Objectif

Construire un arbre de classification avec TANAGRA.

La classification consiste à construire des groupes homogènes d'observations, des classes, du point de vue d'une série de descripteurs de telle sorte que les observations dans les mêmes classes soient le plus similaire possible, et que les observations dans des classes différentes soient le plus dissemblable possible.

Une fois les groupes construits, il nous faut d'une part pouvoir les interpréter, savoir sur quelles caractéristiques les individus d'un même groupe ont été placés ensemble, qu'est-ce que différencient les groupes ; d'autre part, disposer d'une procédure d'affectation qui nous permettra de classer rapidement un nouvel individu dans un des groupes.

Les arbres de classification permettent de répondre très simplement à ces deux exigences. Le modèle de classement est représenté par un arbre de décision, chaque groupe est décrit par une règle logique, l'algorithme détecte automatiquement les variables pertinentes dans leur élaboration, l'interprétation est immédiate. L'affectation à un groupe peut être réalisée simplement en appliquant le système logique sur les descripteurs de l'observation à classer.

Cette technique est peu connue, les principales références sont à ce jour les articles de Chavent[1] (1998) et Blockeel[2] (1998).

Fichier

Nous travaillerons sur le fichier ZOO (UCI). Il s'agit de regrouper des animaux selon leurs caractéristiques. Ils sont par ailleurs classés en 7 familles par des spécialistes.

Nous vérifierons si notre classification concorde peu ou prou avec cette typologie proposée par les biologistes. Nous vérifierons également si notre classification, qui intègre quand même une contrainte forte, la construction d'un arbre logique pour représenter les classes, concorde avec les résultats produits par les méthodes classiques telles que les K-MEANS.

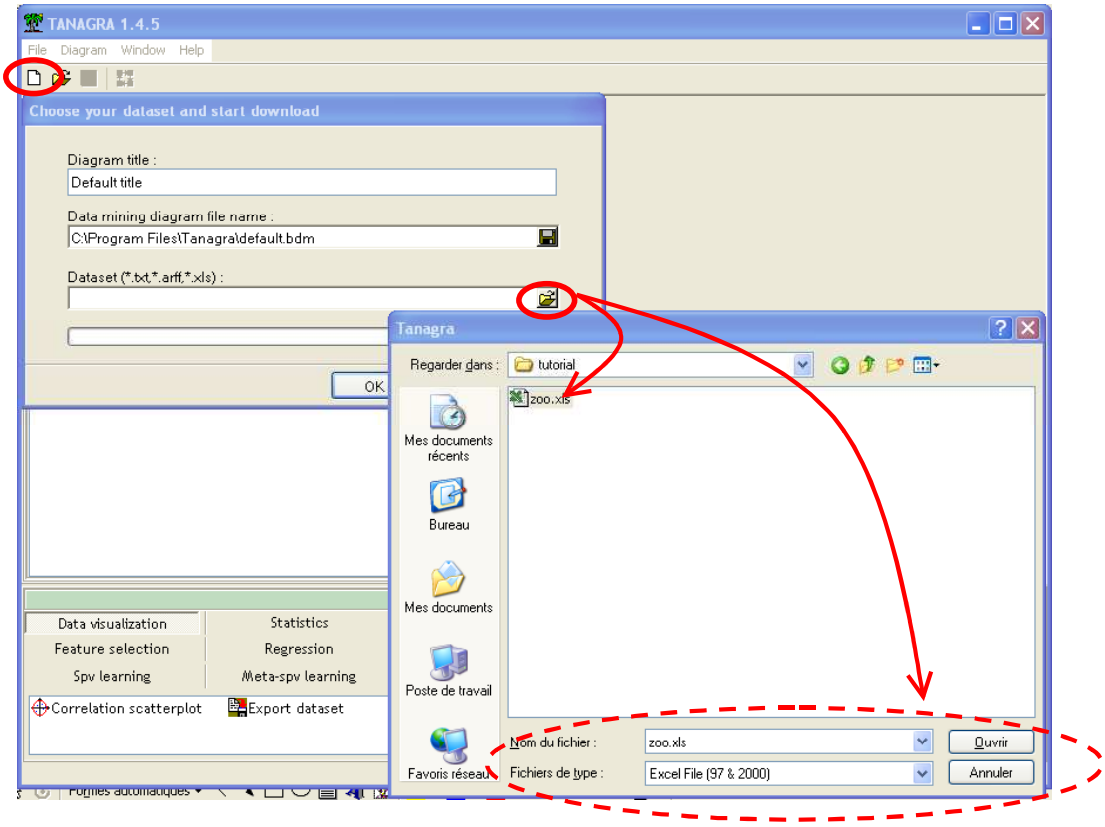
[1] M. Chavent (1998), « A monothetic clustering method », Pattern Recognition Letters, 19, 989—996.

[2] H. Blockeel, L. De Raedt, J. Ramon (1998), « Top-Down Induction of Clustering Trees », ICML, 55—63.

