Recherche opérationnelle et données massives

Arbres de décision optimaux

1 Instances principales

1.1 Méthode F

Testons l'algorithme fourni sur les trois premiers jeux de données. Nous obtenons les résultats de la table 1.

Instance	D	Uni/Multivarié	Temps (s)	Gap	Erreurs train	Erreurs test
Iris	2	U	3.3	0%	5	1
		M	4.2	0%	1	0
	3	U	40.8	0%	0	3
		M	1.7	0%	0	2
	4	U	25.2	0%	0	1
		M	13.4	0%	0	2
Seeds	2	U	41.1	0%	9	4
		M	1.2	0%	0	1
	3	U	120	3.1%	5	2
		M	7.1	0%	0	2
	4	U	120	3.1%	5	3
		M	24.0	0%	0	2
Wine	2	U	31.9	0%	5	3
		M	0.6	0%	0	1
	3	U	120	1.4%	2	0
		M	0.6	0%	0	1
	4	U	120	0.7%	1	1
		M	9.6	0%	0	4

TABLE 1 – Résultats sur les instances principales

Le gap correspond à l'écart à la relaxation continue. Il est non-nul lorsque le temps d'exécution atteint la limite de temps.

On constate que la version multivarié est bien plus rapide que la méthode univariée. De plus, elle est, en général, meilleure : les erreurs sur les données de test (et de train) sont souvent plus faibles.

Il est difficile d'évaluer la qualité des résultats en fonction de la profondeur de l'arbre. En effet, aucune tendance stricte ne se dégage de ces essais.

1.2 Regroupement naïf

Pour réduire le temps d'exécution de cette méthode, regroupons les données en clusters. Nous utilisons la méthode naïve, pour différentes valeurs de γ. Les résultats pour l'instance Iris sont donnés en table 2. Nous avons gardé la limite de temps à 10 secondes par méthode.

Sur toutes les instances, nous constatons que, pour des faibles ou grands pourcentages de regroupements ($\gamma = 0\%$ ou $\gamma \geq 80\%$), les erreurs sont grandes. Pour γ proche de 50%, les résultats sont bons.

2 Instances supplémentaires

Nous allons à présent tester trois nouveaux jeux de données.

Le premier consiste à prédire si des couples vont divorcer ou non. Les attributs sont les réponses à 54 questions, auxquelles les couples peuvent répondre par oui ou non. Pour ce jeu de données, la taille de l'ensemble d'entraînement est 136, et celle de l'ensemble de test est 34.

Le deuxième correspond à des données d'étudiants d'université scientifique. L'enjeu est de prédire si les étudiants vont réussir leurs examens. Il y a 31 attributs, dont 10 questions personnelles (âge, sexe,

Instance	D	Uni/Multiv.	γ et nb clusters	Temps (s)	Gap	Err. train	Err. test
Iris	2	U	0% / 3	0.4	0%	40	10
			20% / 24	0.3	0%	6	2
			40% / 48	0.5	0%	5	1
			60% / 72	1.4	0%	5	1
			80% / 96	2.2	0%	5	1
			100% / 120	3.7	0%	5	1
		M	0% / 3	0.1	0%	40	10
			20% / 24	0.6	0%	1	1
			40% / 48	0.9	0%	1	0
			60% / 72	2.2	0%	1	0
			80% / 96	2.2	0%	1	1
			100% / 120	3.4	0%	1	0
	3	U	0% / 3	0.1	0%	40	10
			20% / 24	2.9	0%	3	2
			40% / 48	10	0%	1	1
			60% / 72	3.6	0%	0	2
			80% / 96	10	0%	1	1
			100% / 120	10	0%	2	1
		M	0% / 3	0.3	0%	40	10
			20% / 24	0.8	0%	0	2
			40% / 48	0.7	0.8%	0	1
			60% / 72	1.6	0%	0	1
			80% / 96	1.7	0.8%	0	0
			100% / 120	4.7	1.7%	0	1
	4	U	0% / 3	0.1	0%	40	10
			20% / 24	10	3.4%	3	2
			40% / 48	10	0.8%	1	0
			60% / 72	10	2.6%	3	2
			80% / 96	10	4.3%	5	1
			100% / 120	10	66.7%	48	13
		M	0% / 3	1.2	0%	40	10
			20% / 24	1.2	0%	0	1
			40% / 48	3.0	0%	0	2
			60% / 72	9.4	0%	0	0
			80% / 96	4.4	0%	0	2
			100% / 120	10	33.3%	3	3

TABLE 2 – Résultats sur l'instance iris avec regroupements

salaire, bourse, ...), 6 questions sur la famille de l'étudiant, et le reste sur des questions d'habitudes de travail. L'ensemble d'entraînement a 116 points, et l'ensemble de test en a 29.

