évaluation sur le jeu de données de validation, ce qui justifie la proposition de méthodes spécifiques à l'apprentissage d'ordonnancement (Geng *et al.*, 2007).

Dans cet article, nous nous intéressons aux approches de pondération en norme ℓ_2 pour l'approximation des problèmes parcimonieux. Dans la suite, nous proposons trois algorithmes de sélection de variables spécifiques à l'apprentissage d'ordonnancement : Rank-ℓ_2-AROM, basé sur une approche de type ℓ_2-AROM (Weston *et al.*, 2003) ainsi que RankRWFS-ℓ_1 et RankRWFS-ℓ_0 qui utilisent des règles de pondération spécifiques aux normes ℓ_1 et ℓ_0 respectivement.

2.2. *Rank-ℓ_2-AROM, approximation de la norme ℓ_0 pour l'apprentissage d'ordonnancement*

L'algorithme ℓ_2-AROM (ℓ_2 *Approximation of the zeRO-norm Minimization*) a été initialement proposé dans le cadre de la discrimination (Weston *et al.*, 2003). Il permet d'approcher la solution des SVM régularisés en norme ℓ_0 par itérations successives des SVM pondérés en norme ℓ_2.

A chaque itération, un algorithme standard de résolution des SVM en norme ℓ_2 est utilisé. Le vecteur de poids obtenus, noté $\mathbf{w}^{(t)}$ est utilisé pour pondérer les valeurs des caractéristiques courantes. Ainsi, pour chaque observation $\mathbf{x}^{j(t)}$, la valeur de chaque caractéristique i, notée $\mathbf{x}_i^{j(t)}$, est multiplié par le poids courant, tel que $\mathbf{x}_i^{j(t)} \leftarrow \mathbf{w}_i^{(t)}\mathbf{x}_i^{j(t-1)}$. Le processus est répété jusqu'à convergence, ou, dans le cas de la sélection de variables, jusqu'à ce que le nombre de variables souhaitées soit atteint. Dans ce dernier cas, le sous-ensemble de variables sélectionnées est utilisé pour apprendre les poids finaux via l'algorithme standard.

Nous adaptons cette méthode à l'apprentissage d'ordonnancement en utilisant un solveur spécifique à l'apprentissage d'ordonnancement. Par ailleurs, il est important de noter que la norme ℓ_2 n'étant pas parcimonieuse, il est nécessaire de fixer un seuil de nullité des poids du vecteur $\mathbf{w}$, afin d'éliminer les caractéristiques non pertinentes. L'algorithme ainsi obtenu est nommé Rank-ℓ_2-AROM.

2.3. *RankRWFS : approche de repondération des normes ℓ_1 et ℓ_0 pour la sélection de variables en apprentissage d'ordonnancement*

L'algorithme Rankℓ_2-AROM est conçu pour approcher la solution du problème en norme ℓ_0 uniquement. Nous proposons une approche basée sur les SVM itérativement pondérés qui considère les normes ℓ_0 et ℓ_1, via des règles de pondération spécifiques. Ces travaux sont inspirés de l'algorithme RW pour l'approximation de la régularisation ℓ_1 proposé dans le cadre de la discrimination (Kujala *et al.*, 2009).

2.3.1. *Approximation des normes ℓ_1 et ℓ_0*

Le principe des méthodes d'approximation par pondération du problème en norme ℓ_2, comme par exemple l'algorithme RW, est de ré-écrire les normes parcimonieuses pour faire apparaître la régularisation ℓ_2. Considérons le problème des SVM par paire en norme ℓ_1 présenté en section 2.1. Soit $\mathbb{I} = \{i|\mathbf{w}_i \neq 0\}$, la régularisation ℓ_1 peut s'écrire :

$$\|\mathbf{w}\|_1 = \sum_{i=1}^{m} |\mathbf{w}_i| = \sum_{i\in\mathbb{I}} |\mathbf{w}_i| = \sum_{i\in\mathbb{I}} \frac{\mathbf{w}_i^2}{|\mathbf{w}_i|} \qquad [2]$$

Soit un instant t donné, $b_i^{(t)} = \frac{1}{\sqrt{|\mathbf{w}_i^{(t)}|}}$ et $\mathbf{r}_i^{(t)} = b_i^{(t)}\mathbf{w}_i^{(t)}$, alors la norme ℓ_1 devient :

$$\|\mathbf{w}^{(t)}\|_1 = \sum_{i\in\mathbb{I}} (b_i^{(t)}\mathbf{w}_i^{(t)})^2 = \sum_{i\in\mathbb{I}} (\mathbf{r}_i^{(t)})^2 = \|\mathbf{r}^{(t)}\|_2^2$$

