MIR@CL
Informatique & multimédia
ISIMS - FSEGS - FSS

Atelier

" La recherche d'information personnalisée sur le web "

En conjonction avec :

La 10ème Conférence Internationale Francophone

"Extraction et Gestion des Connaissances"

Apprentissage Actif de Fonctions d'Ordonnancement d'Alternatives

Faïza DAMMAK, Hager KAMMOUN

Introduction (1)

- Avec le développement des technologies d'information
  - Nouveaux cadres d'apprentissage automatique pour la recherche de données pertinentes

Paradigme d'apprentissage supervisé de fonctions d'ordonnancement

Introduction (2)

- Apprentissage de fonctions d'ordonnancement
  - Apprendre des fonctions qui ordonnent des éléments entre eux, plutôt que de les classer

L'originalité : comparer les entrées entre elles, au lieu de les traiter indépendamment

  - En effet, l'ordonnancement est issu de la classification

Plan

- **Ordonnancement et RI**
- **Problématique : Solutions**
- **Proposition d'une Méthode d'Ordonnancement Actif d'Alternatives**
- **Conclusion /Travail à suivre**

Ordonnancement et RI (1)

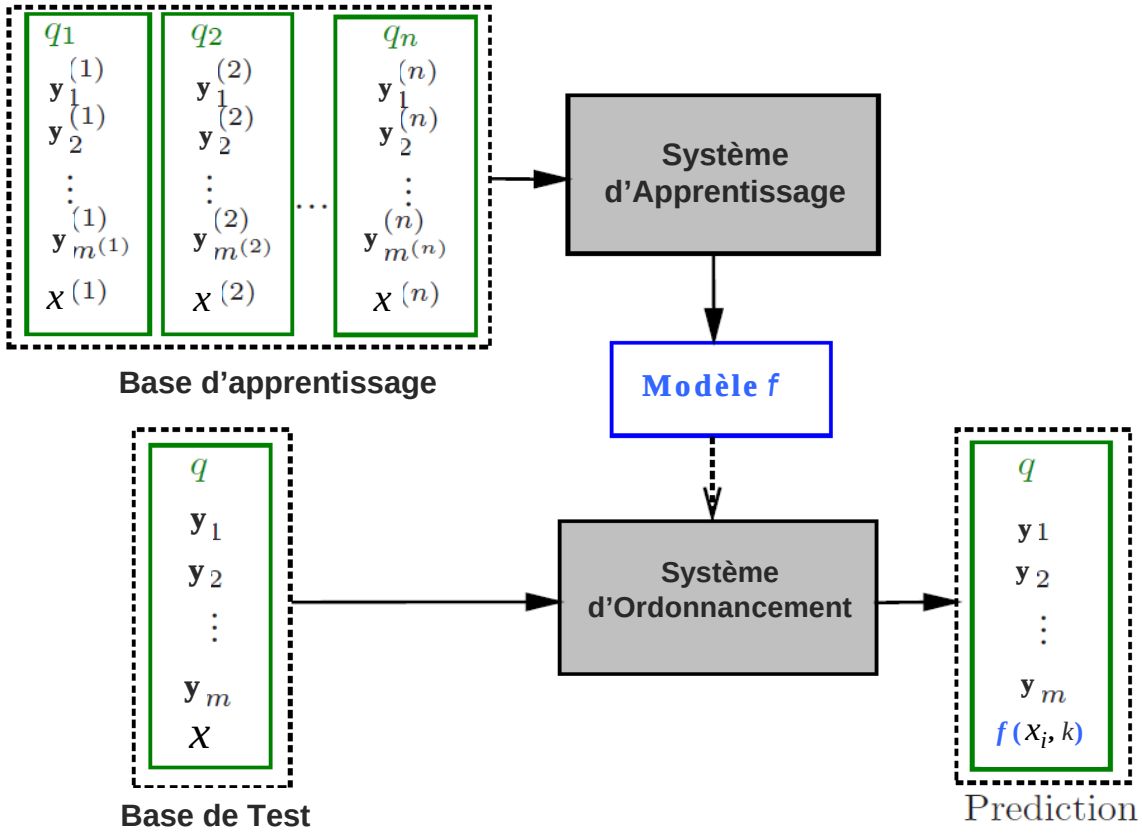
- Soit un ensemble d'exemples pertinents et de non-pertinents
- But : ordonner les exemples pertinents au dessus des exemples non-pertinents
  - cadre différent de la classification
- Fonction de score
  - fonction à valeur réelle
  - induit un ordre sur l'espace de départ
  - x est préféré à x' ssi f(x) > f(x')
- Donner un score plus élevé à un exemple pertinent qu'à un exemple non-pertinent