Enfin, le troisième cherche à prédire l'accent d'un interlocuteur, parmi 6 pays différents. A partir d'enregistrements audio, les auteurs ont extrait 12 attributs. L'ensemble d'entraînement a 263 points, et l'ensemble de test, 66.

2.1 Méthode F

Les résultats sont donnés en table 3.

Nous constatons que la première instance se résout bien en moins de 2 minutes (avec au plus une erreur sur l'ensemble d'entraînement). En revanche, pour les deux autres, les erreurs sur les ensembles de test sont au mieux à 59% pour l'exemple des étudiants et 30% pour l'origine de l'accent.

2.2 Avec regroupements

Essayons à présent de regrouper les données des nouvelles instances "Étudiants" et "Accent" (l'instance "Divorce" se résolvant déjà rapidement sans). Nous regardons uniquement la modélisation multi-

Instance	D	Uni/Multivarié	Temps (s)	Gap	Erreurs train	Erreurs test
Divorce (taille train = 136, taille test = 34)	2	U	9.9	0%	0	1
		M	1.5	0%	0	0
	3	U	5.9	0%	0	1
		M	6.0	0%	0	1
	4	U	24.7	0%	0	1
		M	76.1	0%	0	1
Etudiants (taille train = 116, taille test = 29)	2	U	120	67.9%	71	20
		M	3.1	0%	39	23
	3	U	120	176.2%	74	19
		M	120	4.5%	5	21
	4	U	120	197.4%	77	17
		M	79.6	0%	0	25
Accent (taille train = 263, taille test = 66)	2	U	120	59%	121	32
		M	120	7.9%	61	24
	3	U	120	70.8%	109	36
		M	120	21.8%	41	20
	4	U	120	99.2%	131	33
		M	120	1215%	243	57

TABLE 3 – Résultats sur les instances supplémentaires

variée, pour des pourcentages de regroupement de 20, 40 ou 60%. Limitons le temps d'exécution à 60 secondes par méthode.

Les résultats sont fournis en table 4. Pour l'instance "Accent", l'erreur moyenne diminue sur les ensembles de test. En revanche, pour l'instance "Etudiants", le gap est nul (ou proche de zéro) mais les regroupements ont plutôt augmenté l'erreur sur les ensembles de test.

Instance	D	γ et nb clusters	Temps (s)	Gap	Err. train	Err. test
Etudiants (taille train = 116, taille test = 29)	2	20% / 23	1.6	0%	39	23
		40% / 46	1.1	0%	39	23
		60% / 69	2.0	0%	39	23
	3	20% / 23	60	5.5%	6	20
		40% / 46	60	5.5%	6	21
		60% / 69	60	4.5%	5	25
	4	20% / 23	17.2	0%	0	25
		40% / 46	31.3	0%	0	22
		60% / 69	34.9	0%	0	25
Accent (taille train = 263, taille test = 66)	2	20% / 52	28.3	0%	76	21
		40% / 105	60	17.2%	74	27
		60% / 157	60	10.7%	66	24
	3	20% / 52	60	32.8%	65	24
		40% / 105	60	35.6%	66	23
		60% / 157	60	44.5%	72	20
	4	20% / 52	60	37%	57	22
		40% / 105	60	30.2%	58	27
		60% / 157	60	42.2%	67	23

TABLE 4 – Résultats sur les nouvelles instances avec regroupements

3 Approfondissement

3.1 Ajout d'inégalités valides

Pour $k \in \mathcal{K}$, notons $\mathcal{I}^k$ les données de la classe k et pour $t \in \mathcal{N}$, notons $A(t)$ l'ensemble des ancêtres de t.