Un changement de variables similaire permet de ré-écrire la fonction de perte. Soit $B^{(t)} = diag(b_i^{(t)})\forall i \in \{1,\ldots,m\}$ la matrice diagonale de $\mathbb{R}^{m\times m}$ qui contient les valeurs $b_i^{(t)}$ sur sa diagonale et zéro ailleurs, tel que $\mathbf{r} = B\mathbf{w}$. En constatant que $\tilde{\mathbf{x}}^p\mathbf{w} = \tilde{\mathbf{x}}^p B^{-1}B\mathbf{w} = \tilde{\mathbf{x}}^p B^{-1}\mathbf{r}$ et en posant $\tilde{\mathbf{z}}^p = \tilde{\mathbf{x}}^p B^{-1}$, le problème en norme ℓ_1 équivaut à un instant t au problème pondéré suivant :

$$\min_{\mathbf{r}} \frac{1}{2}\|\mathbf{r}\|_2^2 + C\sum_{p\in\mathbb{P}} L(\tilde{\mathbf{z}}^p\mathbf{r}) \qquad [3]$$

Un raisonnement similaire peut être mené dans le cadre du problème d'optimisation en norme ℓ_0 (*cf.* section 2.1) en utilisant la ré-écriture suivante :

$$\|\mathbf{w}\|_0 = \sum_{i=1}^{m} \mathbb{1}_{\{\mathbf{w}_i\neq 0\}} \sum_{\mathbf{w}_i\neq 0} 1 = \sum_{\mathbf{w}_i\neq 0} \frac{\mathbf{w}_i^2}{\mathbf{w}_i^2} \qquad [4]$$

et en posant $b_i^{(t)} = \frac{1}{|\mathbf{w}_i^{(t)}|}$.

L'algorithme RW (Kujala *et al.*, 2009) est une implémentation de l'approximation du problème régularisé en norme ℓ_1 par pondération de la norme ℓ_2 dans le cas de la discrimination. A chaque itération t, considérant un vecteur $\mathbf{v}^{(t)} \in \mathbb{R}^m$ tel que

$\mathbf{v}^{(1)} = [1 \dots 1]$ et $\forall t > 1$, $\mathbf{v}^{(t)} = [\prod_{k=1}^{(t-1)} \mathbf{w}_1^{(k)} \dots \prod_{k=1}^{(t-1)} \mathbf{w}_m^{(k)}]$, une observation $\mathbf{x}$ et une caractéristique i, la mise à jour $\mathbf{z}_i^{(t)}$ de $\mathbf{x}_i^{(t)}$ est calculée comme suit :

$$\mathbf{z}_i^{(t)} \leftarrow \sqrt{|\mathbf{w}_i^{(t-1)} \mathbf{v}_i^{(t-1)}|}\mathbf{x}_i \quad [5]$$

Le processus est itéré T fois, où T est le nombre d'itérations fixé par l'utilisateur. A la dernière itération, les vecteurs finaux $\mathbf{w}$ et $\mathbf{v}$ sont multipliés pour obtenir les poids solutions. Les valeurs inférieures à un seuil de nullité donné sont alors annulées. L'algorithme RW n'est pas spécifiquement conçu pour la sélection de variables en apprentissage d'ordonnancement, ni pour la prise en compte de la norme ℓ_0. Nous proposons une approche générique que nous nommons RWFS, qui permet de considérer les problèmes en norme ℓ_0 et ℓ_1 en sélectionnant la règle de mise à jour spécifique à chaque régularisation. Nous nommons ces algorithmes RankRWFS-ℓ_0 et RankRWFS-ℓ_1.

2.3.2. *RankRWFS : repondération en norme ℓ_2 pour la sélection de variables en apprentissage d'ordonnancement*

Notre approche fournit un algorithme générique pour la sélection de variables et l'apprentissage du modèle final, pouvant utiliser aussi bien la norme ℓ_0 que la ℓ_1. Cette méthode est basée sur trois points clés :

1) Le choix de la règle de mise à jour adéquate pour chaque régularisation parcimonieuse (ℓ_0 ou ℓ_1) pour la résolution itérative par pondération de la norme ℓ_2 ;

2) La définition d'une structure de contrôle du nombre de variables à atteindre et

3) L'adaptation à l'apprentissage d'ordonnancement.

Règles de mises à jour : Nous utilisons les ré-écritures présentées respectivement aux équations 2 et 4. Dans le cas de la norme ℓ_1, nous considérons la règle de mise à jour définie à l'équation 5, tandis que la résolution de la norme ℓ_0 fait appel à la règle de pondération suivante :

$$\mathbf{z}_i^{(t)} \leftarrow |\mathbf{w}_i^{(t-1)} \mathbf{v}_i^{(t-1)}|\mathbf{x}_i \quad [6]$$

Un seuil de nullité est utilisé pour fixer les poids à zéro.

