

## The alternating decision tree learning algorithm

Yoav Freund – AT&T Labs, USA

Llew Mason – Department of Systems Engineering, University of Canberra, Australia

Résumé: Laurent Siczkowski – M2R Orsay

### Résumé:

Cet article présente une généralisation d'arbre de décision (arbres votés de décision et « stumps<sup>1</sup> » votés). Ce type de classifieur possède une interprétation plus simple, notamment par rapport à C5.0. Les auteurs présentent également un algorithme d'apprentissage, basé sur le «boosting», pour ce nouveau classifieur. L'expérimentation montre que cet algorithme est compétitif avec les références dans le domaine, tel que C5.0. Il est démontré que cette technique combine les avantages des arbres de décisions et des « stumps » votés.

### Introduction:

Cet article se base sur des travaux récents d'apprentissage tel que Adaboost ( Schapire et al. 97). Les classifieurs utilisant ce procédé ne manque pas, parmi les plus connus, CART<sup>2</sup> (Breiman 84), C4,5 (Quinlan 93) ou C5,0.

Les auteurs proposent une nouvelle combinaison d'arbre de décision avec boosting. Cet arbre est introduit sous le nom de *alternating decision trees*<sup>3</sup> (ADTrees). Le boosting est abordé par le biais des travaux de (Kearns et al. 96) pour être utilisé comme une méthode d'apprentissage. Les ADTrees utilisent une mesure de confiance (la marge de classification), cette mesure a été traité et validé (Schapire et al. 98).

### Arbres alternés de décision:

Un ADTree comporte deux types de noeuds, les noeuds de décision, qui sont identique à ceux utilisés dans les arbres classiques, et des noeuds de prédiction qui sont associés à une valeur Réel. Dans un arbre de ce type, l'évaluation se fait en estimant l'intégralité du chemin de la racine à la feuille de sortie. Ainsi, la classification s'effectue par le signe de la somme des règles, en ajoutant tout les poids des noeuds de prédictions compris dans ce chemin.

On trouve donc trois couches dans ce type d'arbre: la racine, un ensemble de noeuds de décision et les noeuds de prédictions associés aux noeuds de décision. La racine est la meilleur constante de prédiction sur l'ensemble des données. On ajoute ensuite itérativement les règles ( une règle correspondant à un sous arbre avec un noeud de décision et deux noeuds de prédiction). La règle rajouter et considérer comme fils d'un noeud de prédiction.

Il n'y a pas de moyen précisé pour arrêter l'apprentissage sur cette arbre. Les auteurs ont choisis d'utiliser la validation croisée pour effectuer ce choix. La phase d'entraînement se fait donc sur des ensembles différents des ensembles de test, et s'arrête lorsque l'algorithme détecte le sur-apprentissage.

### Boosting des arbres alternés de décision:

Le boosting s'effectue par une version d'Adaboost suggéré par Schaphire et Singer (Improved boosting algorithms using confidence-rated predictions – 98). Cette version permet d'utiliser des valeurs Réels dans les règles de prédiction. Cette référence d'article défini aussi une formule de calcul donnant la meilleur prédiction pour une partition des données. Les auteurs

---

1 Cas particulier où la structure de l'arbre se compose d'une noeud de décision avec deux noeuds de prédiction.

2 Classification And Regression Trees : arbre de classification et de régression

3 Arbre alterné de décision.

utilisent donc cette formule pour déterminer le seuil des règles.

L'algorithme se décompose en 4 étapes:

- Pour chaque couple (précondition, condition), calcul de  $Z_t(\text{précondition}, \text{condition})$
- Trouver le couple minimisant  $Z_t$ , ajouter les règles correspondantes et les valeurs de prédictions calculées.
- Ajout des préconditions sous la forme  $\text{précondition} \wedge \text{condition}$ ,  $\text{précondition} \wedge \text{non condition}$  (les 2 choix possibles)
- Mise à jour des poids

On trouve donc deux ensembles gérés par cette méthode, les préconditions et la base de règles (respectivement  $P_t$ ,  $R_t$  pour l'itération  $t$  donné, la première précondition étant toujours vrai).

Il est donc relativement proche d'algorithme comme C4,5 et CART (ajout pas à pas d'une règle définissant la meilleure séparation). Il y a cependant quelques différences:

- Un nœud de décision s'ajoute à n'importe quel endroit.
- Critère de séparation équivalent à l'erreur de poids d'une règle ajoutée.

#### Interprétation des arbres alternés de décision:

Globalement, la représentation d'un ADTree lui confère une bonne compréhension. Chaque nœud de décision est explicable séparément (condition) et la somme des contributions donnent la classification (les nœuds de prédiction valorisant une classe). On notera également que la marge de classification désigne le niveau d'influence d'un nœud de décision. Il s'agit du rang de création du nœud, les premiers étant les plus importants.

#### Expériences et résultats:

Les résultats montrent que l'ADTree est compétitif avec un C5,0+boost et autre référence du domaine. En fait sans réellement les dépasser, il prend les avantages du StumpBoost et de C5,0 (somme votée et arbre de décision). Le résultat est plus constant. Lorsque C5,0 est meilleur que StumpBoost, ADTree est meilleur également, et réciproquement quand StumpBoost est meilleur que C5,0, ADTree aussi.

#### Conclusion:

Cet article décrit un nouveau classifieur interprétable et un procédé d'apprentissage par boosting. Les performances le rapprochent de la référence C5,0+boost avec des résultats plus intéressants parfois.

Les perspectives sont :

- Utilisation d'un graphe de calibration (Quand ne pas mettre de prédiction).
- Utilisation d'un boosting moins sensible au bruit par rapport à Adaboost.
- Optimisation en temps pour des larges ensembles de données.

#### Mon avis:

Dans l'ensemble cet article est bien écrit, j'aurais apprécié un approfondissement ou plutôt un rappel appuyé par un exemple pour la distribution des poids.

Au niveau du travail en lui-même, l'article fait l'objet de nombreuses citations positives confirmant les résultats d'expériences.