Considérons les inégalités suivantes :

$$u^{i_1}_{t,l(t)} + u^{i_2}_{t,r(t)} \leq 1 \qquad \forall t \in \mathcal{N},\ \forall j \in \mathcal{J},\ \forall i_1, i_2 \in \mathcal{I} : x_{i_1,j} \geq x_{i_2,j} \qquad (1)$$

$$c_{k,t} + u^i_{t,w} + u^i_{t,l(t)} + u^i_{t,r(t)} \leq 1 \qquad \forall t \in \mathcal{N},\ \forall k \in \mathcal{K},\ \forall i \in \mathcal{I}\backslash\mathcal{I}^k \qquad (2)$$

$$2\sum_{k\in\mathcal{K}} c_{k,t} + u^{i_1}_{t,r(t)} + u^{i_2}_{t,l(t)} + \sum_{\substack{j\in\mathcal{J}:\\ x_{i_1,j}\leq x_{i_2,j}}} a_{j,t} \leq 2 \qquad \forall t \in \mathcal{N},\ \forall i_1, i_2 \in \mathcal{I} \qquad (3)$$

$$\sum_{j\in\mathcal{J}} a_{j,t} + \sum_{k\in\mathcal{K}} (c_{k,t} + \sum_{t'\in A(t)} c_{k,t'}) = 1 \qquad \forall t \in \mathcal{N} \qquad (4)$$

Ce sont toutes des inégalités valides. En effet :

— L'inégalité 1 vient du fait que si, pour un attribut j, deux données i_1 et i_2 sont telles que $x_{i_1,j}$ est supérieur ou égal à $x_{i_2,j}$ alors soit les deux données sont bien classées en t à gauche, soit elles sont toutes les deux bien classées à droite, soit i_1 est bien classée à droite et i_2 à gauche. On ne peut pas bien classer i_1 à gauche et i_2 à droite.

— L'inégalité 2 affirme que si une donnée i n'est pas dans une classe k, alors à un nœud t, soit on prédit la classe k, soit la donnée i va à gauche ou à droite ou en w (soit il ne se passe aucune de ces situations).

— L'inégalité 3 vient du fait que si on prédit une classe k en t alors aucune donnée ne va à droite ou gauche en t et il n'y a pas de séparation. A l'inverse, si on ne prédit rien en t, mais si t effectue une séparation sur une caractéristique t telle que $x_{i_1,j} \leq x_{i_2,j}$ alors on ne peut pas bien classifier i_1 et i_2.

— Enfin, l'inégalité 4 représente le fait que si le sommet t effectue un séparation alors, ni t, ni aucun de ses ancêtre n'est une feuille et si le sommet t n'effectue pas de séparation alors soit lui, soit un de ses ancêtre doit prédire une classe.

Les inégalités 1, 2 et 3 étant en très grand nombre (à cause de la dépendance aux données), elles sont ajoutées au cours de la résolution, via l'utilisation du callback. En revanche, l'inégalité 4 est ajoutée dès le début.

Dans le cas d'une résolution multivariée, les inégalités 1 et 3 ne sont plus valide tandis dans l'inégalité 4, $a_{j,t}$ est remplacé par $\hat{a}_{j,t}$. En effet, pour deux données i_1 et i_2 telles que $x_{i_1,j} \geq x_{i_2,j}$, il est possible de bien les classer mêmes si la séparation a en partie lieu sur la caractéristique j.

3.2 Premiers Tests

Nous commençons par essayer simplement avec la nouvelle contrainte 4. On remarque que le temps de résolution est amélioré pour la séparation univariée (entre x2 et x3) mais pas pour multivariée (même ordre de grandeur) sur le jeu de données *Iris*. Sur *Seeds*, les gains sont marginaux pour $D = 2$ ou $D = 3$ mais important pour $D = 4$ pour les deux types de séparation. Cela nous permet de valider son ajout au modèle.

On essaye ensuite de renforcer les contraintes de séparation univariée en appliquant des valeures de M_1 et de M_2 plus petites lorsque c'est possible :

$$M^i_2 = \min(1, \max_{i\in\mathcal{I}, j\in\mathcal{J}|x_{i,j}<1} x_{i,j} + \mu^+) - \min_{j\in\mathcal{J}} x_{i,j}$$

$$M^i_1 = \max_{j\in\mathcal{J}} x_{i,j} + \mu^+ - \min_{i\in\mathcal{I}, j\in\mathcal{J}|x_{i,j}>0} x_{i,j}$$

On remarque que sur le jeu de données *Iris*, pour $D = 2$ et $D = 4$, les performances sont détériorées un petit peu et que pour $D = 3$ elles sont améliorées mais sur le jeu de données *Seeds*, elles ne sont améliorées que pour $D = 2$ et détériorées pour les autres profondeurs. On conclut donc que ces modifications n'apportent pas assez pour les implémenter définitivement.

