MIXMOD : un logiciel de classification supervisée et non supervisée pour données quantitatives et qualitatives

Florent Langrognet

MIXMOD : un logiciel de classification supervisée et non supervisée pour données quantitatives et qualitatives

Florent Langrognet

Laboratoire de Mathématiques de Besançon
UMR6623, CNRS & Université de Franche-Comté
16, route de Gray, 25030 Besançon, France

Résumé MIXMOD, logiciel de classification supervisée et non supervisée, traite des données multidimensionnelles en utilisant la richesse des modèles de mélanges. Il propose de nombreuses fonctionnalités (large palette de modèles gaussiens et multinomiaux, d'algorithmes, de critères de sélection, ...) et repose sur une architecture particulièrement adaptée pour atteindre un haut niveau de performance (temps de calcul et robustesse). MIXMOD est téléchargé environ 250 fois par mois, et il est utilisé dans des situations très diverses à la fois dans le milieu de la recherche et, pour une part croissante, par des utilisateurs novices. Des évolutions sont régulièrement proposées pour répondre aux demandes des utilisateurs et intégrer les résultats de la recherche.

Keywords : classification des données, logiciel, modèles de mélanges.

1 Introduction

Par leur flexibilité, les modèles finis de distributions de probabilité sont très utiles pour modéliser une grande variété de phénomènes aléatoires et sont naturellement considérés comme un outil de choix pour traiter des problématiques de classification supervisée et non supervisée. Utilisant ce cadre de travail, le logiciel MIXMOD est un logiciel adapté à de nombreuses situations, y compris dans des situations complexes. Il peut être utilisé dans les environnements Scilab et Matlab et est disponible, sous licence GNU GPL, pour les systèmes d'exploitation Linux et Windows.
Ce article ne vise pas à remplacer le guide de l'utilisateur ni la documentation statistique que l'on peut trouver sur le site web dédié à MIXMOD. Il a pour but de présenter ce logiciel (son historique, son architecture, les différentes façons de l'utiliser) et de montrer, sur des exemples concrets, ses principales fonctionnalités et l'intérêt d'utiliser un tel logiciel dans des situations diverses (données quantitatives ou qualitatives par exemple).

2 Fiche d'identité

2.1 Historique

Le projet MIXMOD est né en 2001 avec pour objectif de développer un outil efficace, rapide et robuste pour traiter des problématiques de classification de données en utilisant les modèles de mélange (voir [11]). Sous l'impulsion des quatre auteurs principaux

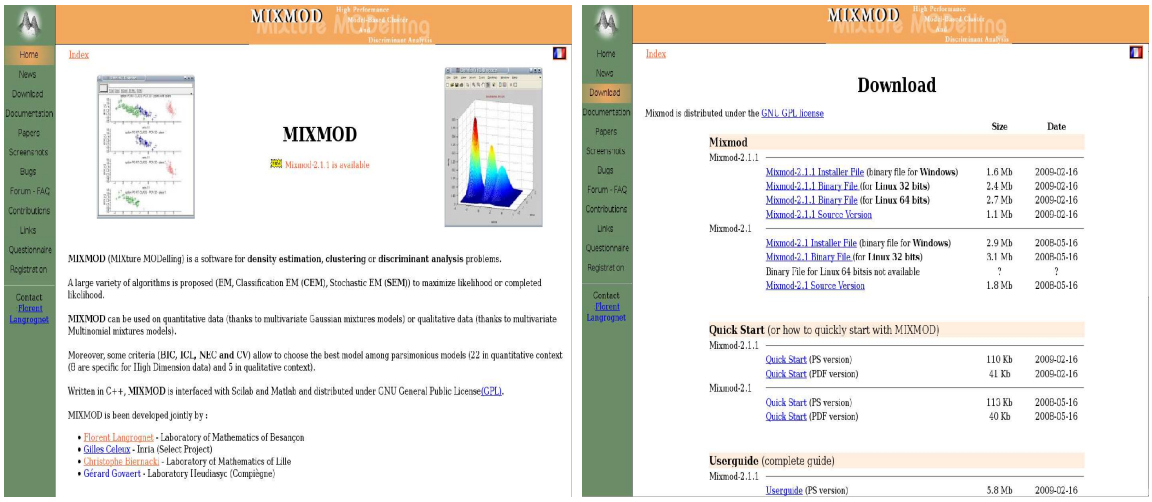


FIG. 1 – Pages 'Home' et 'Download' du site web consacré à MIXMOD.

(C. Biernacki, professeur à l'université de Lille1, G. Celeux, directeur de recherche à l'INRIA, G. Govaert, professeur à l'Université Technologique de Compiègne et F. Langrognet, ingénieur de recherche CNRS), les différentes versions de MIXMOD se sont succédées à un rythme d'environ 2 sorties par an. Il s'agissait d'intégrer des évolutions fonctionnelles, des améliorations en terme de rapidité de calcul ou de meilleures outils graphiques (pour une utilisation dans les environnements Scilab et Matlab notamment).

- Déc. 2001 : MIXMOD 1.0 ;
- Fév. 2007 : MIXMOD 2.0 (traitement des données qualitatives par modèles multinomiaux) ;
- Mai 2008 : MIXMOD 2.1 (traitement des données quantitatives de grandes dimensions par modèles spécifiques (HDDA)) ;
- Fév. 2009 : MIXMOD 2.1.1.

2.2 Distribution

Le site web MIXMOD (*www-math.univ-fcomte.fr/mixmod/* et prochaînement *www.mixmod.org*), consacré au logiciel propose différentes rubriques (en anglais et en français) :

- *Téléchargement* des différents packages (versions binaire, source pour windows et linux) ;
- *Documentations*. Trois types de documentations sont disponibles :
  - documentations pour l'utilisateur final : *userguide* et *quickstart* ;
  - documentation statistique ;
  - documentation pour le développement ;
- *Bugs* ;
- *News* ;
- *FAQ* ;
- ...

Ce site web connaît depuis 2001 un nombre croissant de visites et de téléchargements (aujourd'hui, on compte environ 600 visites et 250 téléchargements par mois).
MIXMOD est distribué sous licence GNU GPL (donc avec le code source) autorisant les

utilisateurs qui le souhaitent à adapter le logiciel à leurs besoins spécifiques. Ce type de licence est particulièrement adapté au domaine de la recherche et a, sans doute, favorisé l'essor de MIXMOD qui a pu ainsi être utilisé et mis à l'épreuve dans des conditions très diverses (type de données, taille des échantillons, ...). Cependant, dans certaines situations particulières, la licence GNU GPL n'est pas adaptée. Dans ce cas et sur demande, MIXMOD peut être distribué sous une autre licence (dans le cadre d'une intégration dans un logiciel dont la licence n'est pas GNU GPL par exemple).

Les 5 instituts de recherche impliqués soutiennent le développement de MIXMOD, avec, entre autres, des dépôts à l'APP (Agence pour la Protection des Programmes), un soutien financier pour le recrutement d'ingénieurs en CDD et de stagiaires et d'autres actions de valorisation.
Une *rencontre MIXMOD* est organisée tous les 2 ans, lieu privilégié d'échanges entre utilisateurs et développeurs. La dernière rencontre MIXMOD qui s'est déroulée à Lille en décembre 2008 a rassemblé une cinquantaine de personnes et a permis de nouer des contacts très enrichissants.

