

Etude de la robustesse de RMixmod (package de classification par modèles de mélanges) en cas de chevauchement de classes



F. Langrognat



Rmixmod

1 Le composant RMixmod du projet MIXMOD

- Projet Mixmod
- Principales fonctionnalités
- RMixmod

2 Etude de la robustesse de RMixmod

- Problématique
- Capacité de RMixmod à trouver les bons paramètres et la bonne classification
- Capacité de RMixmod à trouver le bon nombre de classes (et la classification)

3 Conclusion

Etude de la robustesse de RMixmod

1 Le composant RMixmod du projet MIXMOD

- Projet Mixmod
- Principales fonctionnalités
- RMixmod

2 Etude de la robustesse de RMixmod

- Problématique
- Capacité de RMixmod à trouver les bons paramètres et la bonne classification
- Capacité de RMixmod à trouver le bon nombre de classes (et la classification)

3 Conclusion

Etude de la robustesse de RMixmod

1 Le composant RMixmod du projet MIXMOD

- **Projet Mixmod**
- Principales fonctionnalités
- RMixmod

2 Etude de la robustesse de RMixmod

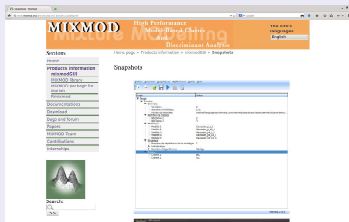
- Problématique
- Capacité de RMixmod à trouver les bons paramètres et la bonne classification
- Capacité de RMixmod à trouver le bon nombre de classes (et la classification)

3 Conclusion

Fiche d'identité

Objectif : diffuser auprès d'un large public un ensemble logiciel de classification par modèles de mélanges

- Projet débuté en 2001, soutenu par 5 organismes
- 4 composants logiciels
Licence GNU GPL - www.mixmod.org
 - ▶ **mixmodLib** (C++) : bibliothèque de calcul
 - ▶ **mixmodGUI** : interface graphique (QT)
 - ▶ **mixmodForMatlab**
 - ▶ **RMixmod**



Etude de la robustesse de RMixmod

1 Le composant RMixmod du projet MIXMOD

- Projet Mixmod
- **Principales fonctionnalités**
- RMixmod

2 Etude de la robustesse de RMixmod

- Problématique
- Capacité de RMixmod à trouver les bons paramètres et la bonne classification
- Capacité de RMixmod à trouver le bon nombre de classes (et la classification)

3 Conclusion

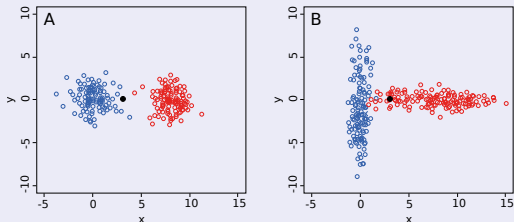
Fonctionnalités (1)

Problématiques traitées

- Classification non supervisée
- Classification supervisée

Modèles de mélange

- Outils souples pour **modéliser** un large spectre de situations
- Calcul des paramètres du modèle sous-jacent - **Caractérisation des classes** (proportion, moyenne, dispersion)
- Classification des individus avec des **métriques** adaptées



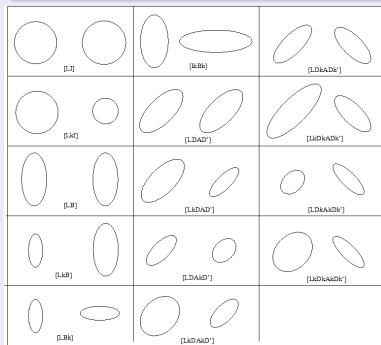
Fonctionnalités (2)

Modèles et métriques

Données quantitatives

14 modèles gaussiens

basés sur la décomposition en valeur sigulière de la matrice de variance



Données quantitatives en grande dimension

8 modèles spécifiques pour la grande dimension

Données qualitatives

5 modèles multinomiaux

basés sur une reparamétrisation de la distribution Multinomiale

Données mixtes

20 modèles hétérogènes

pour les données quantitatives/qualitatives

Fonctionnalités (3)

Algorithmes

Maximisation de la vraisemblance (ou vraisemblance complétée)

- **EM** (Expectation Maximisation)
- **SEM** (Stochastic EM)
- **CEM** (Classification EM)

Critères

- **BIC** (Bayesian Information Criterion)
- **ICL** (Integrated Completed Likelihood)
- **NEC** (Normalized Entropy Criterion)
- **CV** (Cross Validation)

Initialisations et Stratégies

- **6 initialisations**
Ex : 'random', 'short runs of EM',...
- **Algorithmes chaînés**
Ex : 100 iterations de **SEM** puis 50 iterations de **EM**

Et aussi...