Concernant l'ajout des inégalités valides pendant la résolution, on commence par trouver la meilleure en les parcourant toutes afin d'appliquer celles la plus violée mais cela est très inefficace. On change alors

pour le choix de la première trouvée aléatoirement qui est violée. La résolution reste malheureusement peu efficace et détériore globalement les performances.

On décide donc de se pencher plus précisément sur nos inégalités valides. La 1 est finalement dominée par la 3, on ajoutera donc seulement cette dernière. La contrainte 2 étant valide pour les deux types de séparation, on se restreint pour l'instant a celle-ci. On décide aussi de deux nouvelles stratégie d'ajout :

— trouver aléatoirement la première donnée pour laquelle une des contrainte est violée et ajouter la contrainte en question pour tous les noeuds et les classes (le cas échéant) de l'arbre.

— trouver aléatoirement le premier couple (noeud, classe) ne respectant pas la contrainte et l'ajouter pour toutes les données concernées.

Nos premiers tests sur les jeux de données *Iris* et *Seeds* nous montre que la deuxième stratégie est la plus efficace, on gardera donc uniquement celle-ci, même si pour l'instant les améliorations apportées reste à la marge par rapport à la résolution sans callback.

Enfin, comme dans l'objectif on ne minimise pas le nombre de séparation effectué par notre arbre, on peut encore renforcé la formulation par une inégalité qui n'est pas valide pour l'ensemble des solutions optimales mais qui l'est pour certaines d'entre elles :

$$\sum_{i \in \mathcal{I}^k} u^i_{t,w} \geq c_{k,t} \qquad \forall t \in \mathcal{L} \quad \forall k \in \mathcal{K}$$

3.3 Résultats des inégalités valides

Nous n'avons ici pas trouvé de mécanisme efficace de séparation de nos inégalités, ce qui est essentiel lorsqu'une famille d'inégalité valides est très grande, et cela explique la détérioration des performances que l'on observe. Finalement, le temps de calcul ainsi que la précision sont améliorés uniquement par les inégalités rajoutées dès le début sans callback.

Les tables 5 et 6 présentent alors dans cette section les résultats de l'approfondissement que nous avons mené.

Instance	D	Uni/Multivarié	Temps (s)	Gap	Erreurs train	Erreurs test
Iris (taille train = 136, taille test = 34)	2	U	2.1	0%	5	1
		M	1.5	0%	1	2
	3	U	13.6	0%	0	2
		M	11.5	0%	0	1
	4	U	24.9	0%	0	2
		M	4.8	0%	0	2
Seeds (taille train = 116, taille test = 29)	2	U	120	2.1%	9	4
		M	1.1	0%	0	2
	3	U	120	4.3%	7	3
		M	1.4	0%	0	1
	4	U	120	1.8%	3	3
		M	6.2	0%	0	5
Wine (taille train = 263, taille test = 66)	2	U	26	0%	5	3
		M	0.5	0%	0	1
	3	U	9.5	0%	0	2
		M	1.6	0%	0	1
	4	U	120	1.4%	2	2
		M	16.4	0%	0	2

TABLE 5 – Résultats sur les instances de bases

Instance	D	Uni/Multivarié	Temps (s)	Gap	Erreurs train	Erreurs test
Etudiants (taille train = 116, taille test = 29)	2	U	73.6	0%	71	19
		M	0.1	0%	55	25
	3	U	120	17.3%	64	19
		M	0.2	0%	55	22
	4	U	120	5.2%	58	23
		M	1.6	0%	55	23
Accent (taille train = 263, taille test = 66)	2	U	120	36%	115	32
		M	120	6.9%	60	24
	3	U	120	76.5%	113	30
		M	120	16.4%	37	20
	4	U	120	139.1%	136	45
		M	120	46.1%	50	29

TABLE 6 – Résultats sur les instances supplémentaires