3 Une architecture logicielle adaptée aux utilisateurs et aux utilisations

Historiquement MIXMOD s'est d'abord adressé aux spécialistes de la classification en leur proposant un outil paramétrable dans un environnement de travail adapté à leurs besoins et leurs habitudes. C'est donc naturellement que des outils pour utiliser MIXMOD dans les environnements Scilab et Matlab ont été développés. Ainsi, dès 2002, MIXMOD est utilisable à travers ces logiciels grâce à :

- une interface graphique (pour Scilab et Matlab),
- des fonctions dédiées (pour Scilab et Matlab). Parmi ces fonctions, la fonction *mixmodView* permet de visualiser graphiquement les données et les résultats issus de MIXMOD (voir captures d'écran : Fig. 2).

Ainsi, deux offres complémentaires sont proposées répondant aux besoins des utilisateurs : l'interface graphique pour l'utilisateur standard et les fonctions dédiées pour l'utilisateur souhaitant paramétrer plus finement les options de MIXMOD.

L'architecture de MIXMOD (cf. Fig. 3) est donc organisée autour de sa bibliothèque de calcul (plus de 30000 lignes de C++) particulièrement optimisée pour atteindre de hautes performances en terme de temps de calcul. Un effort constant est engagé en ce sens depuis 2001 et chaque nouveau développement est réalisé en tenant compte de cet objectif. Les fonctions pour Scilab et Matlab représentent, quant à elles, environ 20000 lignes de code.

Depuis 2001, MIXMOD s'est de plus en plus ouvert à des utilisateurs novices (non experts en statistiques) souhaitant disposer d'un outil puissant pour le traitement de leurs problématiques de classification. Ces utilisateurs ont naturellement recours aux interfaces graphiques sous Scilab et Matlab mais, pour un nombre croissant d'entre eux, le fait que MIXMOD ne dispose pas de sa propre interface graphique est perçu comme un frein à son utilisation. Une réflexion a donc été engagée en ce sens et le développement d'une interface graphique indépendante de Scilab et Matlab a débuté en 2008 (voir section *Perspectives*).

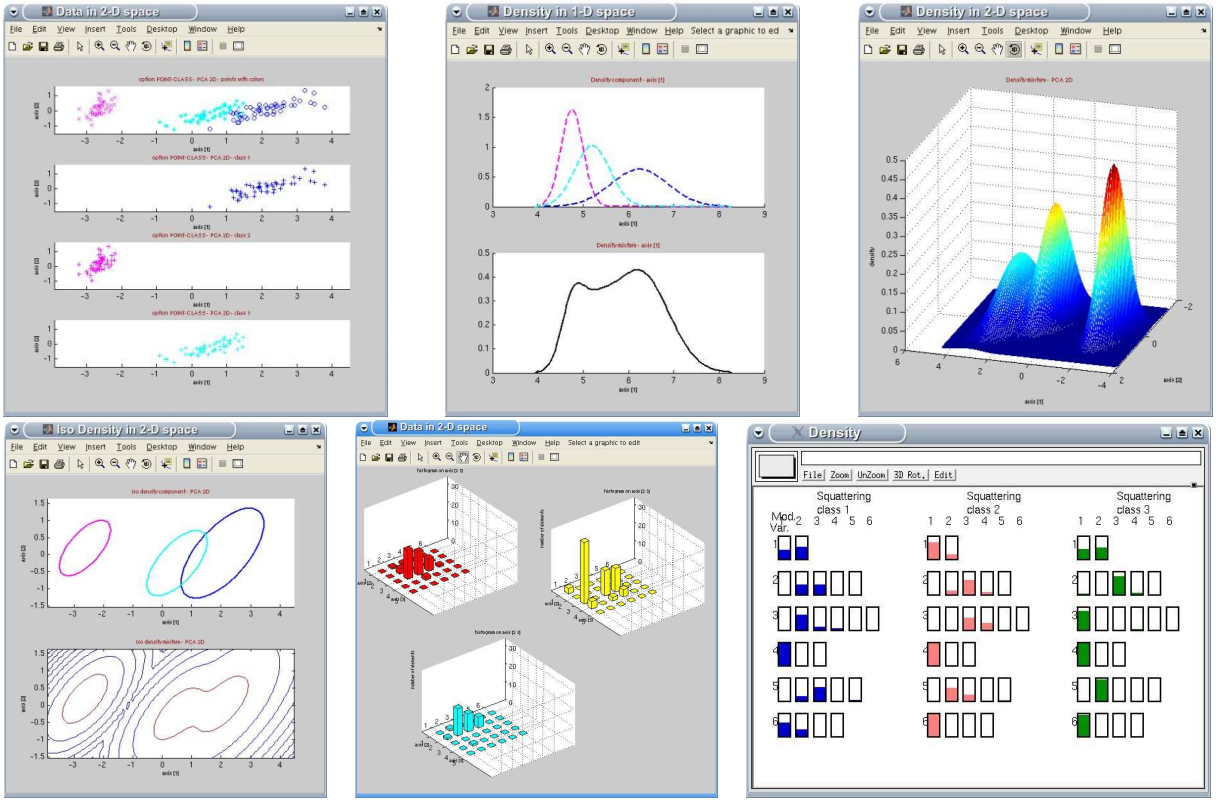


FIG. 2 – Captures d'écran de la fonction mixmodView (visualisation des résultats dans les environnements Scilab et Matlab.)

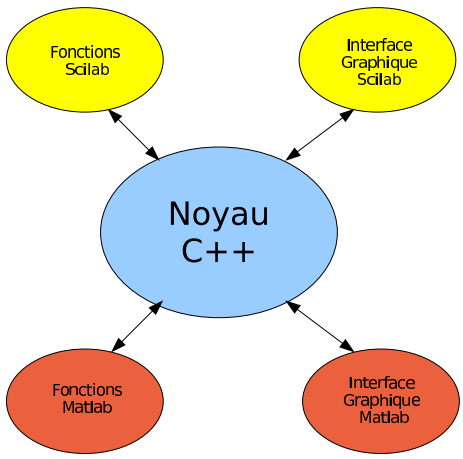


FIG. 3 – Architecture logicielle de MIXMOD organisée autour du noyau de calcul.

4 Principales fonctionnalités

Le but de cet article n'est pas de décrire de manière exhaustive l'ensemble des fonctionnalités et des possibilités de MIXMOD qui sont décrites (avec leurs fondements mathématiques) dans la documentation de l'utilisateur, la documentation statistique (toutes deux disponibles sur le site web de MIXMOD) ou dans les articles [3] et [2].
Il s'agit de citer les principales fonctionnalités dont certaines seront plus détaillées dans la section *Illustrations* :

- Des modèles parcimonieux pour modéliser finement :
  - 14 modèles gaussiens pour le traitement des données quantitatives basés sur la décomposition en valeurs singulières des matrices de variance (voir [9]).
  - 8 modèles spécifiques pour le traitement des données quantitatives en grande dimension (voir [5]).
  - 5 modèles multinomiaux pour le traitement des données qualitatives basés sur une reparamétrisation de la distribution de Bernoulli (voir [7]).
- Des algorithmes pour maximiser la vraisemblance (ou la vraisemblance complétée) : EM, SEM, CEM, M (voir [10], [6], [8]).
- 6 modes d'initialisation et la possibilité de chaîner des algorithmes pour essayer d'atteindre le maximum global de la vraisemblance.
- 4 critères de sélection de modèles
  - BIC (Bayesian Information Criterion)
  - ICL (Integrated Completed likelihood)
  - NEC (Normalized Entropy Criterion)
  - CV (Cross Validation)
- Pondération des individus.
- ...