- Connaissance partielle des labels des individus (**semi-supervisé**)
- Individus **pondérés**

Etude de la robustesse de RMixmod

1 Le composant RMixmod du projet MIXMOD

- Projet Mixmod
- Principales fonctionnalités
- **RMixmod**

2 Etude de la robustesse de RMixmod

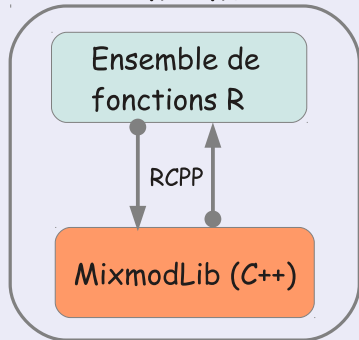
- Problématique
- Capacité de RMixmod à trouver les bons paramètres et la bonne classification
- Capacité de RMixmod à trouver le bon nombre de classes (et la classification)

3 Conclusion

Rmixmod : le package R de Mixmod

Architecture

Rmixmod



Avantages

- **Atouts de R**
 - ▶ Environnement “familier” de l'utilisateur
 - ▶ Interface avec d'autres packages
 - ▶ Outils de visualisation
- **Atouts de mixmodLib (C++)**
 - ▶ Développée depuis 2001
 - ▶ Largement diffusée, et utilisée
 - ▶ **Eprouvée, robuste, rapide**

Classes - Fonctions

Classes Rmixmod

Classes (S4)

Mixmod
MixmodCluster [`<-Mixmod`]
MixmodLearn [`<-Mixmod`]
MixmodPredict
MixmodResults
MixmodDAResults [`<-MixmodResults`]
Model
MultinomialModel [`<-Model`]
GaussianModel [`<-Model`]
Parameter
GaussianParameter [`<-Parameter`]
MultinomialParameter [`<-Parameter`]
Strategy

Fonctions Rmixmod

Fonctions

mixmodCluster
mixmodLearn
mixmodPredict

mixmodStrategy
mixmodGaussianModel
mixmodMultinomialModel
sortByCriterion
nbFactorFromData

summary
print
hist
histCluster
plot
PlotCluster
barplot
barplotCluster

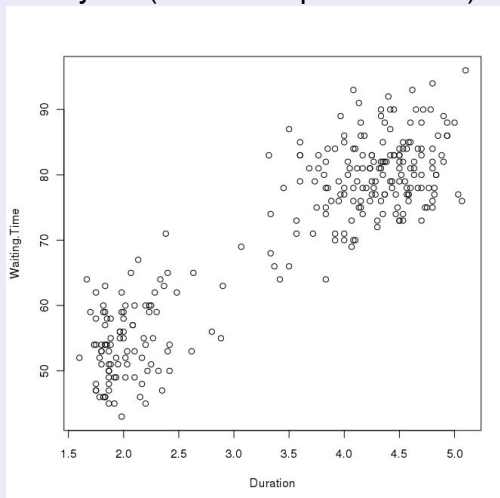
Illustration

Classification non supervisée

Classification non supervisée

Illustration

Geyser (données quantitatives)

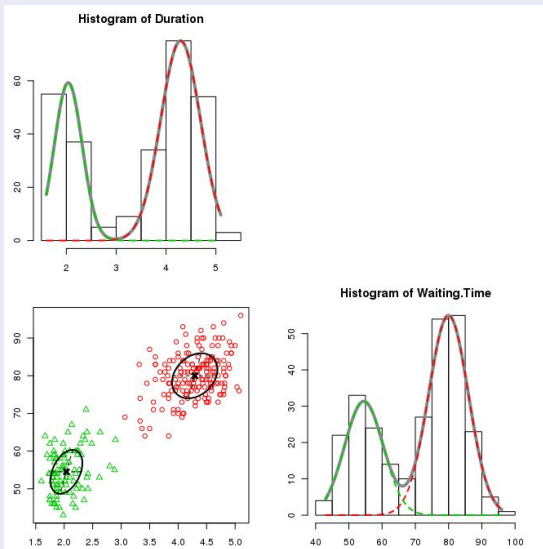


Commandes Rmixmod

```

> data(geyser)
> out<-mixmodCluster(geyser, nbCluster=2)
> summary(out)
*****
* Number of samples      = 272
* Problem dimension      = 2
*****
*      Number of cluster = 2
*      Model Type = Gaussian_pk_Lk_C
*      Criterion = BIC(2322.9719)
*      Parameters = list by cluster
*      Cluster 1 :
*          Proportion = 0.6429
*          Means = 4.2922 79.9964
*          Variances = | 0.1453 0.8301 |
*                   | 0.8301 40.9022 |
*      Cluster 2 :
*          Proportion = 0.3571
*          Means = 2.0397 54.5171
*          Variances = | 0.0984 0.5618 |
*                   | 0.5618 27.6831 |
*      Log-likelihood = -1136.2599
*****