Arbre de classification avec TANAGRA

Charger les données

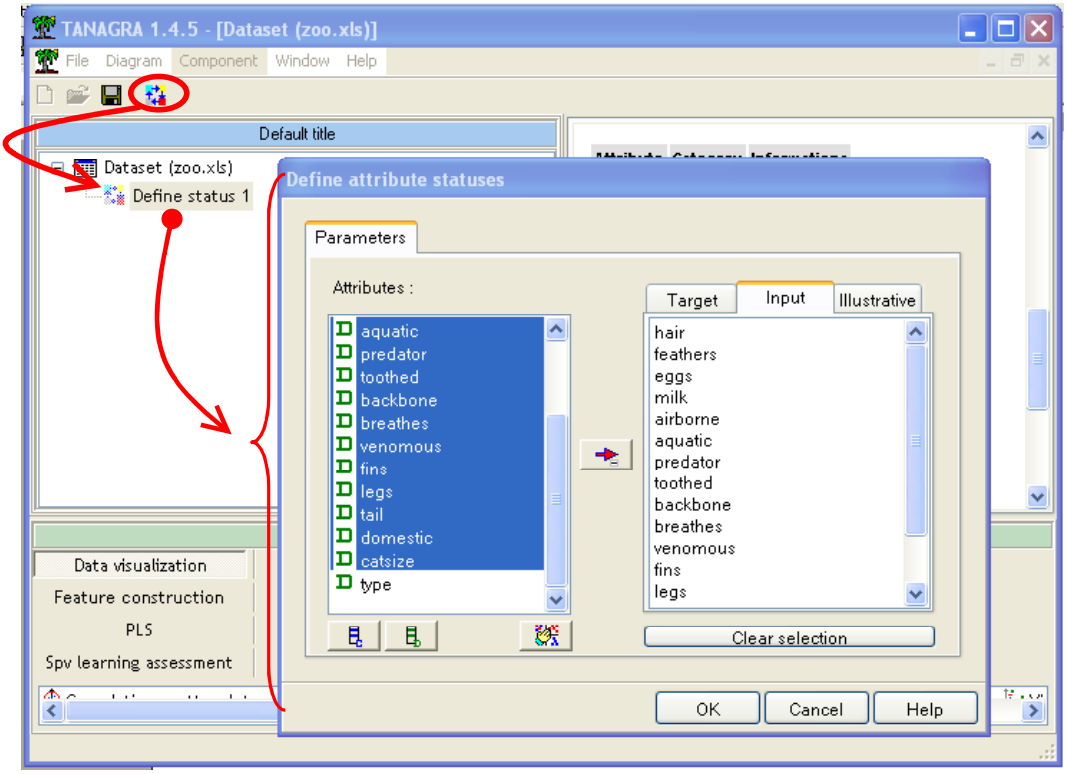
Nous devons dans un premier temps créer un diagramme et charger les données. Pour ce faire, nous cliquons sur le menu FILE/NEW. Nous sélectionnons le fichier ZOO.XLS.



Choix des variables actives

L'étape suivante consiste à définir les variables que nous utiliserons pour caractériser l'homogénéité des groupes. Nous sélectionnons toutes les variables, sauf celle qui a été fournie par l'expert (TYPE), que nous utiliserons comme variable illustrative. L'idée justement est de vérifier s'il existe une classification naturelle des animaux qui correspond à la classification experte.

Nous ajoutons donc le composant DEFINE STATUS dans le diagramme en utilisant le raccourci de la barre d'outil.



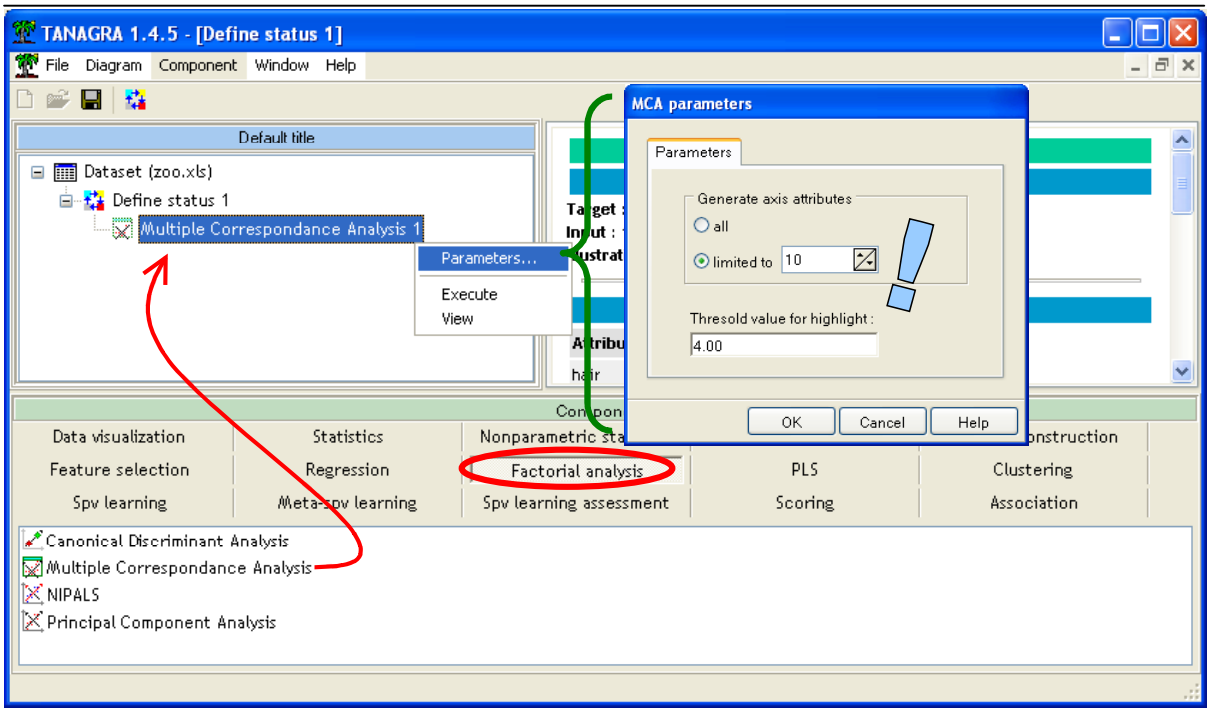
Préparer les données avec une analyse factorielle