Contrôle du nombre de caractéristiques : Comme indiqué dans (Weston *et al.*, 2003), les approches de type ℓ_2-AROM ou RW, qui suppriment des caractéristiques pour approcher une régularisation parcimonieuse, peuvent aller trop loin dans le processus de sélection et dégrader la capacité de généralisation du modèle. Une solution pour remédier à ce problème est de fixer un seuil maximal de variables à conserver r, l'algorithme stoppant la sélection dès que la contrainte est atteinte. Nous avons retenu cette approche dans nos algorithmes.

Adaptation à l'apprentissage d'ordonnancement : A chaque itération t, nous résolvons le problème pondéré en norme ℓ_2 à l'aide d'un algorithme spécifique à l'apprentissage d'ordonnancement. A la fin de l'étape de sélection, un dernier apprentissage est réalisé avec le sous-ensemble de variables sélectionnées et l'algorithme non

parcimonieux en norme ℓ_2, afin de stabiliser les poids finaux de la fonction d'ordonnancement.

Nous appelons cette méthode générique RankRWFS, dont le détail est donné à l'algorithme 1. Dans une première étape, cette approche sélectionne les variables adéquates en approchant un problème de SVM parcimonieux par pondération d'un problème en norme ℓ_2. A chaque itération t, un vecteur de poids $\mathbf{w}$ est appris en utilisant la règle de pondération spécifique à la régularisation parcimonieuse considérée et l'algorithme d'apprentissage d'ordonnancement en norme ℓ_2. Les caractéristiques dont le poids est inférieur à un seuil fixé sont retirées du modèle. Le processus est répété jusqu'à ce que le nombre de variables restantes soit inférieur ou égal à la valeur souhaitée. Dans une seconde étape, le sous-ensemble de caractéristiques sélectionnées est utilisé par l'algorithme d'apprentissage en norme ℓ_2, afin d'apprendre les poids finaux. Nous étudions deux versions de cette approche, RankRWFS-ℓ_0 et RankRWFS-ℓ_1, qui considèrent respectivement les régularisations ℓ_0 et ℓ_1.

3. Cadre expérimental

3.1. *Données, mesures d'évaluation et références*

Nous évaluons nos algorithmes sur les trois jeux de données Ohsumed, MQ2008 et TD2004[1] issus des collections internationales de référence LETOR[2]. Le nombre de requêtes, de préférences et de caractéristiques est indiqué dans le tableau 1. Le nombre de caractéristiques non nulles est indiqué entre parenthèses. Chaque jeu de données est décomposé en échantillons d'apprentissage, validation et test. Une procédure de validation croisée 5-folds est réalisée (chaque algorithme est évalué sur 5 répétitions correspondant à des échantillons de test, apprentissage et validation différents). Cette structure en cinq répétitions est fournie lors du téléchargement des jeux de données et est utilisée dans l'ensemble des travaux sur ces collections.

Tableau 1. *Composition des jeux de données*

Nom	Caractéristiques	Requêtes	Paires	Préferences
TD2004	64 *(64)*	75	74146	1079810
Ohsumed	45 *(39)*	106	16140	582588
MQ2008	46 *(40)*	784	15211	80925

Nous utilisons deux mesures usuelles en RI pour évaluer la qualité de l'ordonnancement : la moyenne des précisions moyennes (MAP) et le *Normalized Discounted*

1. Basés sur les tâches Million Queries de Trec 2008 et Topic Distillation de Trec 2004

2. Disponibles sur `http://research.microsoft.com/en-us/um/beijing/projects/letor/letor3download.aspx` et `http://research.microsoft.com/en-us/um/beijing/projects/letor/letor4download.aspx`

Algorithme 1 Approche générique RankRWFS (ℓ_0 et ℓ_1)