```



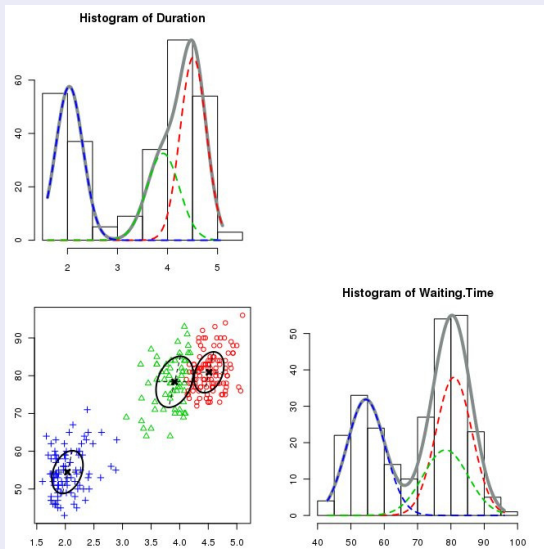
Plusieurs modèles et plusieurs critères

Commandes Rmixmod

```
> out<-mixmodCluster(geyser, nbCluster=2:3, criterion=c("BIC","ICL"),
+ models=mixmodGaussianModel())
> summary(out)
*****
* Number of samples      = 272
* Problem dimension      = 2
*****
*      Number of cluster = 3
*      Model Type = Gaussian_p_L_C
*      Criterion = BIC(2312.6006) ICL(2377.6923)
*      Parameters = list by cluster
*      Cluster 1 :
*          Proportion = 0.3333
*          Means = 3.9765 78.7195
*          Variances = | 0.0798 0.5341 |
*                   | 0.5341 34.2108 |
*      Cluster 2 :
*          Proportion = 0.3333
*          Means = 4.5545 81.0528
*          Variances = | 0.0798 0.5341 |
*                   | 0.5341 34.2108 |
*      Cluster 3 :
*          Proportion = 0.3333
*          Means = 2.0390 54.5083
*          Variances = | 0.0798 0.5341 |
*                   | 0.5341 34.2108 |
*      Log-likelihood = -1131.0742
*****
```

Illustration

plot (3 classes)



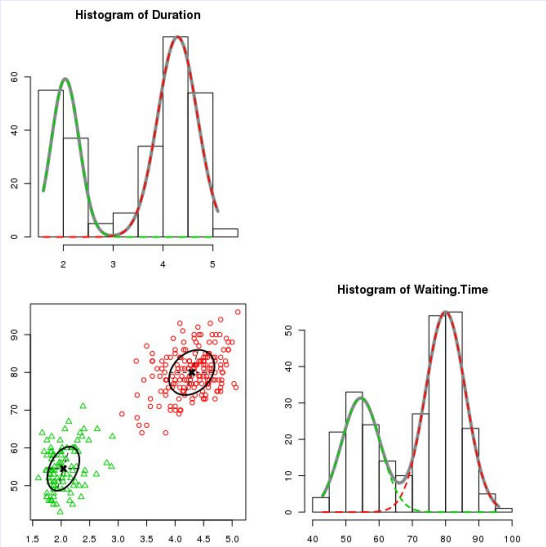
Tri selon ICL

```
> out_icl<-sortByCriterion(out, "ICL")
> summary(out_icl)
*****
* Number of samples      = 272
* Problem dimension      = 2
*****
*
*   Number of cluster = 2
*   Model Type = Gaussian_pk_Lk_D_Ak_D
*   Criterion = BIC(2320.2833) ICL(2320.5794)
*   Parameters = list by cluster
*   Cluster 1 :
*       Proportion = 0.6432
*       Means = 4.2915 79.9893
*       Variances = | 0.1588 0.6810 |
*                   | 0.6810 35.7667 |
*
*   Cluster 2 :
*       Proportion = 0.3568
*       Means = 2.0387 54.5041
*       Variances = | 0.0783 0.6467 |
*                   | 0.6467 33.8930 |
*
*   Log-likelihood = -1132.1126
*****
```

Illustration

plot (2 classes)

2 classes



Etude de la robustesse de RMixmod

1 Le composant RMixmod du projet MIXMOD

- Projet Mixmod
- Principales fonctionnalités
- RMixmod

2 Etude de la robustesse de RMixmod

- Problématique
- Capacité de RMixmod à trouver les bons paramètres et la bonne classification
- Capacité de RMixmod à trouver le bon nombre de classes (et la classification)

3 Conclusion

Etude de la robustesse de RMixmod

1 Le composant RMixmod du projet MIXMOD

- Projet Mixmod
- Principales fonctionnalités
- RMixmod

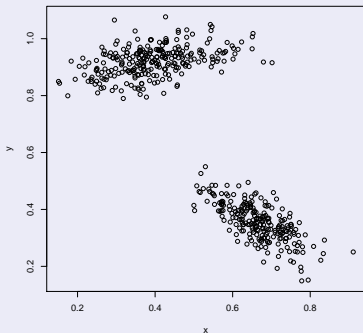
2 Etude de la robustesse de RMixmod

- **Problématique**
- Capacité de RMixmod à trouver les bons paramètres et la bonne classification
- Capacité de RMixmod à trouver le bon nombre de classes (et la classification)

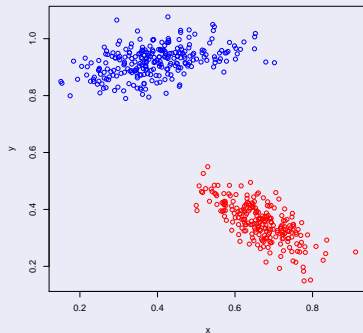
3 Conclusion

Jeu de données sans chevauchement

Jeu de données



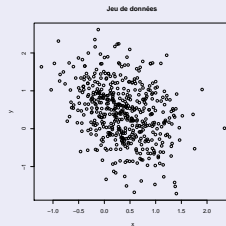
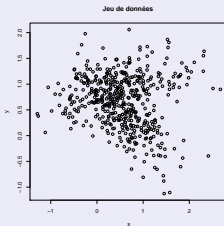
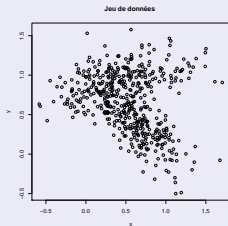
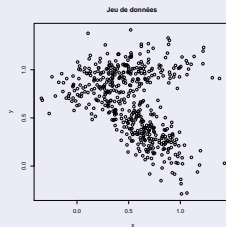
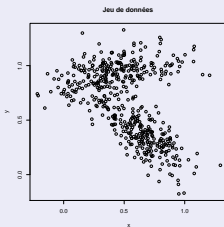
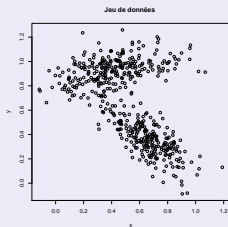
Classification



La classification ne pose pas de problème

Problématique

Et avec plus de chevauchement ?