5 Illustrations

Le but de cette section est de montrer l'intérêt de MIXMOD et son utilisation dans les environnements Scilab et Matlab sur deux exemples concrets.

5.1 Classification non supervisée sur données quantitatives

5.1.1 Données et problématiques

Le premier exemple concerne une problématique de classification non supervisée sur un jeu de données quantitatives en dimension 2 (cf. Fig 4).
On suppose ici que l'on connaît le nombre de classes (3) et que l'on dispose d'informations sur la dispersion des classes (variances de même orientation et volumes libres) et leurs proportions (égales) permettant d'utiliser le modèle gaussien $[p_k \lambda_k C]$.
L'objectif est ici de :

- classer les individus,
- caractériser les 3 classes.

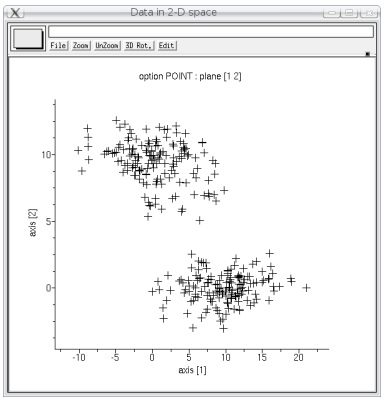


FIG. 4 – Jeu de données de la première illustation : 300 individus en dimension 2.

5.1.2 Une première utilisation de MIXMOD

Dans l'environnement Scilab (ou Mtalb), on peut utiliser MIXMOD à l'aide de l'interface graphique ou, ce qui sera montré ici, via les fonctions dédiées : mixmod.sci et mixmodView.sci. Les seules entrées obligatoires de la fonction *mixmod* sont les données et le nombre de classes. La sortie de cette fonction sert ensuite à la fonction *mixmodView* qui permet de visualiser les résultats graphiquement.
Ainsi, les commandes :
data = read('mesDonnees.dat', 300, 2) ;
out = mixmod(data, 3) ;
permettent de réaliser une classification des données en 3 groupes.
L'ensemble des résultats est disponible dans la variable *out* que l'on utilisera ensuite pour visualiser (cf. Fig. 5) :

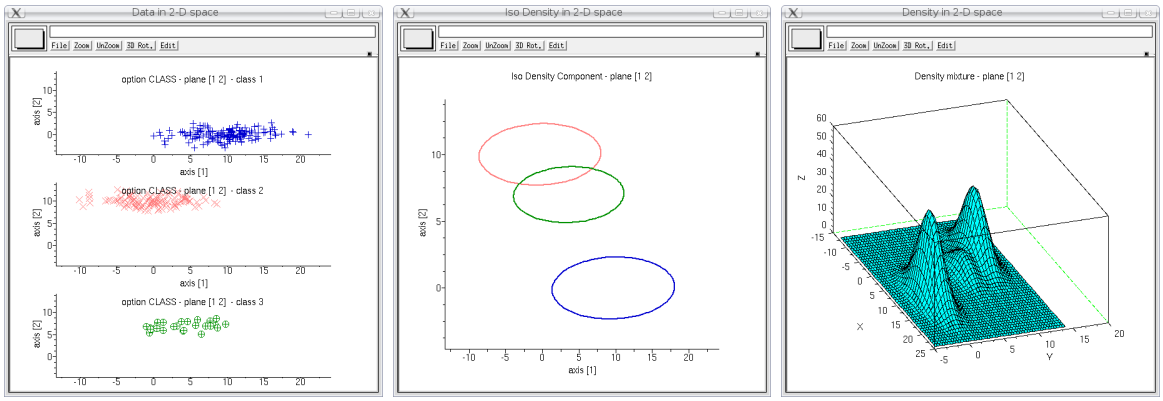
- les individus classés par groupe (cf. Fig. 5 (a) obtenue grâce à la commande *mixmodView(out, [1 2], class)*) ;
- les courbes d'iso-densités de chaque classe (cf. Fig. 5 (b) obtenue grâce à la commande*mixmodView(out, [1 2], 'isoDensityComponent')*) ;
- la densité du mélange (cf. Fig. 5 (c) obtenue grâce à la commande*mixmodView(out, [1 2], 'densityMixture')*).

Les proportions associées à chacune des classes sont respectivement de 0.5, 0.4 et 0.1.

Ainsi, l'utilisation de la fonction *mixmod* avec les options par défaut permet d'obtenir une réponse à la problématique : une classification des données et une caractérisation des classes.

5.1.3 Comment obtient-on ces résultats ?

A ce stade, une des premières questions qui se pose est de savoir comment on obtient ces résultats, autrement dit, quelles sont les valeurs par défaut de MIXMOD ? Parmi les options de MIXMOD, on peut citer ici la possibilité de mettre en concurrence plusieurs modèles et de sélectionner le meilleur modèle grâce à l'un des critères proposés. Cette possibilité ne sera pas étudiée ici car, le modèle gaussien et le nombre de classes font



(a) (b) (c)

FIG. 5 – Visualisation des résultats avec la fonction *mixmodView*.

partie des hypothèses (modèle $[p_k \lambda_k C]$ et 3 classes).
En revanche, la notion de *stratégie* sera détaillée. Une *stratégie*, au sens de MIXMOD, représente la description des étapes choisies pour mener à bien une classification. Elle est constituée :

- d'une méthode d'initialisation,
- d'une succession d'algorithmes (au moins un) avec, pour chacun, une règle d'arrêt.

Concernant les méthodes d'initialisation, on dispose des possibilités suivantes :

- si on dispose d'une information a priori :
  - méthode '*USER*' si l'information concerne l'estimation des paramètres du modèle (dispose t-on d'une information sur les proportions, les moyennes, les dispersions ?),
  - méthode '*USER_PARTITION*' si on a une connaissance partielle de certains labels (par exemple les individus x et y appartiennent à la même classe).
- si on ne dispose pas d'information a priori :
  - méthode *RANDOM* : meilleure configuration (au sens maximum de la vraisemblance) de n tirages au hasard d'individus pour initialiser les centres (par défaut $n = 5$),
  - méthode *SMALL_EM* : meilleure configuration de n tirages au hasard suivis de m itérations le l'algorithme EM (avec $m <= 10$ et le nombre total d'itérations de EM durant l'initialisation = 50),
  - méthode *SEM* : meilleure configuration parmi les n étapes de l'algorithme SEM après tirage au hasard ($n = 500$ par défaut),
  - méthode *CEM* : meilleure configuration de n tirages au hasard suivis de m itérations de CEM ($n = 10$ et $m = 50$ par défaut).

Les règles d'arrêt des algorithmes disponibles sont :

- après un nombre donné d'itérations (pour EM, SEM et CEM).
- à la stationnarité de la vraisemblance (pour EM) ou de la vraisemblance complétée (pour CEM). Cette possibilité n'est évidemment pas offerte pour SEM.
- après un nombre donné d'itérations ou à la stationnarité de la vraisemblance (pour EM) ou de la vraisemblance (complétée pour CEM).

Enfin, il est possible de chaîner plusieurs algorithmes.