Entrée : Jeu de données d'apprentissage $(\tilde{\mathbf{x}}^j, \mathbf{y}^j)_{j=1}^m$ et nombre maximal de variables r à conserver
Sortie : Vecteur de poids $\mathbf{w}$

```
Initialisation v = [1...1] ∈ R^d, t = 1
Tant que ||w||_0 > r Faire
  Pour chaque observation x̃^j Faire
    Pour i = 1 → d Faire
      z_i^j ← x̃_i^j v_i^(t)
    Fin Pour
  Fin Pour
  w^(t) ← solution de RankSVM-Primal({(z^j, y^j)}_{j=1}^m)
  Pour i = 1 → d Faire
    if Norme == ℓ1 then
      v_i^(t+1) ← sqrt(|w_i^(t) v_i^(t)|)   %%%%%%%% MISE À JOUR POUR ℓ1
    else if Norme == ℓ0 then
      v_i^(t+1) ← |w_i^(t) v_i^(t)|   %%%%%%%% MISE À JOUR POUR ℓ0
    Fin if
  Fin Pour
  Pour i = 1 → d Faire
    w_i ← w_i^(t) v_i^(t)
  Fin Pour
  t = t + 1
Fin Tant que
I ← {i | w_i ≠ 0}
Pour j = 1 → m Faire
  Pour i ∈ I Faire
    s_i^j ← x_i^j
  Fin Pour
Fin Pour
w ← solution de RankSVM-Primal({(s^j, y^j)}_{j=1}^m)   %% Modèle final
Retourner w
```

Cumulative Gain (NDCG). Nous mesurons la capacité des algorithmes à supprimer un nombre important de variables en calculant les ratios de parcimonie des algorithmes, *i.e.* le pourcentage de caractéristiques restantes dans le modèle final. Certaines caractéristiques des jeux de données Ohsumed et MQ2008 sont nulles pour l'ensemble des requêtes, elles ne sont pas prises en compte pour le calcul des ratios de parcimonie. Nous comparons également les méthodes selon les temps d'exécution.

Nous comparons nos trois algorithmes (Rankℓ_2-AROM, RankRWFS-ℓ_1, RankRWFS-ℓ_0) à trois approches de l'état de l'art : Liblinear-ℓ_1 (Fan *et al.*, 2008), FenchelRank (Lai *et al.*, 2013a) et FSMRank (Lai *et al.*, 2013b). Liblinear-ℓ_1 est une

implémentation des SVM en norme ℓ_1 pour la discrimination disponible en ligne dans le package Liblinear[3]. FenchelRank[4] et FSMRank[5] sont les algorithmes de sélection de variables pour l'apprentissage d'ordonnancement les plus performants. Ces méthodes considèrent des SVM parcimonieux, elles sont donc particulièrement adaptées pour effectuer une comparaison.

3.2. *Protocole expérimental*

Nous avons effectué trois études sur les jeux de données LETOR. Tout d'abord, nous avons analysé conjointement les valeurs de MAP (respectivement de NDCG@10) et les ratios de parcimonie obtenus pour l'ensemble des algorithmes. Ensuite, nous avons comparé conjointement les temps d'exécution et les ratios de parcimonie de nos méthodes et de l'état de l'art. Enfin, nous avons extrait un sous-ensemble de caractéristiques importantes pour l'ordonnancement à partir des modèles obtenus.

Dans nos expérimentations, nous avons considéré des seuils de 50%, 30% et 10% de variables à conserver. Comme expliqué dans la section 2, les algorithmes stoppent les itérations quand le nombre de variables du modèle est inférieur ou égal au seuil spécifié. La valeur du paramètre C choisie est celle qui maximise la MAP sur l'échantillon de validation, pour chaque algorithme et jeu de données. Nous avons fixé le seuil de nullité des variables à 10^{-5}. Nous avons choisi RankSVM-Primal (Chapelle et Keerthi, 2010) comme solveur pour les SVM en norme ℓ_2. Les algorithmes sont implémentés en Matlab/Octave. Les expérimentations ont été réalisées sur un MacBook Pro, utilisant Mac OS X Snow Leopard, un processeur Intel Core 2 Duo cadencé à 2.4 GHz et 4 Go de RAM.

4. Résultats

4.1. *Expérimentations sur les jeux de données LETOR*

4.1.1. *Ratios de parcimonie et qualité d'ordonnancement*

La figure 1 présente conjointement les valeurs de MAP et les ratios de parcimonie obtenus pour les algorithmes que nous proposons et l'état de l'art. L'interprétation des résultats obtenus avec le NDCG étant similaire, nous ne la présentons pas ici.

Les algorithmes de pondération atteignent des ratios de parcimonie plus faibles que ceux demandés. En pratique, plusieurs caractéristiques peuvent être retirées simultanément du modèle, ce qui explique ce comportement. Nous constatons que les valeurs de MAP restent stables pour tous les algorithmes et tous les seuils de sélection (50%, 30%, 10%), à l'exception peut être de RankRWFS-ℓ_0 sur TD2004 au seuil de

3. `http://www.csie.ntu.edu.tw/~cjlin/liblinear/`
4. Code source disponible sur `http://www.scholat.com/~hanjiang`
5. Code source fourni sur demande par les auteurs

10 %. Cette dégradation reste très légère et marginale. Les algorithmes que nous proposons obtiennent ainsi des qualités d'ordonnancement comparables à l'état de l'art sur les jeux de données de référence.