Chevauchement par paire de classes

Définition (Maitra et Melnykov en 2010)

Le **chevauchement** $\omega_{k_1 k_2}$ entre les classes k_1 et k_2 est défini par

$$\omega_{k_1 k_2} = \omega_{k_1 | k_2} + \omega_{k_2 | k_1}$$

où $\omega_{k_1 | k_2}$ est la probabilité que la variable aléatoire X appartenant à la classe k_2 soit classée **par erreur** dans la classe k_1 .

Utilisation dans le cadre des modèles de mélange

On peut simuler des données

- distribuées selon un mélange gaussien
- et respectant un chevauchement (choisi) entre les différentes classes

Etude de la robustesse de RMixmod

1 Le composant RMixmod du projet MIXMOD

- Projet Mixmod
- Principales fonctionnalités
- RMixmod

2 Etude de la robustesse de RMixmod

- Problématique
- Capacité de RMixmod à trouver les bons paramètres et la bonne classification
- Capacité de RMixmod à trouver le bon nombre de classes (et la classification)

3 Conclusion

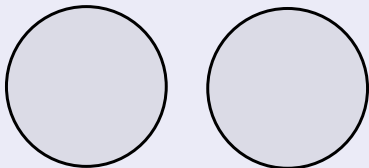
Pour 2 classes

- On **fixe $\bar{\omega}$** (chevauchement moyen) entre les 2 classes (donc $\omega_1 = 2$)
- On **calcule les paramètres** d'un mélange gaussien dont le degré de chevauchement est $\bar{\omega}$ (fonction `MixSim` du package `MixSim`)
- On **simule des données** suivant ces paramètres (fonction `simdataset` du package `MixSim`)
- Sur ce jeu de données (sans connaître ni les labels, ni les paramètres du modèle gaussien), **on cherche une classification** avec la fonction `mixmodCluster` du package `RMixmod`
- On **compare les résultats** obtenus avec `RMixmod` avec les vraies valeurs (labels et paramètres)

1^{re} situation

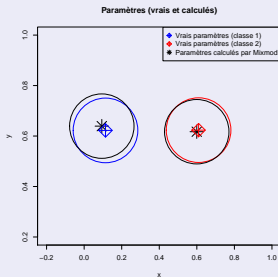
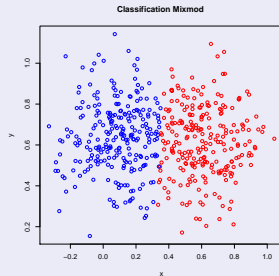
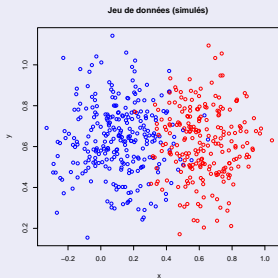
Modèle sphérique et même proportion [Gaussian_p_I]

Les matrices de variance sont identiques et sphériques



```
parameters <- MixSim(BarOmega=0.15, K=2, p=2, sph=TRUE,hom=TRUE)
data <- simdataset(n=500, Pi=parameters$Pi, Mu=parameters$Mu, S=parameters$S)
model <- mixmodGaussianModel(listModels="Gaussian_p_L_I")
res_mixmod <- mixmodCluster(data.frame(data$X), model=model, nbCluster=2)
```

$$\bar{\omega} = 0.15$$



Pourcentage d'individus
mal classés = 7,4%

Probabilité d'appartenance

La class. par modèles de mélanges donne une information complémentaire :

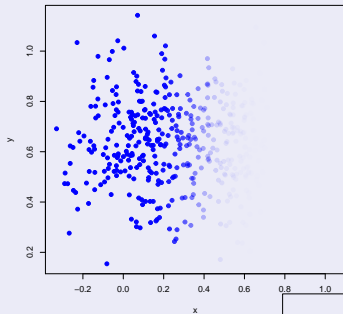
la probabilité d'appartenance de l'individu i à la classe k : p_{ik}

```
> probas <- res_mixmod@bestResult@proba
>
> probas
      [,1]      [,2]
[1,] 8.398762e-01 1.601238e-01
[2,] 7.551083e-01 2.448917e-01
[3,] 9.999366e-01 6.339925e-05
[4,] 2.692252e-01 7.307748e-01
[5,] 9.990483e-01 9.517056e-04
[6,] 9.999801e-01 1.987949e-05
[7,] 9.992282e-01 7.717543e-04
[8,] 9.992276e-01 7.724087e-04
[9,] 9.902374e-01 9.762624e-03
[10,] 3.779903e-01 6.220097e-01
[11,] 9.987837e-01 1.216312e-03
[12,] 9.936269e-01 6.373078e-03
[13,] 9.556769e-01 4.432311e-02
[14,] 9.679888e-01 3.201120e-02
[15,] 9.988387e-01 1.161303e-03
[16,] 6.010903e-02 9.398910e-01
```