Ainsi, il est par exemple possible d'appliquer la stratégie suivante :
– initialisation par *SMALL_EM*
– 100 itérations de *SEM*
– itérations de *EM* jusqu'à la stationnarité de la vraisemblance.

Les valeurs de la stratégie par défaut de MIXMOD sont :
– initialisation par *RANDOM*
– algorithme *EM* arrêté après 200 itérations ou à la stationnarité de la vraisemblance.

5.1.4 Algorithmes de type *EM* et initialisations

Les algorithmes de type *EM* sont de puissants outils pour maximiser la vraisemblance mais présentent quelques défauts bien connus (voir [1]) comme :
– une forte dépendance vis à vis de l'initialisation,
– une convergence vers un maximum local.

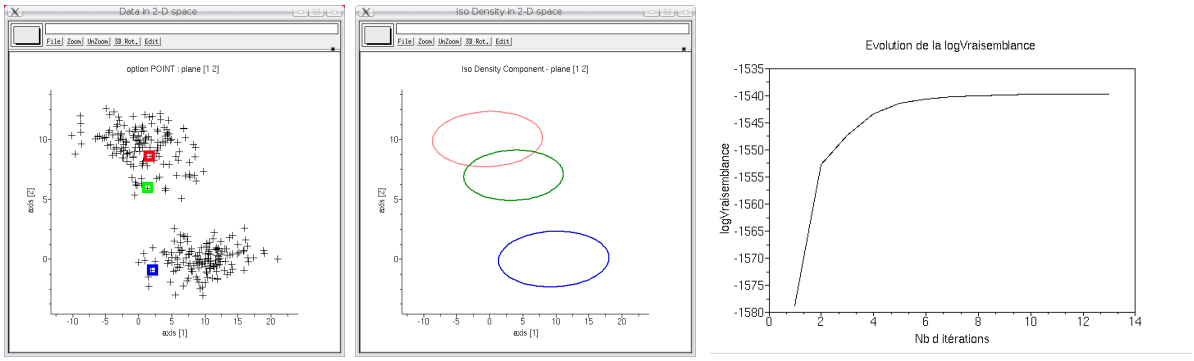
La large palette d'outils dont dispose l'utilisateur dans MIXMOD (méthodes d'initialisation et algorithmes) sont très précieux pour éviter les écueils des algorithmes de type *EM*. Afin d'illustrer les problèmes potentiels liés à l'utilisation de ce type d'algorithme, on propose l'étude du scénario suivant : on tire au hasard (seulement une fois) 3 individus afin d'initialiser les centres des 3 classes puis on lance l'algorithme EM (jusqu'à la stationnarité de la vraisemblance).

Impact de l'initialisation sur la vitesse de convergence

Cette initialisation par tirage au hasard peut tout d'abord avoir un effet non négligeable sur la vitesse de converge de l'algorithme.
Les deux initialisations (cf. Fig. 6 (a) et Fig. 7 (a)) réalisées mènent au même résultat (même classification, mêmes paramètres et même valeur de vraisemblance) mais avec des vitesses de convergence différentes. Dans la première situation (cf. Fig. 6 (a)), les 3 individus tirés au hasard sont à la fois relativement éloignés les uns des autres et suffisamment proches des centres des classes que l'on recherche pour permettre à l'algorithme de converger très rapidement (cf. Fig. 6 (c)) vers la solution décrite par la figure 6 (b). Dans la seconde situation (cf. Fig. 7 (a)), les 3 individus tirés au hasard, bien que relativement proches les uns des autres, permettent à l'algorithme de converger vers la même solution (cf. Fig. 7 (b)) mais nettement plus lentement (cf. Fig. 7 (c)).

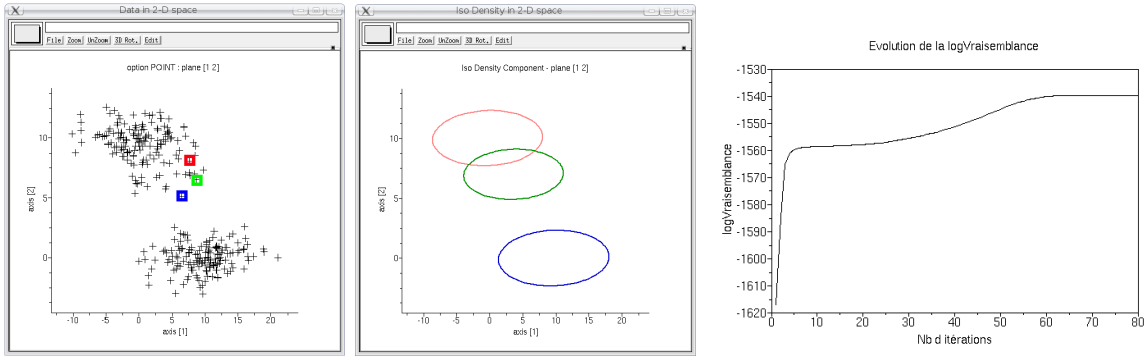
Impact de l'initialisation sur la convergence (vers des maxima différents)

La situation peut être encore plus défavorable avec une initialisation au hasard : dans la troisième situation (cf. Fig. 8 (a)), les individus tirés au hasard sont très proches les uns des autres et l'algorithme EM converge vers une autre solution (cf. Fig. 8 (b)).
La comparaison des valeurs de la vraisemblance permet cependant de considérer cette dernière solution comme étant moins satisfaisante mais cette situation est une illustration de la convergence de l'algorithme EM vers un minimum local.
Les outils disponibles dans MIXMOD pour éviter ces situations sont donc précieux pour éviter les maxima locaux.



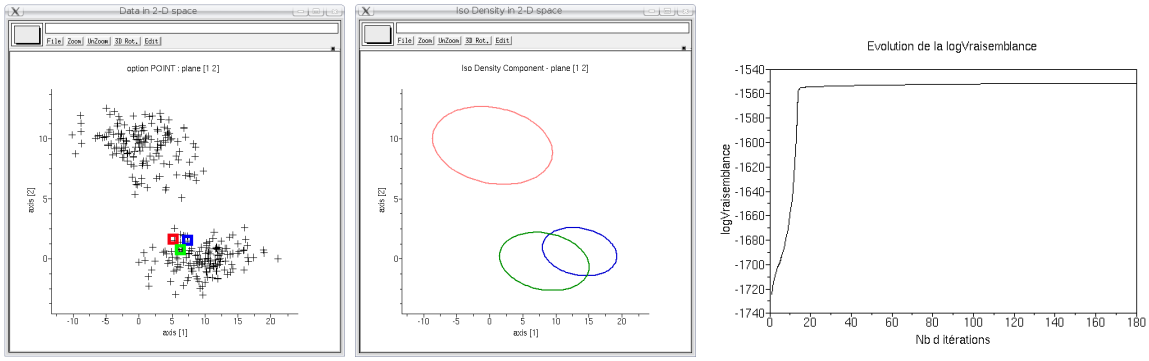
(a) (b) (c)

FIG. 6 – Première initialisation (a) menant à la solution (b) en 10 itérations (c) ($LogVraisemblance = -1539$).



(a) (b) (c)

FIG. 7 – Deuxième initialisation (a) menant à la solution (b) en 80 itérations (c) ($LogVraisemblance = -1539$).



(a) (b) (c)

FIG. 8 – Troisième initialisation (a) menant à la solution (b) en 150 itérations (c) ($LogVraisemblance = -1539$).