Calculer des similarités directement sur des variables catégorielles pose plusieurs types de problèmes. Il est souvent plus intéressant de préparer les données en les projetant dans un nouvel espace de représentation à l'aide d'une analyse factorielle.

Les données étant discrètes, nous utiliserons l'analyse des correspondances multiples, nous cumulons ainsi plusieurs avantages : notre espace est continu, nous pouvons utiliser des distances plus familières telles que la distance euclidienne simple ; d'autant plus que les axes factoriels sont par construction deux à deux indépendants ; enfin, en ne sélectionnant que les 10 premiers axes, nous récupérons l'information « utile » et laissons de côté l'information « bruitée » spécifique au fichier.

Nous plaçons une ACM dans notre diagramme, nous le paramétrons de manière à ce que 10 axes factoriels soit produits (soit à peu près la moitié du nombre des axes factoriels que l'on peut produire).

Remarque : Dans le cas où nos variables auraient été toutes continues, cette étape de préparation peut également s'avérer bénéfique. En effet, le passage par une analyse factorielle (Analyse en Composantes Principales dans ce cas) permet de « lisser » les données, l'algorithme de typologie subséquente peut ainsi se concentrer sur l'information essentielle contenue dans les données.



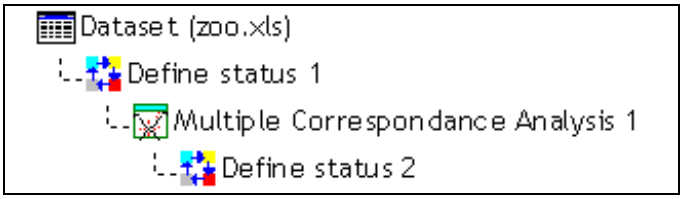
En cliquant sur VIEW, nous produisons le résultat suivant. Nous constatons que les dix premiers axes factoriels résument 90% de l'information disponible, ce qui est amplement suffisant.

Eigen values

Axis	Eigen value	% explained	Histogram	% cumulated
1	0.325530	26.04%		26.04%
2	0.235564	18.85%		44.89%
3	0.174092	13.93%		58.81%
4	0.086711	6.94%		65.75%
5	0.073008	5.84%		71.59%
6	0.062644	5.01%		76.60%
7	0.052588	4.21%		80.81%
8	0.043289	3.46%		84.27%
9	0.042704	3.42%		87.69%
10	0.032932	2.63%		90.33%
11	0.027850	2.23%		92.55%

Sélection des variables pour la construction de l'arbre

Nous pouvons maintenant lancer l'algorithme de classification. Nous voulons construire des groupes similaires sur l'espace des axes factoriels en utilisant les variables qui décrivent les animaux. Pour ce faire, nous ajoutons de nouveaux un composant DEFINE STATUS, nous plaçons en TARGET les 10 axes factoriels, et en INPUT nos variables initiales.