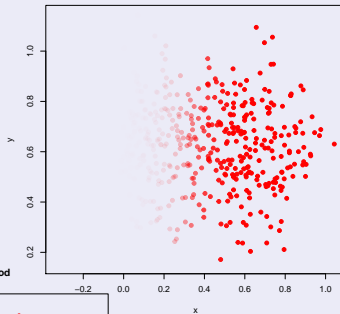
Probabilité d'appartenance

$$\bar{\omega} = 0.15$$

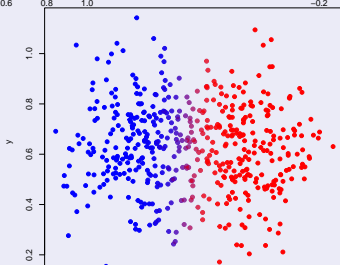
Classification Mixmod - classe bleue



Classification Mixmod - classe rouge



Classification Mixmod



Individus mal classés

Taux d'individus mal classés

Obtenu en comparant les labels calculés (\hat{z}_i) aux vrais labels (z_i)

Rappel : $\hat{z}_i = \arg \max_k p_{ik}$

```
> labels <- res_mixmod@bestResult@partition
>
> labels
[1] 1 1 1 2 1 1 1 1 1 2 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 2 1
[38] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 2 1
[75] 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
[112] 2 1 1 1 2 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
[149] 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 2 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
[186] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2
[223] 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2
[260] 1 1 1 1 1 1 1 1 1 1 2 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
[297] 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
[334] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
[371] 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
[408] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
[445] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
[482] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
>
```

Pour $\bar{\omega} = 0.15$, on obtient 37 individus mal classés (sur 500) -> 7,4%
9 individus classés à tort dans la classe 1, et 28 à tort dans la classe 2

Note : $\bar{\omega} = \omega_{1|2} + \omega_{2|1}$

$\bar{\omega}$ correspond bien **au double** du taux d'individus mal classés

Chevauchement

Estimation du degré de chevauchement a posteriori

En utilisant les probabilités d'appartenance

- On peut calculer (a posteriori) $\hat{\omega}_{k_1|k_2}$

$$\hat{\omega}_{k_1|k_2} = \frac{\sum_{i \text{ tq } z_i=k_2} p_{ik_1}}{n_{k_2}} \text{ où } n_{k_2} \text{ est l'effectif de la classe } k_2$$

- On peut donc calculer (a posteriori) $\hat{\omega}$

$$\hat{\omega} = \hat{\omega}_{k_1|k_2} + \hat{\omega}_{k_2|k_1}$$

Pour $\bar{\omega} = 0.15$, on obtient :

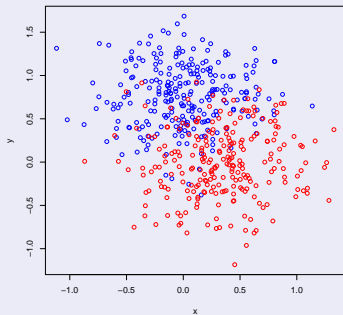
- $\hat{\omega}_{1|2} = 0.1$ et $\hat{\omega}_{2|1} = 0.09$
- $\hat{\omega} = 0.19$

```
> probas <- res_mixmod@bestResult@proba
>
> probas
      [,1]      [,2]
[1,] 8.398762e-01 1.601238e-01
[2,] 7.551083e-01 2.448917e-01
[3,] 9.999366e-01 6.339925e-05
[4,] 2.692252e-01 7.307748e-01
[5,] 9.990483e-01 9.517056e-04
[6,] 9.999801e-01 1.987949e-05
[7,] 9.992282e-01 7.717543e-04
[8,] 9.992276e-01 7.724087e-04
[9,] 9.902374e-01 9.762624e-03
[10,] 3.779903e-01 6.220897e-01
[11,] 9.987837e-01 1.216312e-03
[12,] 9.936269e-01 6.373078e-03
[13,] 9.556769e-01 4.432311e-02
[14,] 9.679880e-01 3.201120e-02
[15,] 9.988387e-01 1.161303e-03
[16,] 6.010903e-02 9.398910e-01
[17,] 9.293650e-01 7.063498e-02
[18,] 9.515216e-01 4.847841e-02
[19,] 9.999441e-01 5.593217e-05
[20,] 9.924275e-01 7.572538e-03
[21,] 9.994153e-01 5.846561e-04
[22,] 9.919695e-01 8.030512e-03
[23,] 9.949279e-01 5.072075e-03
[24,] 9.960419e-01 3.958129e-03
[25,] 6.364523e-01 3.635477e-01
[26,] 7.884666e-01 2.115334e-01
[27,] 9.899079e-01 1.009209e-02
[28,] 9.500445e-01 4.995546e-02
[29,] 7.889285e-01 2.110715e-01
[30,] 9.638811e-01 3.611885e-02
[31,] 9.985369e-01 1.463103e-03
```

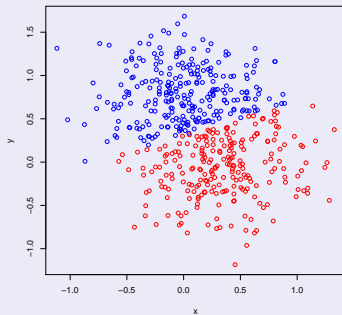
Avec d'autres valeurs de chevauchement

$$\bar{\omega} = 0.3$$

Jeu de données (simulés)