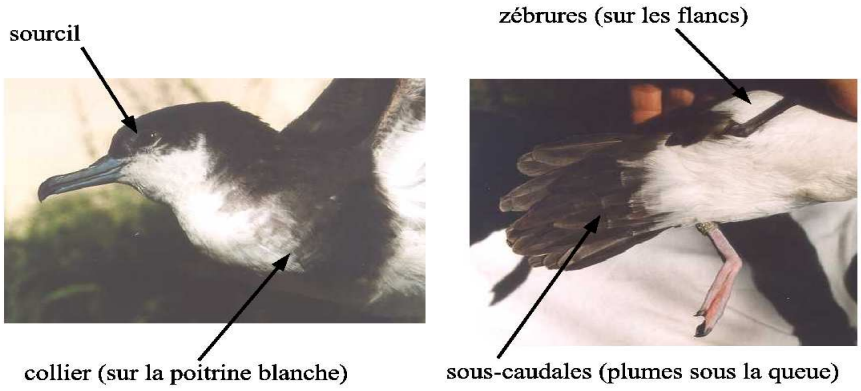


FIG. 9 – Mesures morphologiques des oiseaux.

variable	nombre de niveaux de réponse	valeurs
sexe	2	mâle, femelle
sourcils	5	absent, ... très prononcé
collier	6	absent, ... continu
zébrures	3	absent, peu , présence forte
sous-caudales	5	blanc, noir, noir&blanc, noir&BLANC, NOIR&blanc
liseret	4	absent, ..., beaucoup

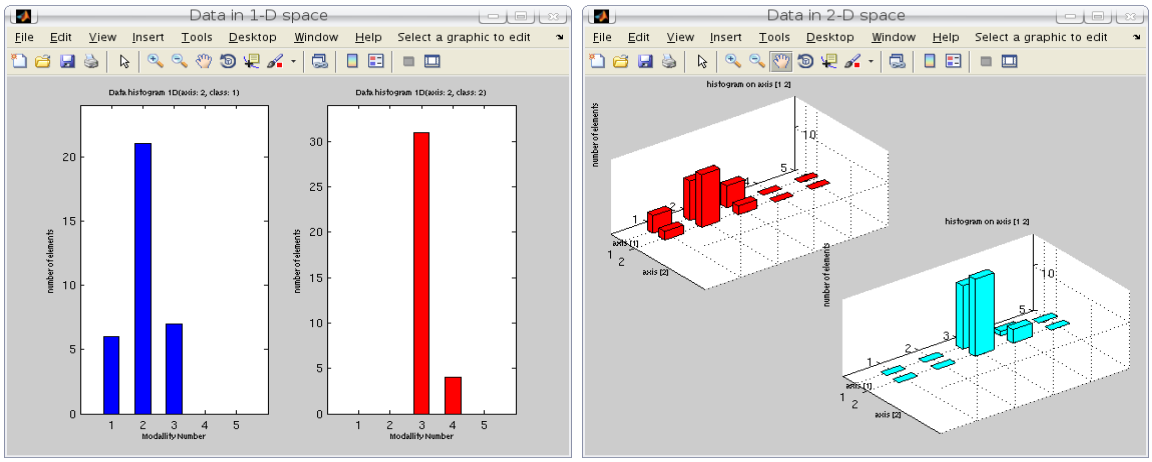
TAB. 1 – Nombre de niveaux de réponse des six variables décrivant chaque oiseau.

5.2 Classification supervisée sur données qualitatives

5.2.1 Données et problématiques

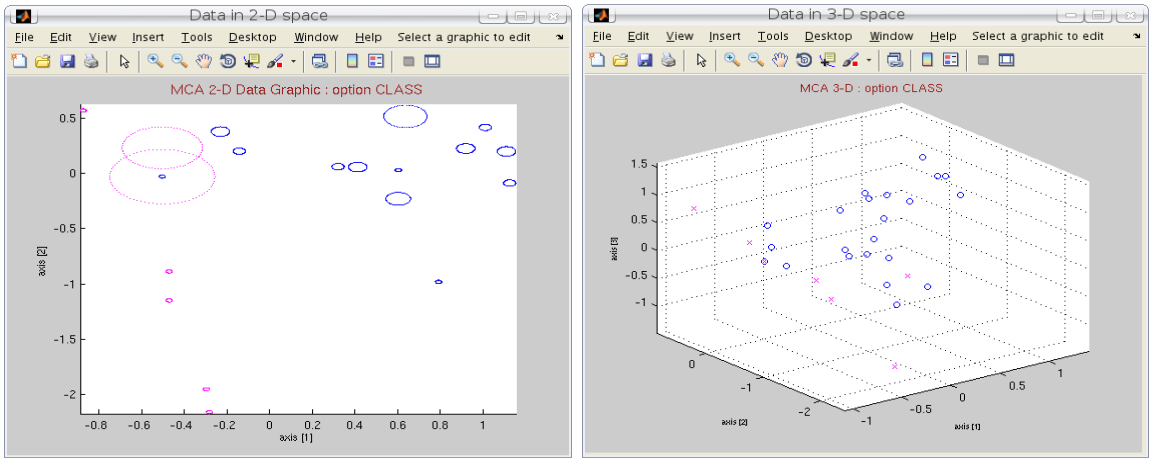
Le jeu de données concerne ici des oiseaux (puffins) sur lesquels six mesures morphologiques ont été effectuées (cf. Fig. 9). Chaque individu (un oiseau) est donc décrit par 6 variables qualitatives dont le nombre de niveaux de réponse est représenté dans le tableau 1.
On dipose d'un *échantillon d'apprentissage* de 69 individus répartis en 2 classes (les sous-espèces *dichrous* et *lherminieri*) et l'objectif est de trouver une règle de classement pour *connaître* la sous-espèce de tout nouvel individu.

Les outils de visualisation des données MIXMOD disponibles sous Scilab et Matlab permettent par exemple de réaliser des graphiques de type *barplots* en dimension 1 (cf. Fig. 10 (a)) ou en dimension 2 (cf. Fig. 10 (b)) et *scatterplots* en dimension 2 (cf. Fig. 11 (a)) ou en dimension 3 (cf. Fig. 11 (b)).



(a) (b)

FIG. 10 – Représentation des données par histogrammes sur l'axe [2] (a) et sur les axes [1 2] (b).



(a) (b)

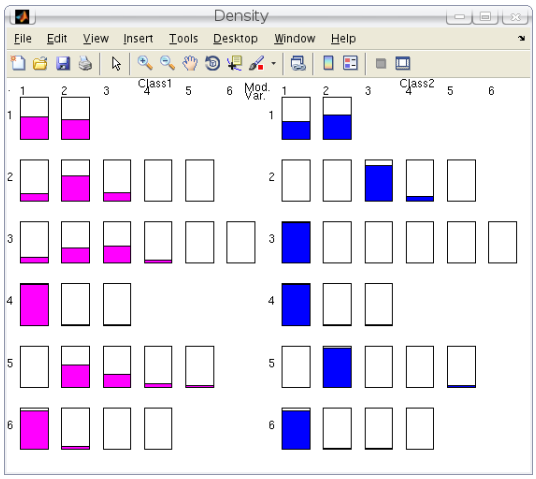
FIG. 11 – Représentation des données dans le premier plan de l'ACM (a) et sur les trois premiers axes de l'ACM (b).