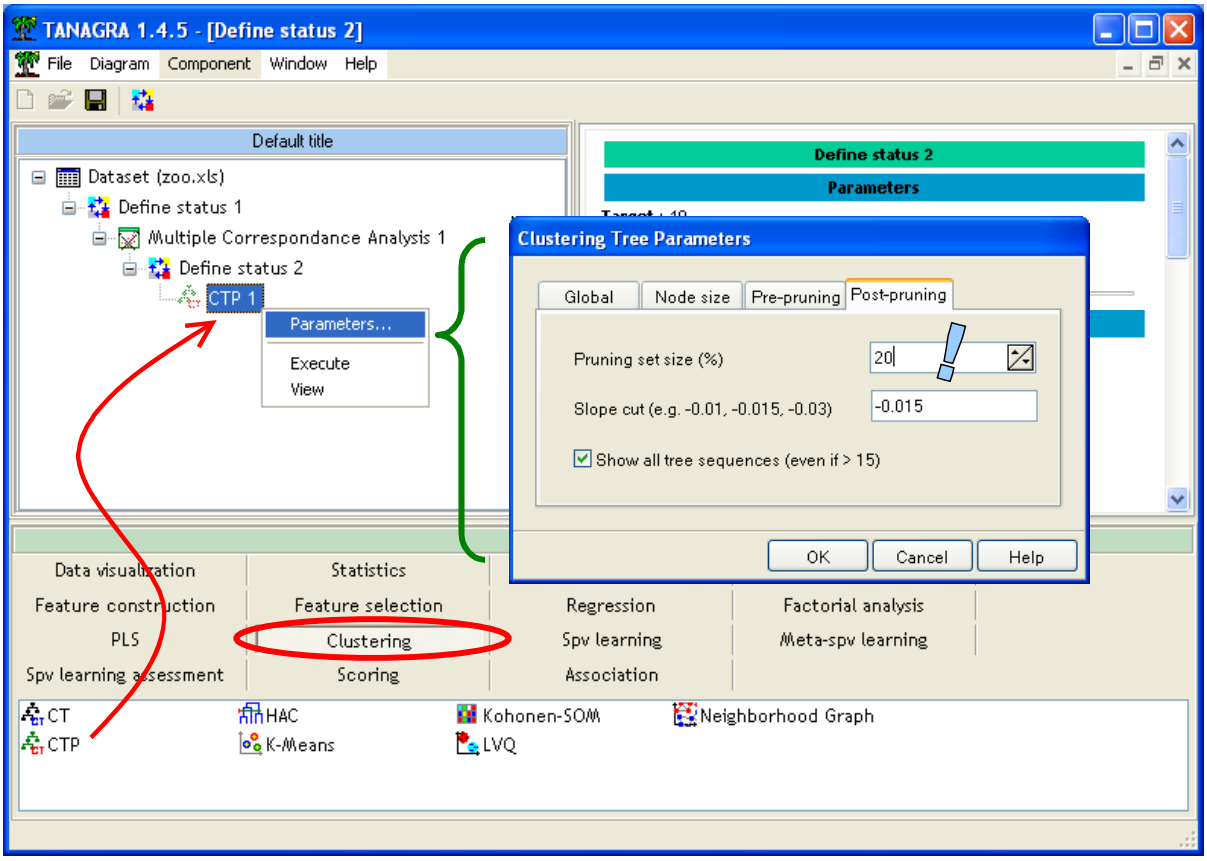
L'exécution du composant (Menu VIEW) donne l'affichage suivant.

Attribute	Target	Input	Illustrative
hair	-	yes	-
feathers	-	yes	-
eggs	-	yes	-
milk	-	yes	-
airborne	-	yes	-
aquatic	-	yes	-
predator	-	yes	-
toothed	-	yes	-
backbone	-	yes	-
breathes	-	yes	-
venomous	-	yes	-
fins	-	yes	-
legs	-	yes	-
tail	-	yes	-
domestic	-	yes	-
catsize	-	yes	-
type	-	-	-
MCA_1_Axis_1	yes	-	-
MCA_1_Axis_2	yes	-	-
MCA_1_Axis_3	yes	-	-
MCA_1_Axis_4	yes	-	-
MCA_1_Axis_5	yes	-	-
MCA_1_Axis_6	yes	-	-
MCA_1_Axis_7	yes	-	-
MCA_1_Axis_8	yes	-	-
MCA_1_Axis_9	yes	-	-
MCA_1_Axis_10	yes	-	-

Remarque : Dans cet exemple, nous utilisons les mêmes variables pour construire l'espace qui nous permet de caractériser les groupes (les axes factoriels) et pour les décrire (les variables qui définissent les segmentations de l'arbre de classification). Nous ne sommes pas tenus de nous conformer systématiquement à ce schéma. Il se peut très bien que dans certains cas, nous voulons définir des classes homogènes sur un ensemble de variables (par exemple, des comportements d'achats) et de les décrire avec un autre ensemble de variables (par exemple, des variables socio-économiques telles que le revenu, la catégorie socioprofessionnelle, le nombre d'enfants à charge, etc.). Dans ce cas, nous parlerons plutôt d'arbre de décision multi-cible (« Predictive Clustering Tree »).

Arbre de classification

Enfin nous pouvons placer le composant Arbre de Classification situé dans l'onglet CLUSTERING. Nous sélectionnons la méthode CTP (CLUSTERING TREE WITH PRUNING).