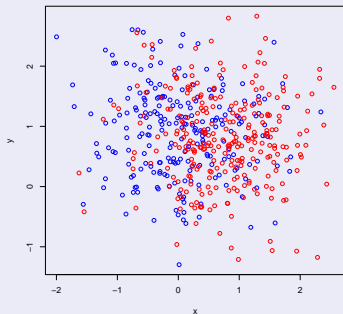
Classification Mixmod



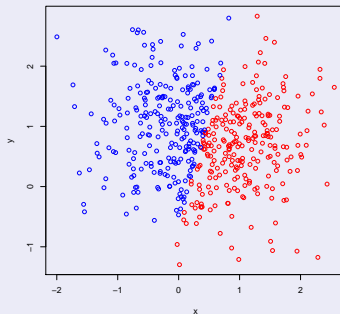
Pourcentage d'individus mal classés = 17,2%

$$\bar{\omega} = 0.65$$

Jeu de données (simulés)



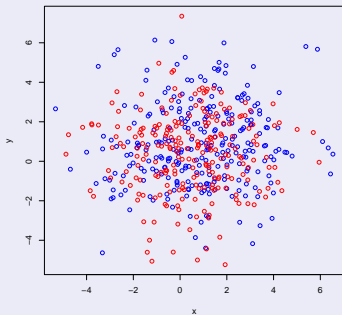
Classification Mixmod



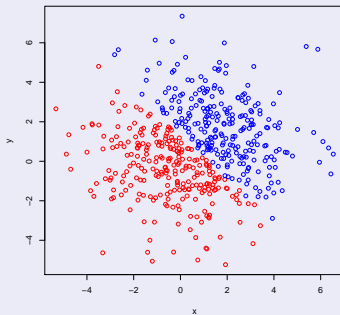
Pourcentage d'individus mal classés = 31,8%

$$\bar{\omega} = 0.9$$

Jeu de données (simulés)



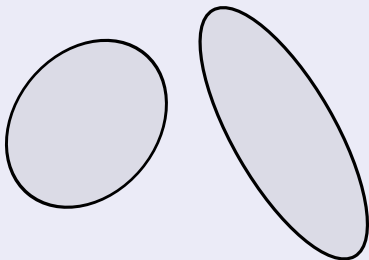
Classification Mixmod



Pourcentage d'individus mal classés = 45,6%

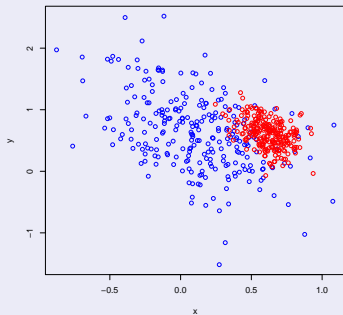
2^e situation

Modèle général et proportions libres [*Gaussian*_ p_k _ L_k _ C_k]

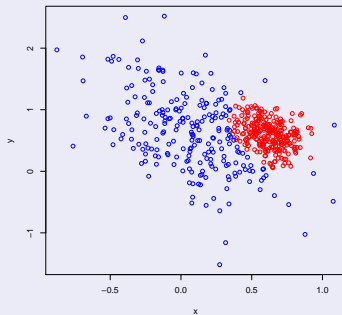


$$\bar{\omega} = 0.15$$

Jeu de données (simulés)



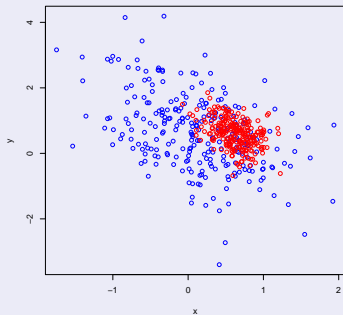
Classification Mixmod



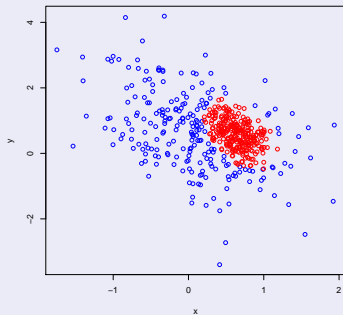
Pourcentage d'individus mal classés = 8,2%

$$\bar{\omega} = 0.3$$

Jeu de données (simulés)



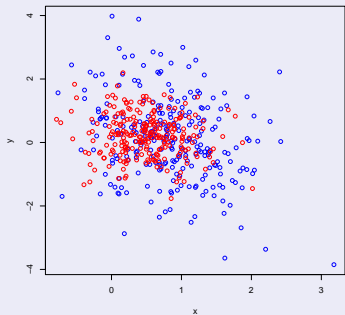
Classification Mixmod



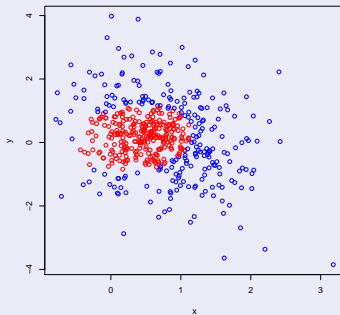
Pourcentage d'individus mal classés = 15%

$$\bar{\omega} = 0.6$$

Jeu de données (simulés)