Ordonnancement et RI (2)

- Ordonnancement d'instances
  - Le système reçoit en entrée des documents
    - trier au fur et à mesure qu'ils arrivent
    - *RankBoost, RankSVM*
- Ordonnancement d'alternatives
  - Le système reçoit en entrée un ensemble X des observations et pour chaque observation un ensemble Y des alternatives
    - ordonner les documents en fonction d'une observation
    - la collection de documents est alors traitée simultanément
      - chaque document obtient un score par rapport à l'observation, et les documents sont présentés à l'utilisateur selon ces scores.
    - *RankBoost adapté à l'ordonnancement d'alternatives (cas biparti)*

Ordonnancement et RI (3)

Ordonnancement d'alternatives



Ordonnancement et RI (4)

RankBoost

- *Boosting* sur les paires
- Soit une ensemble de données étiquetées $S = S_{Lp} \cup S_{Ln}$
- A chaque itération *t* :
  - sélection d'une fonction de base f_t ainsi que son poids α_t associé
- Après apprentissage, la fonction de score finale est la combinaison linéaire des fonctions de base

$$F(x) = \sum_{t=1}^{T} \alpha_t f_t(x)$$

- Minimiser le nombre de paires (pertinent, non-pertinent) mal ordonnées
  - minimiser une erreur d'ordonnancement moyenne de la fonction de score finale F sur la base d'apprentissage

$$R_m^{OA}(F, S) = \frac{1}{m} \sum_{i=1}^{m} \frac{1}{n_i p_i} \sum_{k: y_i^k \in Y_+} \sum_{l: y_i^k \in Y_-} [\![f(x_i, k) - f(x_i, l) \leq 0]\!]$$

Ordonnancement et RI (5)

Algorithme RankBoost adapté à l'ordonnancement d'alternatives

- **Entrée :** Une base d'apprentissage $S_L = \{(x_i, y_i); i \in \{1,..,m\}\}$

pour chaque exemple, il y a m_i alternatives candidates

- **Initialisation :** $\forall$ i $\in$ {1, ..., *m*} $\lambda_1^i = \frac{1}{m}$, $v_1^i(k) = \begin{cases} \frac{1}{p_i} \; si \; y_i^k \in Y_+ \\ \frac{1}{n_i} \; si \; y_i^k \in Y_- \end{cases}$

Début

pour t :=1..T faire

- Sélectionner la fonction de base f_t à partir de D_t
- Calculer α_t
- $\forall$ i $\in$ {1, .., *m*}, $\forall$ (*k*, *l*) $\in$ {1, ..., *mi* }² , mettre à jour $D_{t+1}^i(k, l) = \lambda_{t+1} \; v_{t+1}^i(k) \; v_{t+1}^i(l)$
- $\forall$ i $\in$ {1, .., *m*},

$$\lambda_{t+1}^i = \frac{\lambda_t^i Z_t^{-1i} Z_t^{1i}}{Z_t} \qquad v_{t+1}^i(k) = \begin{cases} \frac{v_t^i(k) \exp(-\alpha_t f_t(x_i, k))}{Z_t^{1i}} \; si \; y_i^k \in Y_+ \\ \frac{v_t^i(k) \exp(\alpha_t f_t(x_i, k))}{Z_t^{-1i}} \; si \; y_i^k \in Y_- \end{cases}$$