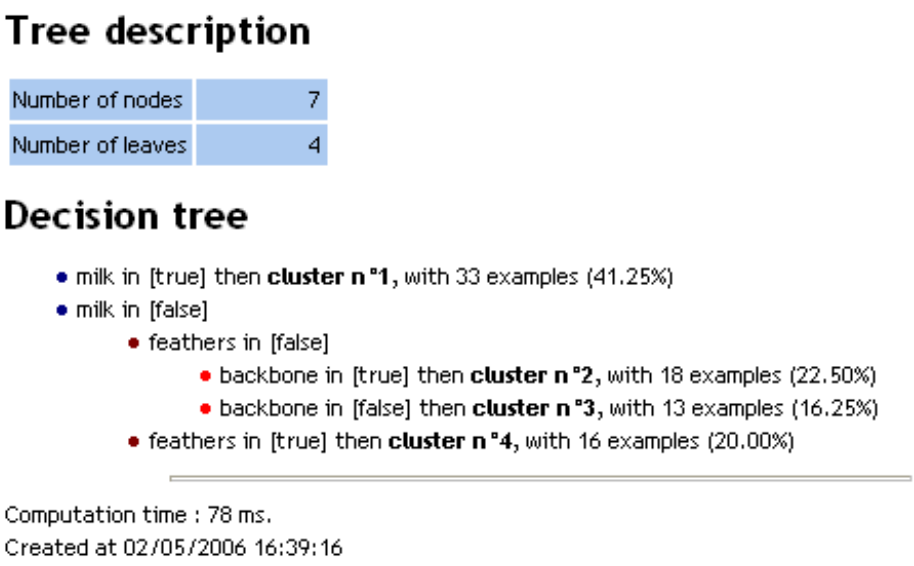


Pour décrire brièvement la méthode, nous dirons qu'il s'agit d'une généralisation des arbres de régression (Méthode CART, Breiman et al., 1984) avec deux points de différenciation :

1. Le critère qui permet de définir le choix de la variable de segmentation ne repose plus sur la décomposition de la variance mais sur la décomposition de l'inertie.
2. Il ne s'agit plus ici de produire une prédiction précise mais plutôt de détecter des groupes naturels. En d'autres termes, nous essayons de repérer automatiquement le « coude » dans la courbe de l'inertie intra-classes calculée sur les données d'élagage (pruning set), nous utilisons à cet effet une régression sur 3 points successifs, dès que la pente est proche de zéro, nous estimons que l'adjonction d'une nouvelle classe n'est plus pertinente[3].

Dans notre exemple, nous subdivisons notre échantillon en deux parties : la première (growing set) sert à l'expansion des nœuds, la seconde (pruning set : 20 % ici) sert au post-élagage de l'arbre.

Nous activons le menu VIEW pour obtenir la typologie.



L'arbre de décision propose une typologie en 4 classes (le nombre de feuilles de l'arbre). Les classes sont décrites avec les règles suivantes :

Si milk = true Alors Cluster 1
Si milk = false et feathers = false et backbone = true alors Cluster 2
Si milk = false et feathers = false et backbone = false alors Cluster 3
Si milk = false et feathers = true alors Cluster 4

[3] Cette approche est très simpliste, elle a le mérite de l'automaticité. Le mieux reste la production du graphique montrant l'évolution de l'inertie intra-classes selon le nombre de groupes, et de repérer visuellement, comme nous le montrons plus loin, les « cassures » dans la courbe.

Le principal intérêt de cette approche est que l'interprétation et l'affectation des observations aux groupes sont très simplifiées. Nous constatons que :

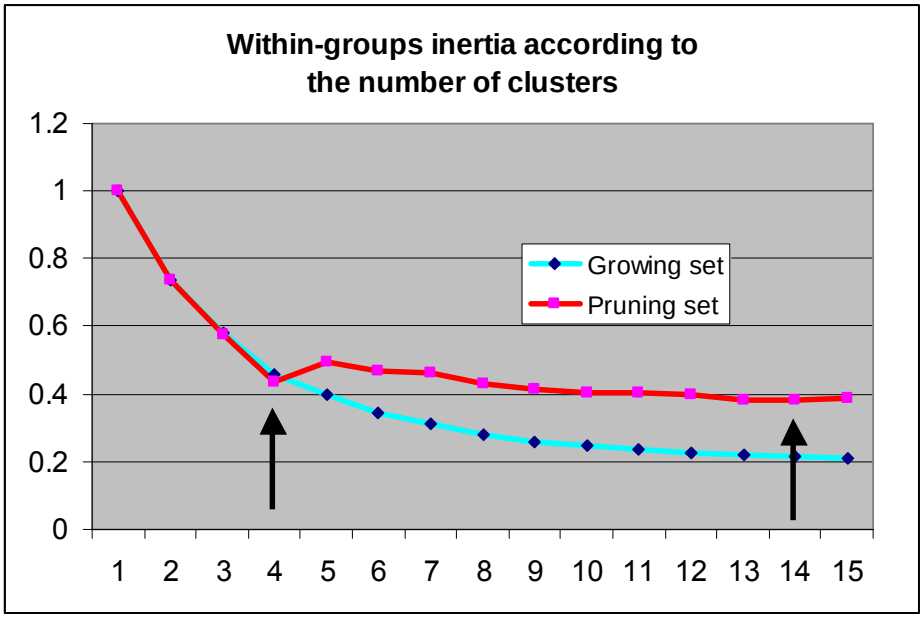
- Le premier groupe est constitué des animaux produisant du lait (Cluster 1).
- Le second groupe correspond à des animaux ne produisant pas de lait, qui n'ont pas de plume, et qui ont une épine dorsale (Cluster 2).
- Le troisième groupe représente les animaux ne produisant pas de lait, n'ont ni de plumes ni d'épine dorsale (Cluster 3).
- Le quatrième groupe correspond aux animaux qui ne produisent pas de lait mais qui ont des plumes (Cluster 4).

Nous disposons également de la décroissance de l'inertie intra-classes selon le nombre de feuilles de l'arbre. C'est une information importante car la classification étant hiérarchique, il nous est possible, si besoin est, de définir manuellement le nombre de classes dans l'algorithme. Nous disposons de cette information à la fois sur la partie growing set et la partie pruning set des données.