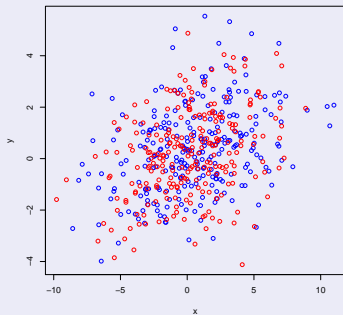
Classification Mixmod



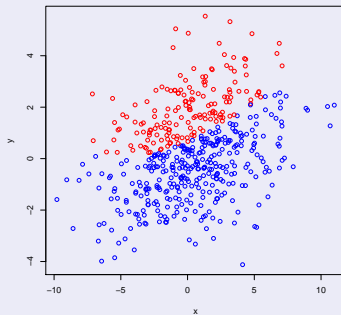
Pourcentage d'individus mal classés = 31,4%

$$\bar{\omega} = 0.9$$

Jeu de données (simulés)



Classification Mixmod



Pourcentage d'individus mal classés = 47,2%

Etude de la robustesse de RMixmod

1 Le composant RMixmod du projet MIXMOD

- Projet Mixmod
- Principales fonctionnalités
- RMixmod

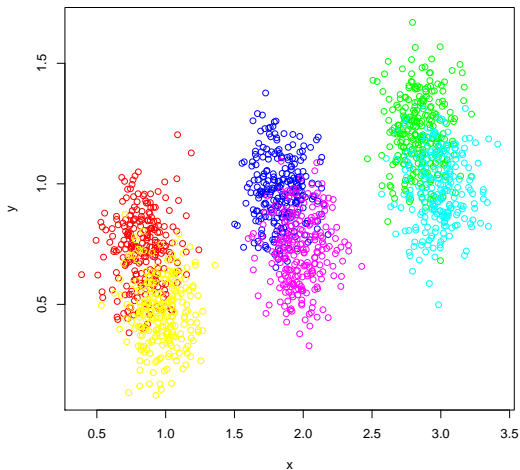
2 Etude de la robustesse de RMixmod

- Problématique
- Capacité de RMixmod à trouver les bons paramètres et la bonne classification
- Capacité de RMixmod à trouver le bon nombre de classes (et la classification)

3 Conclusion

Nombre de classes ?

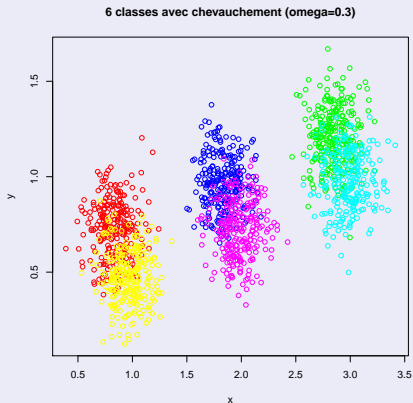
6 classes avec chevauchement ($\omega=0.3$)



Nombre de classes ?

Combien de classes ? 3 ou 6

- En théorie : **6 classes**
- Mais l'utilisateur peut souhaiter une solution avec des classes bien séparées : **3 classes**



Que peut faire RMixmod dans cette situation ?

Nombre de classes

RMixmod : de 2 à 6 classes

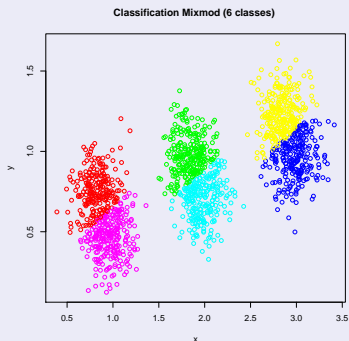
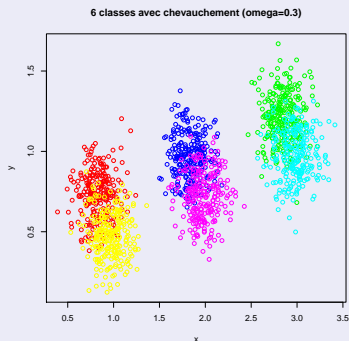
La meilleure solution proposée par RMixmod est avec 6 classes

```
> res_mixmod <- mixmodCluster(data6, model=mod, nbCluster=2:6)
> summary(res_mixmod)
*****
* Number of samples      = 1500
* Problem dimension      = 2
*****
*       Number of cluster = 6
*       Model Type       = Gaussian_p_L_I
*       Criterion        = BIC(1206.5190)
*       Parameters       = list by cluster
*       Cluster 1 :
*           Proportion = 0.1667
*           Means      = 0.9621 0.4730
*           Variances  = | 0.0201 0.0000 |
*                       | 0.0000 0.0201 |
*       Cluster 2 :
*           Proportion = 0.1667
*           Means      = 2.8463 1.2216
*           Variances  = | 0.0201 0.0000 |
*                       | 0.0000 0.0201 |
*       Cluster 3 :
*           Proportion = 0.1667
*           Means      = 1.9756 0.7203
*           Variances  = | 0.0201 0.0000 |
*                       | 0.0000 0.0201 |
*       Cluster 4 :
*           Proportion = 0.1667
*           Means      = 1.8285 0.8703
```

6 classes

Qualité des résultats obtenus par RMixmod

Résultats avec 6 classes (BIC)



