Contexte

Definition sélection d'attributs

$\mathbf{A} = \{a_0, a_1, \ldots, a_n\}$ un ensemble d'attributs;

T une tâche pour laquelle on dispose d'un oracle qui retourne pour chaque sous ensemble d'attributs un vecteur qui décrit les performances de ce vecteur, on représente l'oracle par la fonction suivante :

$$T : \begin{vmatrix} \mathbf{2^A} & \longrightarrow & (\mathbf{R^n}, k) \\ \alpha & \longmapsto & (v, |\alpha|) \end{vmatrix}$$

La tâche consiste donc a déterminer pour une relation donnée $\succeq$ l'ensemble :

$$ND_{\succeq} = \{x | \forall y, y \succeq x \rightarrow x \succeq y\}$$

des solutions non dominées au sens de la relation $\succeq$.

Framework général des approches de sélection d'attributs

Selon cet review de Dash et Liu on peut distinguer trois composantes essentielles dans chaque approche de sélection d'attributs:

- **Méthode d'exploration**: Les méthodes de ranking explorent les sous ensemble de 1 attribut, d'autres méthodes utilisent une heuristique gloutonne ou une méta-heuristique pour générer des sous-ensembles a évaluer.
- **Fonction d'évaluation**: Idéalement on aimerait pouvoir tout le temps évaluer en effectuant une classification avec le sous-ensemble désigné par la méthode de génération, cependant, certaines approches utilisent d'autres fonctions pour évaluer un sous ensemble par souci de temps, ces méthodes peuvent mesurer un distance vis a vis de la classe a prédire, la qualité d'information de l'attribut, la consistance de l'attribut, ou bien la dépendance probabiliste de la classe vers l'attribut.
- **Condition d'arrêt**: Dans le cas des méthodes de ranking la condition d'arrêt est soit d'avoir atteint un nombre d'attribut donné, ou alors d'ajouter un attribut qui diminue la performance.

  Pour les méthodes basées sur une méta-heuristique c'est souvent un seuil de précision ou un nombre d'itérations.

Idées d'application

Objectif

L'objectif de cette proposition est de mettre en oeuvre une approche qui modélise l'efficacité d'un sous ensemble d'attribut pour une tache donnée (typiquement une classification) en décrivant une **relation ordinale sur les sous ensembles d'attributs**.

Cette relation est définie pour deux sous ensembles d'attributs A_1, A_2 par $A_1 \succeq A_2$ Si A_1 domine A_2 au sens de pareto mais peut également être précisée par un décideur externe ce qui permettrait d'arbitrer des décisions comme comparer un ensemble qui fait 80% avec 2 attributs et un autre qui en ferait 81% avec 1000 attributs.

Principe de la méthode

La méthode consiste à reprendre le schéma classique d'une approche de sélection d'attributs: génération, évaluation jusqu'a une certaine condition d'arrêt mais chaque évaluation va créer un certain nombre de relations de préférences qui découlerons de la dominance de pareto ou de préférences de l'utilisateur qui effectue la selection d'attributs, donc par exemple :

- $v(\{1,2,3\}) = (78\%, 3)$; $v(\{4,6\}) = (80\%, 2)$ donc $\{4,6\} \succ \{1,2,3\}$ car il fait un meilleur score de précision avec moins d'attributs.
- $v(\{1,2,3,4,7,8\}) = (60\%, 6)$; $v(\{5\}) = (59\%, 2)$ donc à priori les deux sous ensemble sont incomparables, cependant cette modélisation est agnostique quant à la provenance des préférences donc l'utilisateur peut par exemple dire que pour 6 attributs ce n'est pas acceptable de n'avoir que 60% et donc arbitrer que $\{4,6\} \succ \{1,2,3\}$
- $v(\{1,4,3\}) = (60\%, 3)$; $v(\{5,6,7\}) = (60\%, 3)$; les deux sous ensembles sont absolument identiques en terme de performances mais la encore le décideur pourrait arbitrer pour une raison qui l'incombe que le premier domine le second.

Pistes

A présent je vais décrire les différentes pistes d'utilisation de notions issues de la théorie des relations ordinales pour la sélection d'attributs.