$$\hat{p}_1 = 0.49, \quad \hat{p}_2 = 0.51$$

$$\hat{\alpha}_1 = \begin{bmatrix} \hat{\alpha}_1^1 = & 0.53 & 0.47 & & & & \\ \hat{\alpha}_1^2 = & 0.18 & 0.61 & 0.20 & 0.00 & 0.00 & \\ \hat{\alpha}_1^3 = & 0.15 & 0.37 & 0.41 & 0.06 & 0.00 & 0.00 \\ \hat{\alpha}_1^4 = & 0.98 & 0.01 & 0.01 & & & \\ \hat{\alpha}_1^5 = & 0.00 & 0.55 & 0.32 & 0.10 & 0.04 & \\ \hat{\alpha}_1^6 = & 0.94 & 0.06 & 0.00 & 0.00 & & \end{bmatrix}$$

$$\hat{\alpha}_2 = \begin{bmatrix} \hat{\alpha}_2^1 = & 0.43 & 0.57 & & & & \\ \hat{\alpha}_2^2 = & 0.00 & 0.00 & 0.87 & 0.13 & 0.00 & \\ \hat{\alpha}_2^3 = & 1.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\ \hat{\alpha}_2^4 = & 0.98 & 0.01 & 0.01 & & & \\ \hat{\alpha}_2^5 = & 0.00 & 0.97 & 0.00 & 0.00 & 0.03 & \\ \hat{\alpha}_2^6 = & 0.94 & 0.03 & 0.03 & 0.00 & & \end{bmatrix}$$

(a)



(b)

FIG. 12 – Paramètres estimés (a) et leur vue synthétique (b).

5.2.2 Règle de classement

La première étape d'une classification supervisée consiste à chercher une règle de classement sur la base de l'échantillon d'apprentissage. Autrement dit, il s'agit d'appliquer la fonction M disponible dans MIXMOD permettant l'estimation du maximum de vraisemblance des paramètres du mélange (avec le modèle multinomial par défaut, le plus général, $[p_k \varepsilon_k^{jh}]$). Ceci s'effectue aisément avec MIXMOD dans les environnements Scilab ou Matlab que ce soit avec les fonctions dédiées ou l'interface graphique.
Les paramètres estimés sont représentés sur la figure 12 (a) et graphiquement (avec la fonction mixmodView) sur la figure 12 (b).
Rappel : le terme α_k^{jh} représente la probabilité que la variable x^j prenne la modalité h lorsque l'individu $x = (x^1, \ldots, x^d)$ appartient à la classe k.

A ce stade, il convient d'évaluer la règle de classement obtenue, ce qui peut être réalisé par (au moins) deux critères :

- taux de reclassement par *Validation Croisée* : 97.0% (67 individus bien reclassés sur les 69).
- taux de reclassement par MAP (Maximum A Posteriori) : 98.5% (68 individus bien reclassés sur les 69).

Le reclassement par MAP étant plus optimiste que celui par validation Croisée, son taux est donc, sans surprise, supérieur à celui par Validation Croisée. Quoiqu'il en soit, ces taux sont suffisamment élevés pour accorder de la crédibilité à l'estimation des paramètres et, ce faisant, à la règle de classement obtenue.

5.2.3 Classement d'un nouvel individu

La deuxième étape d'une classification supervisée consiste à appliquer la règle de classement obtenue sur de nouveaux individus pour les classer.
Ici, il s'agit de trouver la sous-espèce (la classe) d'un nouvel individu, un oiseau, pour lequel on ne dispose que des 6 mesures morphologiques ayant pour valeurs (1, 3, 1, 1,

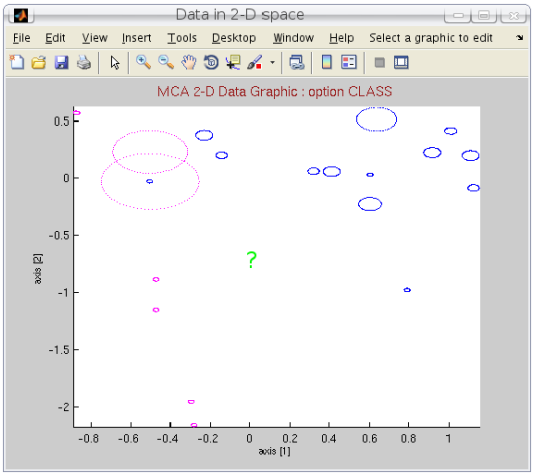


FIG. 13 – Echantillon d'apprentissage et nouvel individu à classer.

5, 2). Représenté dans le premier plan de l'ACM (cf. Fig. 13), ce nouvel individu paraît difficile à classer visuellement. Mais en appliquant la règle de classement, on obtient des probabilités d'appartenance respectivement de 0.06 et 0.94, ne laissant guère de doute à l'affectation de cet oiseau dans la deuxième sous-espèce.

5.2.4 Modèles parcimonieux

Comme dans le cas de données quantitatives, MIXMOD dispose pour des données qualitatives de modèles parcimonieux permettant de proposer, sous le contrôle de critères, des modèles plus simples (avec moins de paramètres à estimer) en imposant des contraintes raisonnables sur les paramètres p_k et α_k.
Ceci passe par une reparamétrisation du paramètre α_k (voir [7]) en faisant apparaître un *centre* et une *dispersion* :

$$\forall k, j : (\alpha_k^{j1}, \ldots, \alpha_k^{jm_j}) \longrightarrow (a_k^{j1}, \ldots, a_k^{jm_j}, \varepsilon_k^{j1}, \ldots, \varepsilon_k^{jm_j})$$

– centre : $a_k^{jh} = \begin{cases} 1 & \text{si } h = \arg\max_h \alpha_k^{jh} \\ 0 & \text{sinon} \end{cases}$

– dispersion : $\varepsilon_k^{jh} = \begin{cases} 1 - \alpha_k^{jh} & \text{si } a_k^{jh} = 1 \\ \alpha_k^{jh} & \text{si } a_k^{jh} = 0. \end{cases}$

Ainsi, comme pour les modèles gaussiens, on peut définir 5 (ou 10 si l'on considère que les proportions peuvent être libres ou égales) modèles parcimonieux en imposant des contraintes sur le terme de dispersion ε_k^{jh} (cf. Tab. 2).

Il est alors intéressant de noter le nombre de paramètres libres (ou degré de liberté (dl)) de chaque modèle (cf. Tab. 3) :
– en fonction de K (le nombre de classes), d (le nombre de variables décrivant chaque individu) et de m_j (nombre de niveaux de réponse de chaque variable),
– pour l'exemple traité($k = 2$, $d = 6$ et $m_j = (2, 5, 6, 3, 5, 4)$).

On voit alors clairement l'intérêt de proposer de tels modèles permettant (quand cela est possible) de choisir un modèle beaucoup plus simple, en particulier lorsque l'on dispose

modèle	la dispersion peut dépendre de ...	dispersions identiques pour ...
$[\varepsilon]$	rien	toutes les classes, variables et tous les niv. de réponse
$[\varepsilon_k]$	classes	toutes les variables et tous les niveaux de réponse
$[\varepsilon^j]$	variables	toutes les classes et tous les niveaux de réponse
$[\varepsilon_k^j]$	classes et variables	tous les niveaux de réponse
$[\varepsilon_k^{jh}]$	classes, variables et niveaux de réponse	dispersions totalement libres

TAB. 2 – Les 5 modèles multinomiaux et leurs contraintes.

modèle	dl	Exemple (oiseaux)
$[p\varepsilon]$	1	1
$[p\varepsilon_k]$	K	2
$[p\varepsilon^j]$	d	6
$[p\varepsilon_k^j]$	Kd	12
$[p\varepsilon_k^{jh}]$	$K\sum_{j=1}^{d}(m_j-1)$	38

TAB. 3 – Les 5 modèles multinomiaux et leurs contraintes (ajouter $K-1$ pour les modèles à proportions libres).

de peu d'individus par rapport au nombre de paramètres à estimer.