Choix du nombre de classes dans RMimxod

BIC - ICL

$$\underbrace{\text{Critère}}_{\text{à maximiser}} = \underbrace{\text{maximum log vraisemblance}}_{\text{adéquation modèle-données}} - \underbrace{\text{pénalité}}_{\text{"coût" du modèle}}$$

critère	pénalité	interprétation	objectif
BIC	$0.5\nu \ln(n)$	complexité modèle	convergence
ICL	$0.5\nu \ln(n) - \sum_{i,k} \hat{z}_{ik} \ln t_{ik}(\hat{\theta})$	complexité modèle + entropie partition	classes bien séparées

ν : nombre de paramètres libres du modèle

Options dans RMixmod

- Par défaut : **BIC**
- Options : **ICL**, NEC (class. non supervisée) et CV (class. supervisée)
Ex : on peut utiliser **BIC et ICL** et **trier** selon l'un ou l'autre des critères

Nombre de classes

RMixmod : de 2 à 6 classes et tri selon ICL

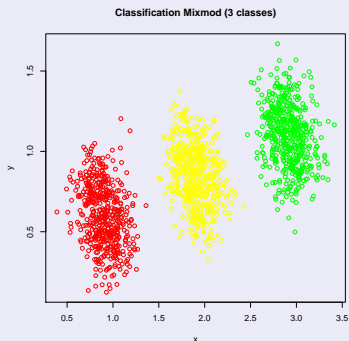
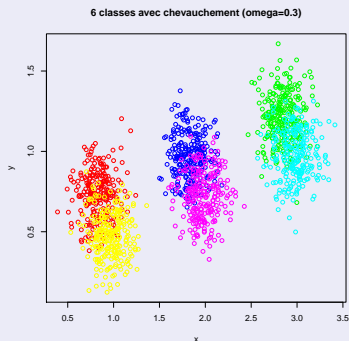
La meilleure solution proposée par RMixmod avec ICL est avec 3 classes

```
> res_mixmod <- mixmodCluster(data6, model=mod, nbCluster=2:6, criterion=c("BIC","ICL"))
>
> res_icl<-sortByCriterion(res_mixmod,"ICL")
>
> summary(res_icl)
*****
* Number of samples   = 1500
* Problem dimension   = 2
*****
*
*   Number of cluster = 3
*   Model Type = Gaussian_p_L_I
*   Criterion = BIC(1375.5983) ICL(1379.2134)
*   Parameters = list by cluster
*
*   Cluster 1 :
*       Proportion = 0.3333
*       Means = 0.8998 0.6007
*       Variances = | 0.0304 0.0000 |
*                   | 0.0000 0.0304 |
*
*   Cluster 2 :
*       Proportion = 0.3333
*       Means = 2.9113 1.0999
*       Variances = | 0.0304 0.0000 |
*                   | 0.0000 0.0304 |
*
*   Cluster 3 :
*       Proportion = 0.3333
*       Means = 1.9053 0.8483
*       Variances = | 0.0304 0.0000 |
*                   | 0.0000 0.0304 |
*
*   Log-likelihood = -662.2029
*****
```

3 classes

Qualité des résultats obtenus par RMixmod

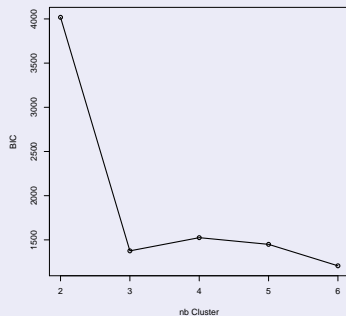
Résultats avec 3 classes (ICL)



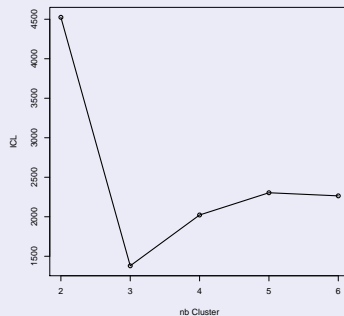
Critères de sélection

Valeurs de BIC et ICL en fonction du nombre de classes

Valeurs de BIC en fonction du nombre de classes



Valeurs de ICL en fonction du nombre de classes



Etude de la robustesse de RMixmod

1 Le composant RMixmod du projet MIXMOD

- Projet Mixmod
- Principales fonctionnalités
- RMixmod

2 Etude de la robustesse de RMixmod

- Problématique
- Capacité de RMixmod à trouver les bons paramètres et la bonne classification
- Capacité de RMixmod à trouver le bon nombre de classes (et la classification)

3 Conclusion

Conclusion

RMimxod et le chevauchement de classes

- RMixmod permet de retrouver la **classification** et les **paramètres** du modèle sous-jacent quelque soit la valeur de chevauchement avec une erreur (**taux d'individus mal classés**) **conforme** à la valeur de chevauchement
- Les **probabilités d'appartenance** aux classes donnent une **information pertinente** pour connaître les individus 'difficilement classables'
- Lors de la recherche du **bon nombre de classes**, les **critères de sélection** disponibles dans RMimxod permettent de tenir compte de l'objectif de l'utilisateur

Etude de la robustesse de RMixmod en cas de chevauchement de classes

FIN

Merci de votre attention

- Site web : <http://www.mixmod.org>
- contact :
 - ▶ contact@mixmod.org
 - ▶ florent.langrognet@univ-fcomte.fr