Plus intéressant, nous constatons que nos méthodes permettent d'obtenir des qualités d'ordonnancement équivalentes à l'état de l'art, tout en supprimant beaucoup plus de caractéristiques. Considérons les résultats pour le seuil de 10 %. Sur TD2004, nous observons que les algorithmes de pondération utilisent 6 à 10 fois moins de caractéristiques que les algorithmes de référence sur TD2004, pour apprendre des modèles de qualité d'ordonnancement équivalente. Le constat est identique sur les deux autres jeux de données. Comparativement aux algorithmes de référence, les approches que nous proposons suppriment 3 à 7 fois plus de caractéristiques sur Ohsumed et 4 à 6 fois plus sur MQ2008.

Les algorithmes de pondération sont donc beaucoup plus efficaces que l'état de l'art pour effectuer la sélection de variables. En effet, elles obtiennent des qualités d'ordonnancement similaires, tout en supprimant jusqu'à 10 fois plus de caractéristiques. En ce sens, elles sont bien plus performantes que les approches de sélection de variables existantes en discrimination et en apprentissage d'ordonnancement.

4.1.2. *Temps d'exécution et ratios de parcimonie*

La figure 2 présente conjointement les temps d'exécution et les ratios de parcimonie obtenus pour les algorithmes que nous proposons et l'état de l'art. Notons que nous ne fournissons pas les valeurs pour FenchelRank, qui est le seul algorithme à ne pas être implémenté en Matlab.

Nous remarquons que les algorithmes de pondération que nous proposons sont plus performants que les méthodes de l'état de l'art. En effet, lorsque nous considérons les plus petits seuils de parcimonie, nos approches sont soit plus parcimonieuses, soit plus parcimonieuses et plus rapides. Sur TD2004, Rankℓ_2-AROM, RankRWFS-ℓ_1 et RankRWFS-ℓ_0 sont respectivement 2.5, 6 et 7 fois plus rapides que Liblinear-ℓ_1, tandis que RankRWFS-ℓ_1 et RankRWFS-ℓ_0 sont respectivement 2 et 3 fois plus rapides que FSMRank. Sur Ohsumed, les méthodes que nous proposons sont en moyenne 4 fois plus rapides que FSMRank et 2 fois plus rapides que Liblinear-ℓ_1. Enfin, sur MQ2008, nos algorithmes sont 2 à 4 fois plus rapides que Liblinear-ℓ_1. Elles sont plus lentes que FSMRank, mais sélectionnent environ 75% de caractéristiques en moins. Parmi les algorithmes de pondération que nous proposons, RankRWFS-ℓ_0 est le plus rapide sur tous les jeux de données.

Les méthodes de sélection par pondération de la norme ℓ_2 que nous proposons sont donc plus performantes que les approches de l'état de l'art en apprentissage d'ordonnancement et en discrimination, puisqu'elles conservent une qualité d'ordonnancement similaire, tout en étant plus parcimonieuses et plus rapides.