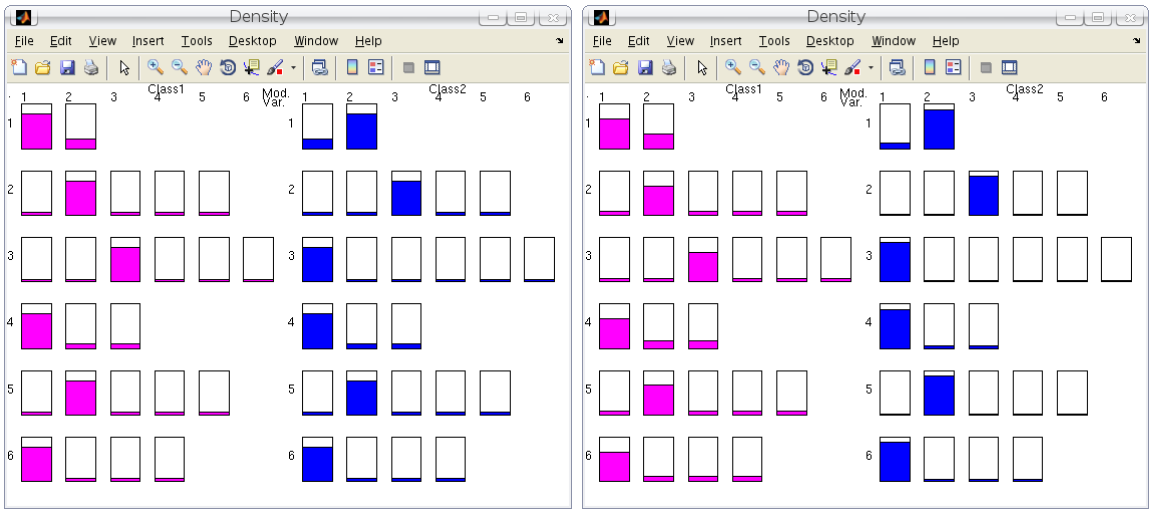
5.2.5 Choix de modèles

On peut appliquer alors la fonction M sur l'échantillon d'apprentissage avec chacun des 10 modèles décrits ci-dessus. Cette fonction donne, par exemple, les paramètres décrits dans la figure 14 pour les modèles $[p\varepsilon]$ (a) et $[p\varepsilon_k]$ (b).

Ayant ainsi obtenu une estimation des paramètres pour les 10 modèles considérés, il reste à choisir le *bon* modèle (voir [4]). Ceci peut être réalisé grâce aux critères disponibles dans MIXMOD.
A titre d'illustration les critères CV et BIC ont été utilisés et les résultats sont représentés dans le tableau 4. Ainsi, le critère CV sélectionnera le modèle $[p\varepsilon_k^{jh}]$, alors que le critère BIC sélectionnera le modèle $[p\varepsilon_k^j]$.

	$[p\varepsilon]$	$[p\varepsilon_k]$	$[p\varepsilon^j]$	$[p\varepsilon_k^j]$	$[p\varepsilon_k^{jh}]$	$[p_k\varepsilon]$	$[p_k\varepsilon_k]$	$[p_k\varepsilon^j]$	$[p_k\varepsilon_k^j]$	$[p_k\varepsilon_k^{jh}]$
LL	-344	-329	-311	-281	-239	-344	-329	-311	-281	-239
CV	87.0%	88.5%	84.1%	92.8%	97.0%	87.0%	85.5%	85.5%	92.8%	97.0%
BIC	693	667	649	613	640	697	671	653	617	645

TAB. 4 – Valeurs de la LogVraisemblance, des critères BIC (à minimiser) et CV(taux de bon reclassement à maximiser) pour les 10 modèles.



(a) (b)

FIG. 14 – Paramètres estimés pour les modèles $[p\varepsilon]$ (a) et $[p\varepsilon_k]$ (b).

individu	hauteur	poids	sexe
1	172.5	66.3	1
2	167.1	54.1	2
⋮	⋮	⋮	⋮

TAB. 5 – Données mixtes.

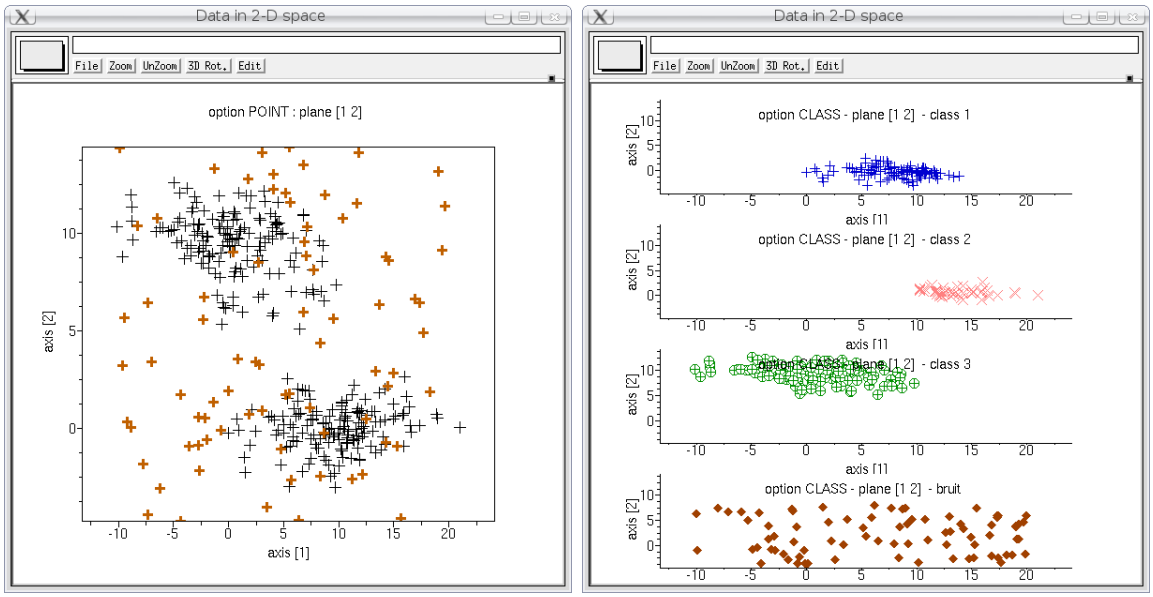
6 Perspectives

MIXMOD est, depuis 2001, un logiciel en pleine évolution. Il intègre régulièrement d'une part de nouvelles fonctionnalités incluant les derniers résultats de la recherche et répondant aux attentes des utilisateurs, et d'autre part, des évolutions informatiques améliorant les performances et rendant le logiciel plus convivial. Les prochaines évolutions envisagées concerneront aussi de nouvelles fonctionnalités et des évolutions informatiques.