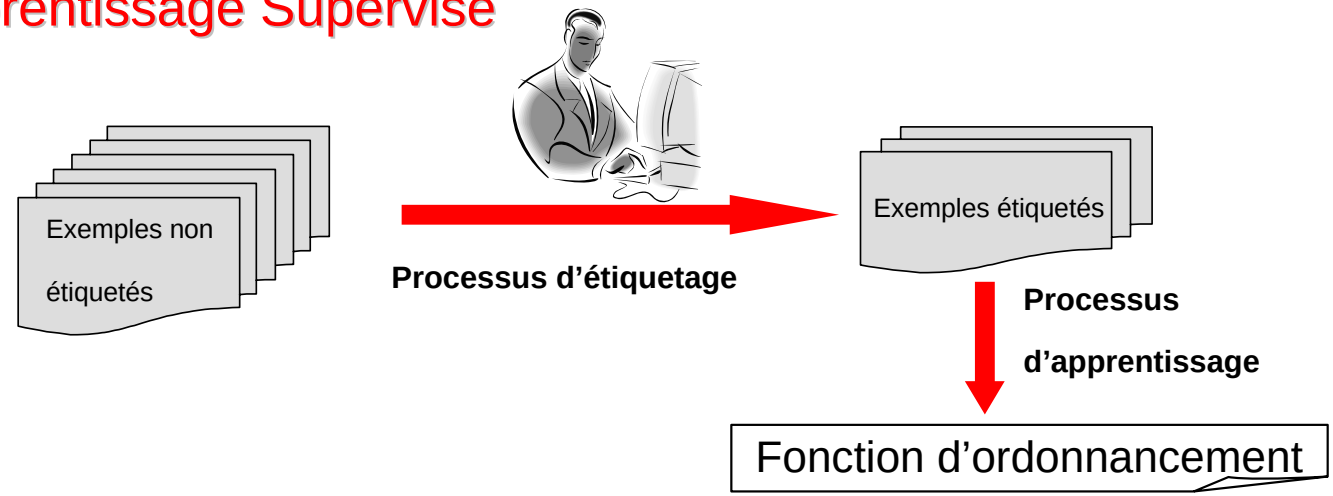
où Z_t^{1i} , Z_t^{-1i} et Z_t sont défini par: $Z_t^{1i} = \sum_{k: y_i^k \in Y_+} v_t^i(k) \exp(-\alpha_t f_t(x_i, k))$ $Z_t^{-1i} = \sum_{l: y_i^l \in Y_-} v_t^i(l) \exp(\alpha_t f_t(x_i, l))$ $Z_t = \sum_{i=1}^m \lambda_t^i Z_t^{-1i} Z_t^{1i}$

Fin

- **Sortie : La fonction de score finale** $F = \sum_{t=1}^T \alpha_t f_t$

Problématique : Solutions (1)

Apprentissage Supervisé



Inconvénient : l'étiquetage manuel

- coûteux en temps et en ressource
- demande des efforts de la part de l'utilisateur

Problématique : Solutions (2)

Solutions

- ***Apprentissage Semi-Supervisé***
- ***Apprentissage Actif***

} *résoudre le même problème mais avec des perspectives différentes*

Prendre en compte les données étiquetées et non-étiquetées dans le processus d'apprentissage.