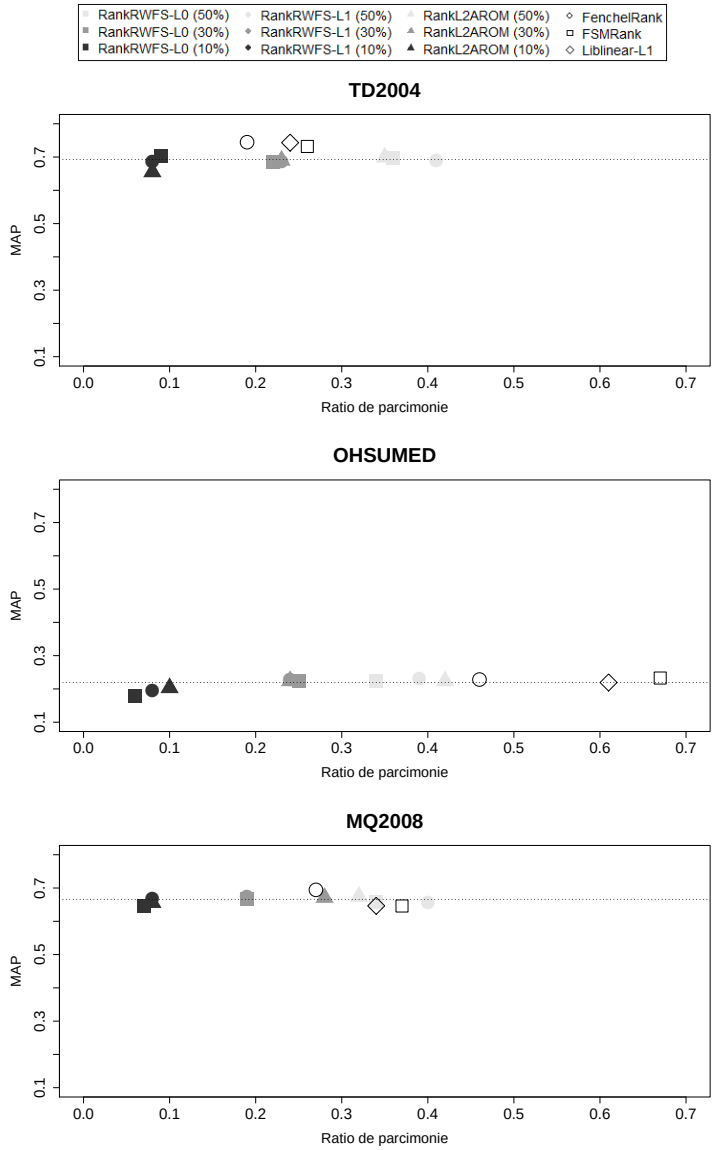

Figure 1. *MAP vs. ratios de parcimonie pour chaque algorithme sur les trois jeux de données de référence. La ligne pointillée représente la valeur de MAP moyenne pour l'ensemble des méthodes. Les algorithmes situés à gauche sont les plus parcimonieux. Les algorithmes que nous proposons sont équivalents en matière de MAP et meilleurs en matière de parcimonie.*

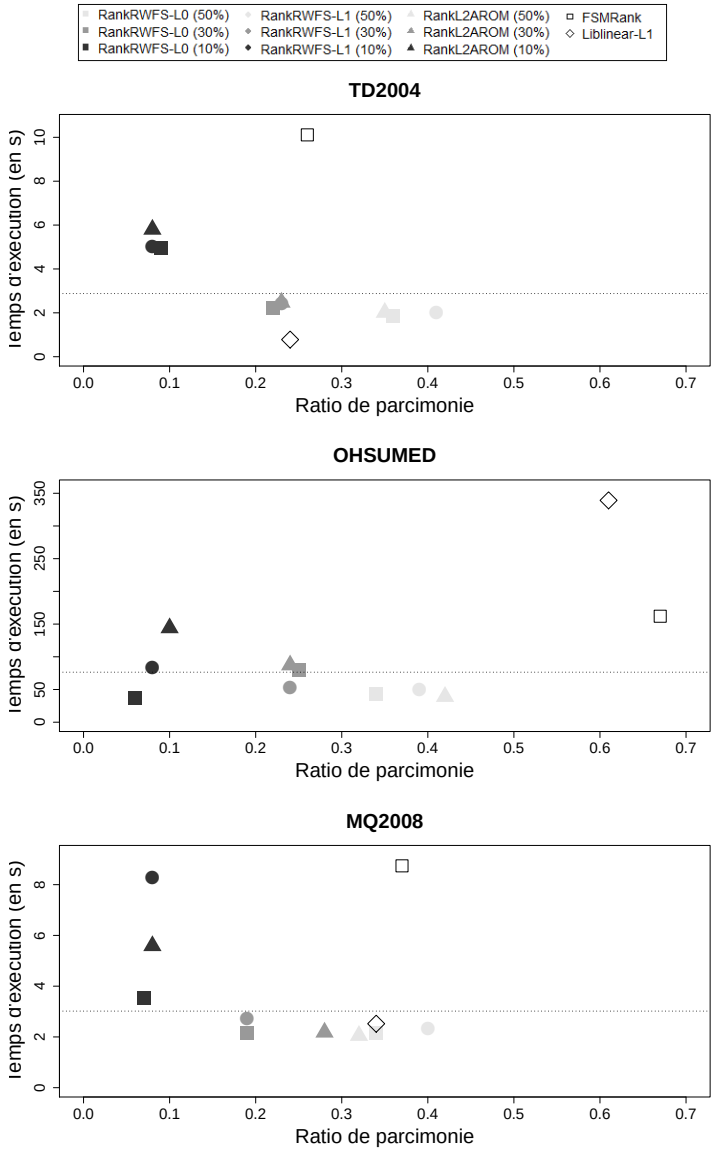

Figure 2. *Temps d'exécution vs. ratios de parcimonie pour chaque algorithme sur les trois jeux de données de référence. La ligne pointillée représente le temps d'exécution moyen pour l'ensemble des méthodes. Les algorithmes situés à gauche et en bas sont les plus parcimonieux et les plus rapides. Les algorithmes que nous proposons sont meilleurs en matière de parcimonie et de rapidité d'exécution.*

4.1.3. *Caractéristiques sélectionnées*

Dans cette étude, nous nous sommes intéressés aux caractéristiques sélectionnées par les modèles, dans le but d'extraire un ensemble de critères importants pour l'ordonnancement. Nous considérons les modèles appris au seuil de 10 %.