Utilisation d'un indice de pouvoir pour l'exploration des sous ensembles d'attributs

Dans la mesure ou l'exploration profiterait d'une génération plus "intelligente" de sous ensembles prometteurs on pourrait imaginer utiliser des indices de pouvoirs afin de tirer des ensembles d'attributs, notons $I_i(R)$ l'indice de pouvoir de l'attribut i en considérant la relation ordinale sur le power set R on pourrait poser par exemple

$$P(i \subset S|R) = \frac{I_i(R)}{\sum_{i \in A} I_i(R)}$$

Cette idée pose cependant trois problèmes :

- L'évaluation d'un sous ensemble d'attribut est couteuse il serait contre productif d'évaluer trop fréquemment $v(S)$.
- La relation d'ordre sur le power set est fortement incompléte, un sous ensemble n'y figure que si il est évalué par classification ce qui n'est pas sensé arriver trop souvent.
- Le calcul de la pluspart des indices de pouvoir tel que l'indice de Shapley ou de Benzhaff recquierent l'énumération d'un ensemble de taille exponentielle de sous ensembles.

Plusieurs pistes peuvent cependant être proposées pour palier a ces difficultés :

- Pour l'énumération d'un ensemble de taille exponentielle de sous ensemble, on peut essayer de restreindre le jeu à un jeu coallitionnel simple ou alors adopter une approche de Monte Carlo consistant a sampler aléatoirement une partie des sous ensembles qu'il faudrait évaluer.
- Pour palier à l'incomplétude de la relation d'ordre et pour limiter le nombre d'appels à l'oracle on peut évaluer la contribution marginale de l'attribut i dans S en utilisant

l'hypothèse formulée plus haut et une méthode pire cas, on pourrait par exemple comparer $max_{\theta \in \theta_R} U_\theta(S)$ et $min_{\theta \in \theta_R} U_\theta(S \cup \{i\})$ ce qui corresponderait à la contribution au pire cas.
- On peut également proposer une nouvelle regle de calcul de l'indice de pouvoir qui soit plus facile, en se basant par exemple sur le regret chaque ensemble pourrait donner une voix pour ou une voix contre chaque attribut en se basant sur $min_{\theta \in \theta_R} U_\theta(S)$ et $max_{\theta \in \theta_R} U_\theta(S/\{i\})$
- On peut aussi exploiter les paramètre θ du modèle en définissant pour chaque attribut i les paramètres θ_i tel que toute modification du reste des paramètres n'influe pas sur la différence entre $U_\theta(S)$ et $U_\theta(S \cup \{i\})$, et ensuite sommer ces paramètres pour déduire l'impact de l'attribut considéré.

Exploration de solutions potentiellement optimales

Une autre piste serait de trouver un moyen de restreindre l'exploration des sous-ensembles à ceux qui sont potentiellement optimales selon U_θ, pour U_θ 2-additive Ariane Ravier a proposée une méthode pour générer une solution potentiellement optimale.

On peut également déduire une méthode générique qui consisterait à :

- À chaque itération pour chaque S on calcule $U_{min}(S) = min_{\theta \in \theta_R} U_\theta(S)$ et $U_{max} = max_{\theta \in \theta_R} U_\theta(S)$ et on retire les sous ensembles S tel que $\exists S_2; U_{min}(S_2) \geq U_{max}(S)$ pour ne garder que les solutions qui ne sont pas dominés.
- Avant d'évaluer un ensemble S par classification on vérifie qu'il n'existe pas d'ensemble S_2 tel que $U_{min}(S_2) \geq U_{max}(S)$.

Méthode de Monte Carlo pour la génération

On peut voire le tirage comme un processus séquentiel ou chaque attribut tiré modifie les probabilités de tirer les autres et ce de plusieurs manières.

- **En prennant le pire cas**: A chaque étape ayant déja tiré un ensemble S, pour chaque élément i qu'il est encore possible de tirer on évalue $\frac{U_{min}(S \cup i)}{U_{min}(S)}$ qui va représenter la probabilité de transition vers i.
- **En fixant au préalable** θ : On peut par exemple exploiter **"l'explication la plus simple"** qui consiste à choisir θ de manière à ce qu'il soit compatible avec les préférences et que $|\theta|$ soit minimale.