Trees sequence (# 15) -- Inertia Within-Groups

N°	# Leaves	Inertia (growing set)	Inertia (pruning set)
15	1	1.0000	1.0000
14	2	0.7389	0.7378
13	3	0.5809	0.5769
12	4	0.4564	0.4337
11	5	0.3992	0.4938
10	6	0.3470	0.4696
9	7	0.3104	0.4625
8	8	0.2791	0.4290
7	9	0.2598	0.4128
6	10	0.2453	0.4032
5	11	0.2348	0.4027
4	12	0.2274	0.3982
3	13	0.2203	0.3826
2	14	0.2134	0.3809
1	15	0.2082	0.3862

La typologie en 14 classes minimise l'inertie intra-classes sur l'ensemble d'élagage (en vert), en revanche une cassure dans l'évolution du critère se produit à la constitution de la partition en 4 classes (en rouge). Le graphique associé montre clairement cette évolution.

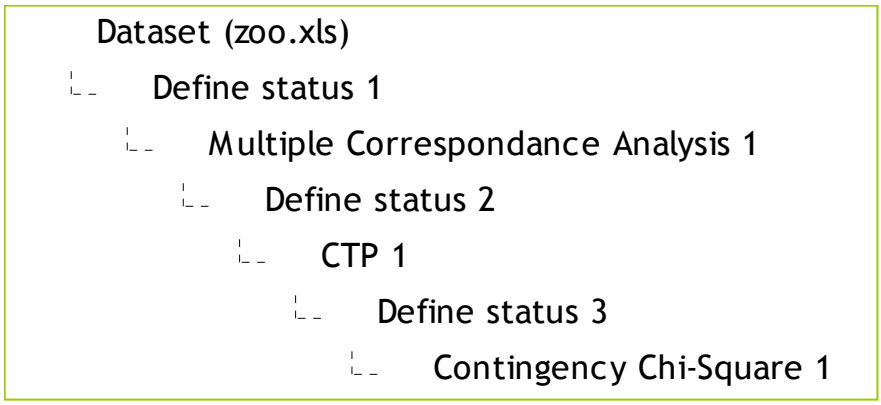


Validation des résultats – Confrontation avec l'expertise

Pour valider cette approche peu connue dans le Data Mining, nous décidons dans un premier temps de la confronter avec les connaissances du domaine. Elle n'est malheureusement pas toujours disponible, dans notre cas qui est un cas d'école, cette information est bien commode pour évaluer la qualité de la classification automatique.

L'expert propose une typologie en 7 groupes, nous voulons la croiser avec notre classification en 4 groupes pour en évaluer la pertinence. Pour cela, nous ajoutons de nouveau un composant DEFINE STATUS dans le diagramme, nous plaçons en TARGET la variable TYPE fournie par les experts, en INPUT la variable CLUSTER_CTP_1 fournie par la méthode. Puis nous insérons à la suite un composant CONTINGENCY CHI-SQUARE (onglet NONPARAMETRIC STATISTICS) pour croiser les deux classifications.

Le diagramme est le suivant.



Son exécution montre que les deux typologies sont complètement cohérentes.

Contingency Chi-Square 1

Parameters

Cross-tab parameters	
Sort results	non
Input list	Target (Row) and input (Column)
Contribution thresold	2.0

Results

Row (Y)	Column (X)	Statistical indicator		Cross-tab					
		Stat	Value		c_ct_1	c_ct_2	c_ct_3	c_ct_4	Sum
		Tschuprow's t	0.840896	mammal	41 (+0.12)	0	0	0	41
		Cramer's v	1.000000	fish	0	13 (+0.12)	0	0	13
		Phi²	3.000000	bird	0	0	0	20 (+0.21)	20
type	Cluster_CTP_1	Chi²	303.000000	invertebrate	0	0	10 (+0.13)	0	10
		Pr(Chi²)	0.000000	insect	0	0	8 (+0.10)	0	8
				amphibian	0	4	0	0	4
				reptile	0	5	0	0	5
				Sum	41	22	18	20	101

Computation time : 0 ms.

Chaque classe de l'expert a été placée dans un seul cluster. Et la classification automatique a produit des groupes soit purs (Cluster 1 et Cluster 4), soit mélangeant des espèces proches[4] (Cluster 2 et Cluster 3).