Les algorithmes de pondération sélectionnent globalement les mêmes caractéristiques pour un même jeu de données. Ils se comportent donc de façon cohérente. Sur MQ2008, les algorithmes ont sélectionné les caractéristiques 23 et 39 pour l'ensemble des répétitions et ainsi que les critères 32 et 37 pour la majorité des modèles. Toutes ces caractéristiques correspondent à des modèles de langue, connus en RI pour être très informatifs. Sur Ohsumed, les caractéristiques 3, 4 (basées sur la fréquence des termes), 11, 41 (BM25 sur le titre et le document complet) et 43 (modèle de langue) sont les plus fréquemment sélectionnées. Enfin, sur TD2004, les algorithmes sélectionnent une combinaison des caractéristiques 22 (BM25 de l'ancre), 23 (BM25 du titre), 46 (hyperlink based feature propagation) et 52 (HostRank). Toutes ces mesures sont connues pour être hautement informatives.

Les algorithmes proposés dans ces travaux sont donc capables d'apprendre des fonctions d'ordonnancement facilement interprétables, cohérentes et de bonne qualité.

5. Conclusion

Dans cet article, nous nous sommes intéressés à l'adaptation et à l'analyse des méthodes de pondération des SVM en norme ℓ_2 pour la sélection de variables via des SVM parcimonieux en norme ℓ_1 et ℓ_0. A notre connaissance, il s'agit des premiers travaux à proposer l'utilisation de la norme ℓ_0 et des méthodes de pondération de la norme ℓ_2 en sélection de variables pour l'apprentissage d'ordonnancement.

Nos expérimentations ont montré que :

1) Les méthodes de pondération de la norme ℓ_2 que nous proposons sont plus efficaces que l'état de l'art pour supprimer un grand nombre de caractéristiques des fonctions d'ordonnancement. Elles conservent une qualité d'ordonnancement équivalente aux méthodes de référence, tout en utilisant de 3 à 10 fois moins de caractéristiques.

2) Les méthodes proposées obtiennent des qualités d'ordonnancement comparables à l'état de l'art, tout en étant de 2 à 7 fois plus rapides.

3) Les algorithmes que nous proposons permettent d'extraire un sous-ensemble de caractéristiques très informatives pour chaque jeu de données. Ces sous-ensembles sont cohérents.

Les méthodes de pondération de la norme ℓ_2 que nous proposons constituent des approches performantes en sélection de variables pour l'apprentissage d'ordonnancement. Les trois algorithmes Rankℓ_2-AROM, RankRWFS-ℓ_1 et RankRWFS-ℓ_0 présentent des performances globalement similaires, bien que RankRWFS-ℓ_0 semble plus rapide. Par ailleurs, l'approche générique RankRWFS est plus flexible, puisqu'elle

permet d'utiliser différentes régularisations parcimonieuses par une modification de la règle de mise à jour. Notons également que n'importe quel algorithme de résolution des SVM en norme ℓ_2 peut être incorporé au sein de ces méthodes. D'autres ré-écritures des normes (Chartrand et Yin, 2008 ; Zhang et Kingsbury, 2010) pourraient également être utilisées dans le cadre de cette approche. Leur apport fera l'objet de travaux futurs. Nous prévoyons également d'intégrer de nouvelles ré-écritures pour approcher la norme ℓ_0 par pondération de SVM en norme ℓ_1. L'utilisation de cette régularisation parcimonieuse nous permettrait de ne pas avoir à fixer de seuil de nullité des variables, qui est actuellement choisi par défaut.

Remerciements

Les auteurs remercient l'entreprise Nomao ainsi que la Région Midi-Pyrénées, qui ont contribué au financement de ces travaux (financement 10009018).

6. Bibliographie

Burges C., Shaked T., Renshaw E., Lazier A., Deeds M., Hamilton N., Hullender G., « Learning to rank using gradient descent », *Proceedings of the 22nd International Conference on Machine Learning*, ICML'05, p. 89-96, 2005.

Cao Z., Qin T., Liu T.-Y., Tsai M.-F., Li H., « Learning to rank : from pairwise approach to listwise approach », *Proeedings of the 24th International Conference on Machine learning*, ICML'07, p. 129-136, 2007.

Chapelle O., Keerthi S. S., « Efficient algorithms for ranking with SVMs », *Information Retrieval*, vol. 13, n° 3, p. 201-215, jun, 2010.

Chartrand R., Yin W., « Iteratively reweighted algorithms for compressive sensing », *ICASSP*, p. 3869-3872, 2008.