Problématique : Solutions (3)

- Apprentissage Semi-Supervisé
  - Entrée
    - un petit ensemble de données étiquetées S_L
    - un large ensemble de données non-étiquetées S_U
  - Entrainer un model M sur $S_L \cup S_U$

Problématique : Solutions (4)

- Algorithmes d'ordonnancement semi-supervisé
  - Techniques transductives à base de graphe
  - Ne peuvent pas étiqueter les exemples absents de la phase d'apprentissage puisqu'ils ne font pas partie des nœuds du graphe

Inconvénient : peut être évité par l'utilisation d'une approche Active

Problématique : Solutions (5)

- Apprentissage Actif
  - Entrée
    - un petit ensemble de données étiquetés S_L
    - un large ensemble de données non-étiquetés S_U
  - Repéter
    - Entrainement du model(s) *M* sur S_L
    - Utilisation de *M* pour tester S_U
    - *Selection de l'exemple le plus utile* de S_U
    - Interrogation d'un expert humain pour l'étiqueter
    - Addition de l'exemple étiqueté à S_L

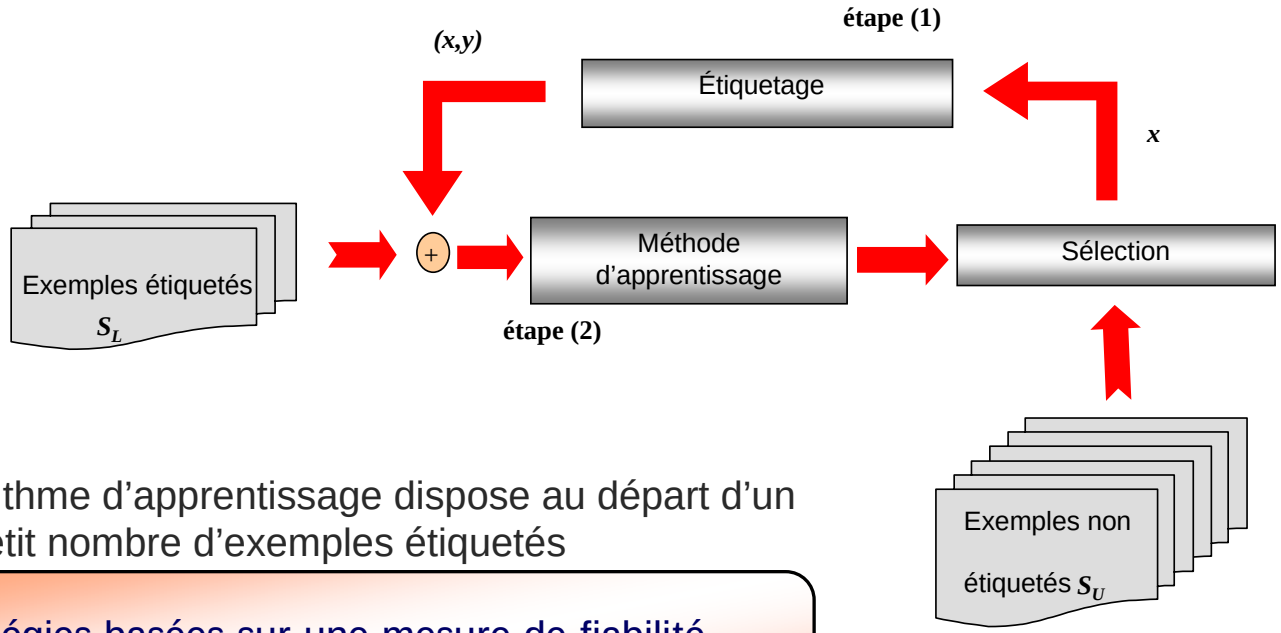
  Jusqu'à *M* atteigne un certain niveau de performance ou un certain nombre de requêtes

Proposition d'une Méthode d'Ordonnancement Actif d'Alternatives (1)

- Objectif ?
  - Développer un algorithme ***Actif***
    - capable de donner un score à n'importe quel exemple
    - capable de gérer un grand nombre de données non-étiquetées
      - étiqueter les exemples non-étiquetés
      - apprendre un nouveau modèle
- Processus d'étiquetage
  - Comment étiqueter l'exemple non-étiqueté à chaque itération ?
    - Approche sélective
    - Approche constructive

Proposition d'une Méthode d'Ordonnancement Actif d'Alternatives (2)

Approche sélective en apprentissage actif



Algorithme d'apprentissage dispose au départ d'un petit nombre d'exemples étiquetés

Stratégies basées sur une mesure de fiabilité, une estimation de la réduction de l'erreur

Proposition d'une Méthode d'Ordonnancement Actif d'Alternatives (3)

- Particularité de l'apprentissage actif réside dans

Sélection incrémentale des exemples d'apprentissage les plus utiles à un instant donné pour effectuer l'apprentissage

Accélération de l'apprentissage en considérant d'abord les exemples les plus informatifs.