Validation des résultats – Confrontation avec une autre méthode

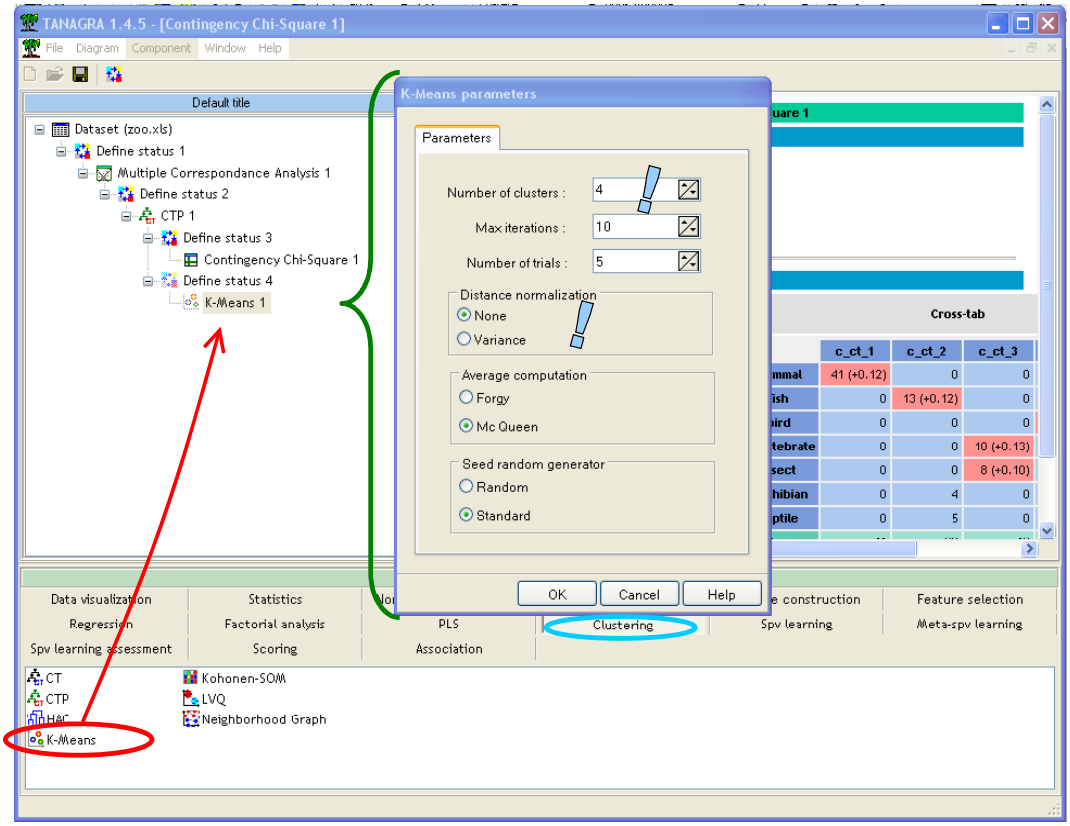
L'arbre de classification intègre des contraintes fortes lors de sa construction : la connaissance est nécessairement représentée par un arbre (biais de représentation), la recherche se fait pas à pas, en considérant le rôle individuel de chaque variable lors de chaque segmentation (biais d'apprentissage).

Notre idée ici est de confronter nos résultats avec ceux obtenus avec les K-MEANS, une méthode très connue et dont les qualités ont été maintes fois étudiées.

Nous complétons donc notre diagramme en rajoutant un composant DEFINE STATUS sous l'arbre de classification (CTP 1), nous plaçons en INPUT tous les axes factoriels; puis nous insérons le composant K-MEANS, que nous paramétrons de manière à ce que les résultats

[4] Autant que puisse en juger le néophyte que je suis !

des deux approches (arbre et k-means) soient comparables : nous voulons 4 clusters, les variables (les axes factoriels) ne doivent pas être normalisées.



La typologie explique près de la moitié de l'inertie.

Results

Clustering results

Clusters	4

Cluster	Description	Size
cluster n°1	c_kmeans_1	24
cluster n°2	c_kmeans_2	20
cluster n°3	c_kmeans_3	18
cluster n°4	c_kmeans_4	39

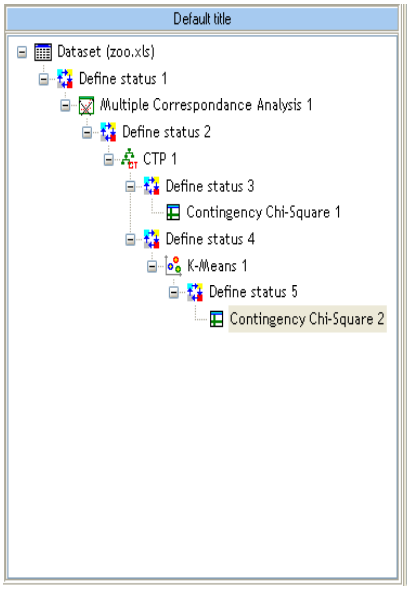
Ratio explained evolution

Number of trials	5

Trial	Ratio explained
1	0.398329
2	0.372209
3	0.477354
4	0.375221
5	0.493703

Reste à confronter cette typologie avec celle produite par l'arbre de classification.

Nous ajoutons dans le diagramme un composant DEFINE STATUS, nous plaçons en TARGET la classification de l'arbre (CLUSTER_CTP_1), en INPUT celle des K-MEANS (CLUSTER_KMEANS_1). Puis nous insérons le composant tableau croisé pour apprécier la concordance des classes proposées par les deux méthodes.