Dang V., Croft B., « Feature Selection for Document Ranking using Best First Search and Coordinate Ascent », *SIGIR Workshop on Feature Generation and Selection for Information Retrieval*, 2010.

Fan R.-E., Chang K.-W., Hsieh C.-J., Wang X.-R., Lin C.-J., « LIBLINEAR : A Library for Large Linear Classification », *Journal of Machine Learning Research*, vol. 9, p. 1871-1874, June, 2008.

Freund Y., Iyer R., Schapire R. E., Singer Y., « An efficient boosting algorithm for combining preferences », *Journal of Machine Learning Research*, vol. 4, p. 933-969, December, 2003.

Geng X., Liu T.-Y., Qin T., Li H., « Feature selection for ranking », *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and development in information retrieval*, SIGIR'07, p. 407-414, 2007.

Hastie T., Tibshirani R., Friedman J. H., *The Elements of Statistical Learning*, corrected edn, Springer, July, 2003.

Hua G., Zhang M., Liu Y., Ma S., Ru L., « Hierarchical feature selection for ranking », *Proceedings of the 19th International Conference on World Wide Web*, WWW'10, p. 1113-1114, 2010.

Joachims T., « Optimizing search engines using clickthrough data », *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data mining*, KDD'02, p. 133-142, 2002.

Kira K., Rendell L. A., « A practical approach to feature selection », *Proceedings of the ninth international workshop on Machine learning*, ML'92, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, p. 249-256, 1992.

Kohavi R., John G. H., « Wrappers for feature subset selection », *Artificial Intelligence*, vol. 97, nº 1-2, p. 273-324, December, 1997.

Kujala J., Aho T., Elomaa T., « A Walk from 2-Norm SVM to 1-Norm SVM », *Proceedings of the 2009 Ninth IEEE International Conference on Data Mining*, ICDM '09, IEEE Computer Society, Washington, DC, USA, p. 836-841, 2009.

Lai H., Pan Y., Liu C., Lin L., Wu J., « Sparse Learning-to-Rank via an Efficient Primal-Dual Algorithm », *IEEE Transaction on Computers*, vol. 62, nº 6, p. 1221-1233, 2013a.

Lai H., Pan Y., Yong T., Yong R., « FSMRank : A Feature Selection Algorithme for Learning-to-Rank », *IEEE Transactions on Neural Networks and Learning Systems*, 2013b.

Lai H., Tang Y., Luo H.-X., Pan Y., « Greedy feature selection for ranking », *Proceedings of the 15th International Conference on Computer Supported Cooperative Work in Design*, CSCWD'11, p. 42-46, 2011.

Laporte L., Flamary R., Canu S., Déjean S., Mothe J., « Nonconvex regularizations for feature selection in ranking with sparse SVMs », *IEEE Transactions on Neural Networks and Learning Systems*, accepté.

Pahikkala T., Airola A., Naula P., Salakoski T., « Greedy RankRLS : a Linear Time Algorithm for Learning Sparse Ranking Models », *SIGIR 2010 Workshop on Feature Generation and Selection for Information Retrieval*, p. 11-18, 2010.

Pahikkala T., Tsivtsivadze E., Airola A., Järvinen J., Boberg J., « An efficient algorithm for learning to rank from preference graphs », *Machine Learning*, vol. 75, nº 1, p. 129-165, April, 2009.

Pan F., Converse T., Ahn D., Salvetti F., Donato G., « Feature selection for ranking using boosted trees », *Proceedings of the 18th ACM Conference on Information and Knowledge Management*, CIKM'09, p. 2025-2028, 2009.

Sun Z., Qin T., Tao Q., Wang J., « Robust sparse rank learning for non-smooth ranking measures », *Proceedings of the 32nd Annual International ACM SIGIR Conference on Research and development in information retrieval*, SIGIR'09, p. 259-266, 2009.

Weston J., Elisseeff A., Schölkopf B., Tipping M., « Use of the zero norm with linear models and kernel methods », *Journal of Machine Learning Research*, vol. 3, p. 1439-1461, 2003.

Xu J., Li H., « AdaRank : a boosting algorithm for information retrieval », *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and development in information retrieval*, SIGIR'07, p. 391-398, 2007.

Yu H., Oh J., Han W.-S., « Efficient feature weighting methods for ranking », *Proceedings of the 18th ACM Conference on Information and knowledge management*, CIKM'09, p. 1157-1166, 2009.

Zhang Y., Kingsbury N., « FAST L0-based sparse signal recovery », *Machine Learning for Signal Processing (MLSP), 2010 IEEE International Workshop on*, p. 403-408, 2010.