Interaction du modèle avec son environnement.

Réduction de la base nécessaire à l'élaboration d'un modèle pertinent surtout lorsque les données sont coûteuses.

Proposition d'une Méthode d'Ordonnancement Actif d'Alternatives (4)

- Adaptation à l'apprentissage actif
  - sélectionner les entrées de la base étiquetée S_L
  - apprendre une nouvelle fonction de score sur cette base en utilisant **RankBoost**
  - sélectionner une ou plusieurs alternatives d'un ensemble non-étiqueté S_U et interroger un oracle pour obtenir leur étiquette
  - ajouter ces alternatives à l'ensemble d'apprentissage pour apprendre une nouvelle fonction de score

Ces étapes vont être répétées jusqu'à ce qu'une condition d'arrêt soit satisfaite.

Proposition d'une Méthode d'Ordonnancement Actif d'Alternatives (5)

Algorithme l'ordonnancement d'alternatives actif

Entrée :
Une base d'apprentissage S_L et une base d'apprentissage non-étiquetée S_U

L'algorithme d'ordonnancement supervisé RankBoost adapté à l'ordonnancement d'alternatives

k : nombre de partitions de S_L

Pour t : = 1,..., k **faire**

- Apprendre une fonction d'ordonnancement avec RankBoost sur S_L
- étiqueter les alternatives associées en se focalisant uniquement sur les données non-étiquetées disponibles S_U
- Les retirer de S_U et les ajouter dans S_L avec les étiquettes sur les alternatives

Fin

Proposition d'une Méthode d'Ordonnancement Actif d'Alternatives (6)

- Evaluation : Benchmark standard LETOR
  - Réaliser pour l'apprentissage de fonctions d'ordonnancement dans le domaine de la RI
  - Propose un ensemble de résultats issus de plusieurs algorithmes d'ordonnancement tels que RankBoost
  - LETOR3.0 : OHSUMED et .GOV

Proposition d'une Méthode d'Ordonnancement Actif d'Alternatives (7)

- Mesures d'évaluation
  - *Précision à une position n : P@n*

$$P@n = \frac{\#docs\ pertinents\ dans\ les\ n\ instances\ les\ mieux\ ordonnées}{n}$$

  - *Précision moyenne : MAP (Mean Average Precision)*

$$AP = \frac{\sum_{n=1}^{N}(P@n * rel(n))}{\#total\ docs\ pertinents\ pour\ cette\ requête}$$

  - *Gain cumulatif normalisé escompté : NDCG (Normalized Discounted Cumulative Gain)*

$$NDCG(n) = Z_n \sum_{j=1}^{n} \frac{2^{r(j)} - 1}{\log(1 + j)}$$

Conclusion

- Amélioration de la qualité de la recherche documentaire
  - Stratégie d'apprentissage actif de fonctions d'ordonnancement d'alternatives
    - L'apprentissage actif permet de créer un modèle pertinent avec des données d'apprentissage peu nombreuses car bien choisies
    - L'application de l'algorithme RankBoost pour l'apprentissage d'un modèle d'ordonnancement
    - L'élaboration d'une base étiquetée à faible coût

Travail à suivre

- Evaluation de notre algorithme
- Plus d'expériences avec des ensembles de données différents
- Tester et combiner différentes approches pour la sélection des exemples à étiqueter pour l'apprentissage actif
- …

MIR@CL
Informatique & multimédia
ISIMS - FSEGS - FSS

MERCI POUR VOTRE ATTENTION