Définir comme condition d'arrêt de la recherche l'existence d'une solution nécessairement optimale

On peut tester si une solution S est nécessairement optimale en résolvant

$$\theta_{min} = argmin_\theta(U_\theta(S))$$

Puis en résolvant

$$U = max_{S_i}(U_{\theta_{min}}(S_i))$$

Et si on trouve $U = U_{\theta_{min}}(S)$ ça veut dire que l'ensemble S est nécessairement optimal.

Si on trouve des ensembles nécessairement optimales de tailles différentes, cela signifie qu'un arbitrage de l'utilisateur doit être effectué car la relation de dominance de pareto ne pourra jamais les départager et donc la fonction U la modélisant non plus.

Ce n'est pas anodin comme remarque car jamais auparavant un critère d'arrêt aussi précis n'a été formulé, alors que ici sous une hypothèse sur la fonction qui modélise la relation ordinale entre les sous-ensemble on dispose d'un critère d'arrêt et d'un critère qui détermine la nécessité d'un arbitrage de l'utilisateur.

Utilisation de méthodes alternatives d'évaluation pour générer un vecteur de performances

L'utilisation d'une fonction $U_\theta(S)$ change drastiquement la compléxité d'une approche de sélection d'attributs car contrairement à l'oracle T le temps d'éxécution d'un programme impliquant U_θ n'augmente pas en fonction du nombre d'examples mais uniquement en fonction du nombre d'attributs ce qui peut être particulièrement pratique dans un contexte de big data.

Une autre piste a envisager pour continuer dans cette même veine est d'utiliser plusieurs mesures d'évaluation ensemblistes afin de calculer pour chaque sous ensemble d'attributs un vecteur de performance v et ensuite utiliser $U_\theta(v)$ plutôt que $U_\theta(S)$.

De plus, réduire le nombre de dimensions du problème permet de reconsidérer l'utilisation d'un indice de pouvoir tel que l'indice de Shapley.

Mieux caractériser les performances de classification

Sur les classes :

En réalité, utiliser la précision c'est déja faire un choix arbitraire qui consiste à évaluer le modèle en sommant les vrais positifs avec les vrais négatifs et ceci est un choix discutable sur le fond car cette matrice de confusion :

Prédis/Réel	**Vrai**	**Faux**
Vrai	3	1
Faux	1	3

Donne la même précision que celle la :

Prédis/Réel	**Vrai**	**Faux**
Vrai	2	0
Faux	2	4

Alors que cette seconde a un moins bon taux de détection des vrais positifs ce qui pourrait être moins intéressant dans un contexte de dépistage par exemple. le choix de la précision est fait dans un optique de simplification afin d'obtenir des valeurs cardinales à comparer mais étant donné que de toute façon les relations que nous manipulons sont ordinales nous pouvons nous affranchir de ce parti pris et utiliser un oracle qui renvoie pour chaque sous ensemble A le vecteur $(VP, VN, |A|)$.

Evidemment c'est encore plus valable sur du multi-classe. et en plus ça permettrait au décideur d'expliciter encore mieux ces préférences en terme de performances.

Sur des parties de l'ensemble de données

Une autre moyenne qu'on a l'habitude d'exploiter pour déterminer la qualité de la classification est d'évaluer le sur-apprentissage, pour se faire il est souvent courrant de faire de la cross-validation qui fournit des préférences testés sur plusieurs ensembles.

La moyenne de ces précisions est ensuite utilisée pour représenter les performances de classification mais encore une fois c'est un choix arbitraire, on pourrait penser a un autre aggrégateur ou même représenter chaque performance a part dans le vecteur de performances retourné par l'oracle.

Sur des données qui concernent différentes catégories de la population

Une autre façon de voire les choses consiste a évaluer les performances de notre modèle sur différentes catégories de la population pour vérifier qu'il ne discrimine personne, par exemple, dans une application de credit scoring on essaie de prédire la probabilité qu'un crédit soit remboursé et à la fin on peut s'intéresser à la différence dans le ratio de faux négatif chez les hommes et chez les femmes.

Etant donné que cette mesure est très précise et sophistiqué, il est rare qu'elle soit inclue dans le processus de Feature Engineering autrement que juste sous forme d'une validation finale, mais étant donné la grande fléxibiltié de notre approche, on peut également décomposer les performances suivant les performances sur différents groupes humains.