

Objectif

Montrer le fonctionnement des Listes de Décision (Decision Lists – DL) dans TANAGRA.

Les listes de décision font partie des méthodes d'apprentissage supervisé très en vogue au début des années 90. La méthode est très proche de l'induction par arbres, elle produit des règles de production triées et mutuellement exclusives du type « **Si** condition_1 **Alors** conclusion_1 **Sinon Si** condition_2 **Alors** conclusion_2 **Sinon Si**... ».

Si le biais de représentation est très proche des arbres de décision, les DL construisent des hyper-rectangles les plus homogènes possibles du point de vue de la variable à prédire, le biais de préférence est différent, elles procèdent par « Separate-and-conquer » et non par « Divide-and-conquer », elles disposent donc un degré de liberté supplémentaire pour mettre en avant les solutions les plus spécialisées. Cette liberté a un prix, cette méthode est souvent sujette au sur-apprentissage à force de vouloir détecter à tout prix les sous-populations les plus intéressantes, son paramétrage est alors primordial pour éviter qu'elle ne produise des règles couvrant trop peu d'individus et de ce fait non significatives.

La méthode que nous avons implémentée dans TANAGRA est très largement inspirée de CN2 (Clark & Niblett, ML-1989). Nous avons pris quelques libertés : (1) en simplifiant l'exploration de l'espace des hypothèses, nous utilisons un « hill-climbing » simple plutôt qu'un « best-first search », nettement plus coûteux en CPU mais pas vraiment plus efficace ; (2) en introduisant un paramètre supplémentaire, le support de la règle, pour éviter que des règles sur-spécialisées et non-significatives apparaissent.

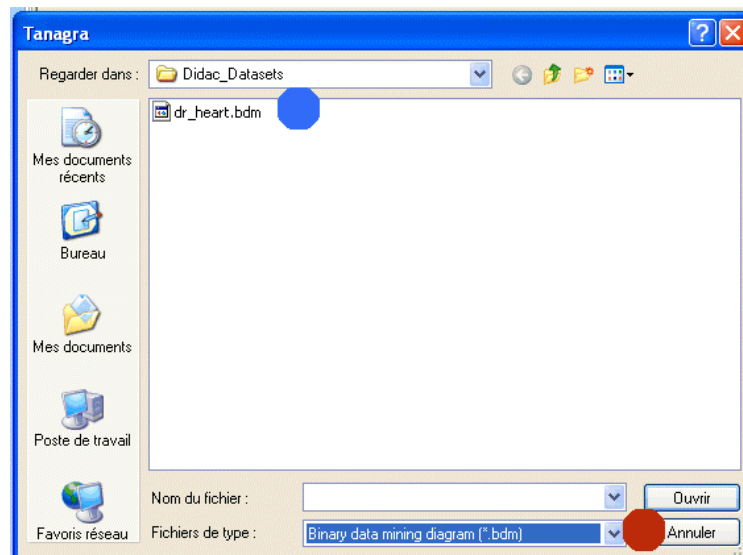
Fichier

Nous utilisons le fichier HEART déjà présenté dans d'autres didacticiels, il s'agit de prédire l'occurrence d'une maladie cardio-vasculaire à partir des caractéristiques des patients.

DECISION LIST

Charger le fichier de données

Ouvrir le fichier de données DR_HEART.BDM à partir du menu « Fichier / Ouvrir ».



Préparation des données

DL ne peut traiter que les descripteurs discrets, il est donc nécessaire de discrétiser au préalable les attributs continus, en utilisant par exemple la méthode MDLPC.

Sélectionnez en tant qu'INPUT tous les descripteurs continus et en tant que TARGET l'attribut CŒUR, placez alors le composant MDLPC (Feature Construction). Vous devez obtenir le résultat suivant.

Define status 1

Target : 1
Input : 6
Illustrative : 0

Attribute	Target	Input	Illustrative
age	-	yes	-
sexe	-	-	-
type_douleur	-	-	-
pression	-	yes	-
cholester	-	yes	-
sucre	-	-	-
electro	-	-	-
taux_max	-	yes	-
angine	-	-	-
depression	-	yes	-
pic	-	yes	-
vaisseau	-	-	-
coeur	yes	-	-

MDLPC 1

Attributes discretized: 6
Examples: 270

Generated attributes

Source	New att	Intervals	Cut points
age	d_mdipc_age_1	2	(54.5000)
pression	d_mdipc_pression_1	1	()
cholester	d_mdipc_cholester_1	1	()
taux_max	d_mdipc_taux_max_1	2	(147.5000)
depression	d_mdipc_depression_1	2	(1.7000)
pic	d_mdipc_pic_1	2	(1.5000)

Computation time : 0 ms.
Created at 22/04/2005 09:03:25

Apprentissage des DL

L'étape suivante consiste à sélectionner les variables adéquates puis placer les composants d'apprentissage.

Mettez en INPUT tous les attributs discrets (y compris les variables discrétisées) à l'exception de la variable CŒUR, placez cette dernière en TARGET.