Contingency Chi-Square 2

Parameters

Cross-tab parameters	
Sort results	non
Input list	Target (Row) and input (Column)
Contribution thresold	2.0

Results

Row (Y)	Column (X)	Statistical indicator		Cross-tab					
		Stat	Value		c_kmeans_1	c_kmeans_2	c_kmeans_3	c_kmeans_4	Sum
Cluster_CTP_1	Cluster_KMeans_1	Tschuprow's t	0.957184	c_ct_1	3	0	0	38	41
		Cramer's v	0.957184	c_ct_2	21 (+0.17)	0	0	1	22
		Phi²	2.748604	c_ct_3	0	0	18 (+0.25)	0	18
		Chi²	277.608957	c_ct_4	0	20 (+0.23)	0	0	20
		Pr(Chi²)	0.000000	Sum	24	20	18	39	101

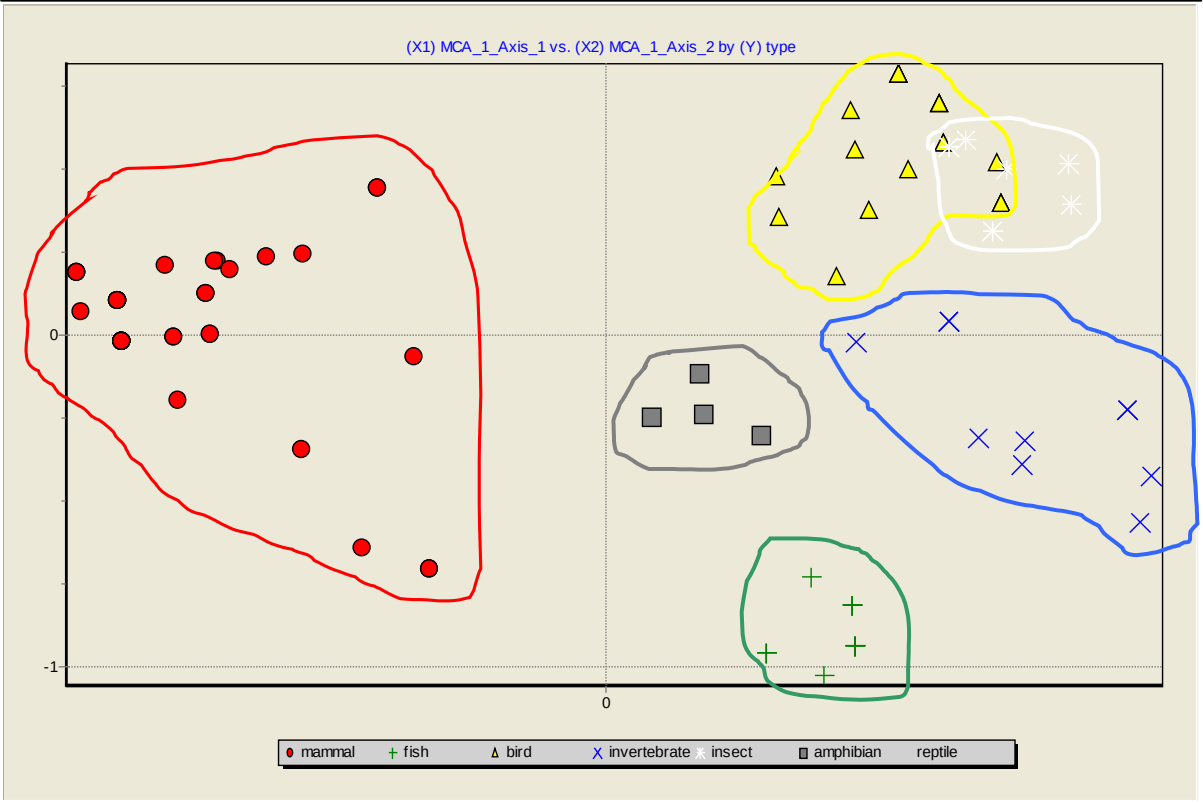
Computation time : 0 ms.
Created at 03/05/2006 08:55:57

Nous constatons que les deux typologies sont cohérentes, il y a une correspondance *quasi*-exacte entre les groupes proposés par l'arbre et les K-MEANS.

De manière générale, nous avons constaté, dans les très nombreuses expérimentations que nous avons mené pour éprouver cette méthode, que le gain en compréhensibilité des résultats apporté par l'arbre, tant en termes de représentation des connaissances qu'en termes de sélection de variables pertinentes pour la caractérisation des classes, n'est pas contrebalancée par une perte significative de qualité du partitionnement.

Visualisation des groupes

Pour situer les groupes d'expert dans notre espace de description des données, nous avons voulu représenter nos points dans le premier plan factoriel, qui traduit rappelons-le près de la moitié de l'information disponible, nous les avons illustrés selon leur groupe d'appartenance fournie par l'expert. L'idée est de voir si certains groupes ne se démarquent pas visuellement.



Finalement, les deux premiers axes factoriels permettaient déjà de discerner les groupes. Nous les distinguons bien.

Cela montre bien que les outils visuels telle que l'analyse factorielle emmènent des perspectives très enrichissantes dans l'exploration des données. Encore faut-il savoir interpréter les résultats et pouvoir se ramener à l'espace des variables initiales. La lecture des tableaux des cosinus carrés et des contributions peut se révéler rapidement insurmontable pour les praticiens qui ne sont pas familiarisés avec ces outils.