6.1 De Nouvelles fonctionnalités

Parmi les prochaines fonctionnalités que MIXMOD proposera, on peut citer :

- Intégration des modèles spécifiques à la grande dimension (modèles *HD*) dans le cadre de la classification non supervisée (voir [5]) ;
- Traitement des *données mixtes* (quantitatives et qualitatives) avec des modèles spécifiques.
  De telles données sont assez courantes, comme dans l'exemple illustré dans le tableau 5 (individus caractérisés par la hauteur, le poids et le sexe) ;
- Traitement des données *bruitées* en intégrant une classe (de bruit) supplémentaire destinée à accueillir tout individu considéré comme étant du bruit.
  Ainsi, sur un jeu de données entaché de bruit comme représenté dans la figure 15 (a), il sera possible de réaliser une classification non supervisée faisant apparaître les trois classes réelles et la classe de bruit (cf. Fig. 15 (b)).



(a) (b)

FIG. 15 – Classification non supervisée sur un jeu de données bruitées (a) faisant apparaître une classe de bruit (graphique du bas de (b)).

6.2 Des évolutions informatiques

De nombreuses améliorations et évolutions informatiques sont programmées pour les prochains mois :

- Amélioration des performances.
  Ce travail de fond permet à MIXMOD d'être un logiciel capable de traiter de très gros jeux de données dans des temps raisonnables. Grâce à des outils de *profiling* (cachegrind, valgrind), une étude précise est réalisée afin de connaître les fonctions et les portions de code les plus gourmandes en temps de calcul et ainsi d'optimiser celles-ci le cas échéant.
- Evolutions dans l'utilisation de MIXMOD :
  - Simplifier l'utilisation des fonctions dédiées à MIXMOD pour Scilab et Matlab.
  - Proposer une fonction MIXMOD pour le logiciel R.
  - Créer des communautés MIXMOD/SCILAB et MIXMOD/MATLAB, MIXMOD/R pour compléter l'éventail d'outils autour de MIXMOD.
  - Développer une interface graphique MIXMOD indépendante de tout autre logiciel.

6.2.1 Une nouvelle interface graphique

Ce développement permettra aux utilisateurs d'avoir accès à MIXMOD sans recourir à l'utilisation de logiciel tiers (Scilab, Matlab ou R) et répondra ainsi à une véritable attente de nombreux utilisateurs (et utilisateurs potentiels) pour lesquels l'obligation d'utiliser des tels logiciels est un frein. Ceci n'impliquera pas l'abandon des fonctions disponibles pour ces logiciels car elles représentent un grand intérêt pour d'autres utilisateurs.
Le travail de conception et de développement a déjà commencé et, on attend une première

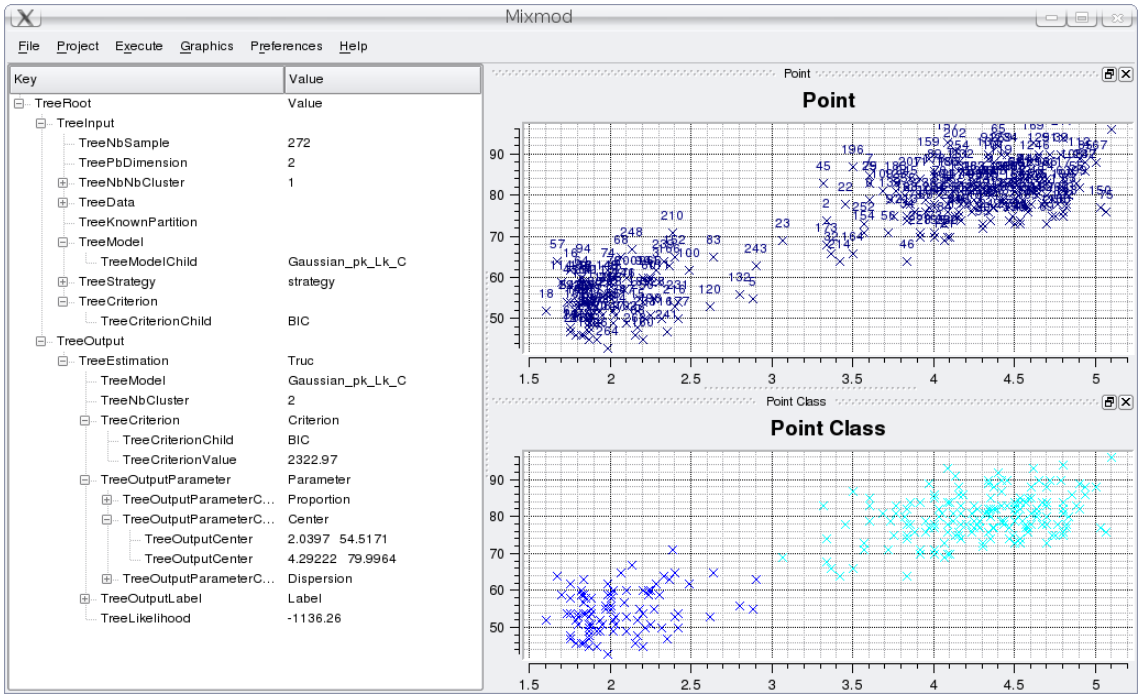


FIG. 16 – Prochaine interface graphique MIXMOD.

version de cette interface MIXMOD indépendante de tout autre logiciel pour le début de l'année 2010. Des captures d'écran réalisées sur le prototype déjà développé (cf. Fig. 16) donne les contours de ce que sera probablement cette interface.

Références

[1] Biernacki, C., Celeux, G., Govaert, G., 2003. Choosing starting values for the EM algorithm for getting the highest likelihood in multivariate gaussian mixture models. Computational Statistics and Data Analysis 41, 561–575.

[2] Biernacki, C., Celeux, G., Echenim, A., Govaert, G., Langrognet, F., 2007. Le logiciel MIXMOD d'analyse de mélange pour la classification et l'analyse discriminante. La Revue de Modulad 35, 25–44.

[3] Biernacki, C., Celeux, A., Govaert, G., Langrognet, F., 2006. Model-Based Cluster and Discriminant Analysis with the MIXMOD Software. Computational Statistics and Data Analysis, vol. 51/2, 587–600.

[4] Biernacki, C., Govaert, G., 1999. Choosing models in model-based clustering and discriminant analysis. J. Statis. Comput. Simul. 64, 49–71.

[5] Bouveyron C., Girard S. and Schmid C., 2007. High Dimensional Discriminant Analysis. Communications in Statistics : Theory and Methods, vol. 36, 2607–2623.

[6] Celeux, G., Diebolt, J., 1985. The SEM algorithm : A probabilistic teacher algorithm derived from the EM algorithm for the mixture problem. Computational Statistics Quarterly 2, 73–82.

[7] Celeux, G., Govaert, G., 1991. Clustering Criteria for Discrete Data and Latent Class Models. Journal of Classification, vol. 8, 157–176.

[8] Celeux, G., Govaert, G., 1992. A classification EM algorithm for clustering and two stochastic versions. Computational Statistics and Data Analysis 14 (3), 315–332.

[9] Celeux, G., Govaert, G., 1995. Gaussian parsimonious clustering models. Pattern Recognition 28 (5), 781–793.

[10] McLachlan, G. J., Krishnan, K., 1997. The EM Algorithm and Extensions. Wiley, New York.

[11] McLachlan, G. J., Peel, D., 2000. Finite Mixture Models. Wiley, New York.