Placez alors les composants d'apprentissage, le diagramme de traitements correspondant est le suivant.

The screenshot displays a software interface with two main panels. The left panel, titled 'Default title', shows a workflow diagram with the following components: Dataset (heart.txt), Define status 1, ANDLPC 1, Define status 2, and Supervised Learning 1 (Decision List). The right panel, titled 'Results', contains the following information:

Classifier performances

Error rate		0.2074	
Values prediction		Confusion matrix	
Value	Recall	1-Precision	
presence	0.6583	0.1596	
absence	0.9000	0.2330	

	presence	absence	Sum
presence	79	41	120
absence	15	135	150
Sum	94	176	270

Classifier characteristics

Number of rules = 9

Knowledge-based system

Antecedent	Consequent	Distribution
IF vaisseau in [D]	coeur in [presence]	(16; 3)
ELSE IF type_douleur in [B]	coeur in [absence]	(6; 35)
ELSE IF vaisseau in [C] -- engine in [oui]	coeur in [presence]	(15; 0)
ELSE IF sexe in [feminin] -- type_douleur in [C]	coeur in [absence]	(1; 31)
ELSE IF vaisseau in [B]	coeur in [presence]	(35; 9)
ELSE IF type_douleur in [C]	coeur in [absence]	(7; 24)
ELSE IF type_douleur in [A]	coeur in [absence]	(4; 13)
ELSE IF d_mdpc_depression_1 in [m_>=_1.70000005]	coeur in [presence]	(13; 3)
ELSE IF type_douleur in [D]	coeur in [absence]	(23; 32)
ELSE (DEFAULT RULE)	coeur in [absence]	(0; 0)

Computation time : 16 ms.

Evaluation de l'erreur

Pour avoir une idée des performances réelles de la méthode sur ce fichier, nous évaluons l'erreur à l'aide de la procédure BOOTSTRAP, les résultats montrent que le taux d'erreur en généralisation est proche de 24%.

The screenshot shows a software interface with a project tree on the left and a results panel on the right. The project tree includes a dataset 'heart.txt', 'Define status 1', 'MDLPC 1', 'Define status 2', 'Supervised Learning 1 (Decision List)', and 'Bootstrap 1'. The results panel is titled 'Bootstrap 1' and shows 'Replications : 25'. Below this is a table for 'Bootstrap error estimation'.

Error rate	
.632+ bootstrap	0.2359
.632 bootstrap	0.2343
Resubstitution	0.2074
Avg test set	0.2499

Comparaison avec C-RT

En comparaison, la méthode d'induction d'arbres de Breiman et al. (C-RT -- 1984) produit un classifieur avec un taux d'erreur estimé de 24%, notons qu'il n'est pas nécessaire dans ce cas de procéder à une discrétisation préalable des attributs continus, nous avons donc directement placé CŒUR en TARGET et tous les autres attributs en INPUT.

L'arbre de décision est le suivant.

The screenshot shows a software interface with a project tree on the left and a 'Tree description' panel on the right. The project tree includes a dataset 'heart.txt', 'Define status 1', 'MDLPC 1', 'Define status 2', 'Supervised Learning 1 (Decision List)', 'Bootstrap 1', 'Define status 3', and 'Supervised Learning 2 (C-RT)'. The 'Tree description' panel includes the following information:

Tree description

Number of nodes	7
Number of leaves	4

Decision tree

- type_douleur in [D]
 - depression < 0.5500
 - vaisseau in [D,B,C] then coeur = **presence** (73.33 % of 3 examples)
 - vaisseau in [A] then coeur = **absence** (73.68 % of 19 examples)
 - depression >= 0.5500 then coeur = **presence** (90.91 % of 55 examples)
- type_douleur in [C,B,A] then coeur = **absence** (78.02 % of 53 examples)

Computation time : 16 ms.
Created at 22/04/2005 09:20:48

L'estimation de l'erreur par BOOTSTRAP correspond au diagramme de traitements suivant.

The screenshot displays a software interface for a machine learning project. On the left, a project tree shows the following structure:

- Dataset (heart.txt)
 - Define status 1
 - MDLPC 1
 - Define status 2
 - Supervised Learning 1 (Decision List)
 - Bootstrap 1
 - Define status 3
 - Supervised Learning 2 (C-RT)
 - Bootstrap 2

On the right, the 'Bootstrap 2' results panel is shown, including a 'Parameters' section with 'Replications : 25' and a 'Results' section titled 'Bootstrap error estimation'. A table displays the error rates:

Error rate	
.632+ bootstrap	0.2406
.632 bootstrap	0.2370
Resubstitution	0.1963
Avg test set	0.2607

Below the table, the 'Computation time : 531 ms.' is indicated.

La plupart des tests à grande échelle menés sur des données artificielles et réelles ont montré que les listes de décision et les arbres de décision présentaient des performances similaires.

Cette méthode est très peu connue en dehors du cercle des chercheurs en apprentissage automatique, essentiellement parce que, à ma connaissance, elle n'a jamais été implémentée dans des logiciels de grande diffusion.