

Méthodes de classification et de clustering avec les fonctions de croyance

D. Mercier et É. Lefèvre
(Fortement inspiré des talks de T. Denœux)

PFIA 2022

Tutoriel "Fonctions de croyance et applications à l'apprentissage automatique"

Lundi 27 juin 2022

Motivations

- ▶ L'apprentissage automatique a connu un essor important ces dernières années.
- ▶ L'objectif est de faire des prédictions (par exemple : l'appartenance d'un objet à une classe).
- ▶ L'un des problèmes actuels est celui de quantifier les incertitudes liées à ces prédictions :
 - ▶ Sources d'incertitudes nombreuses (incertitudes épistémique ou aléatoire).
 - ▶ Intérêt de la gestion de l'incertitude : l'utilisation de règles de décision, pouvoir rendre la main au décideur lorsque cette incertitude est trop grande, faire de la fusion d'informations,...

Méthodes de classification

Rappel Classification

Evidential K-NN

Pro-DS

Méthodes de clustering

Rappel Clustering

Clustering évidentiel

ECM

EVCLUS

Autres approches

Méthodes de classification

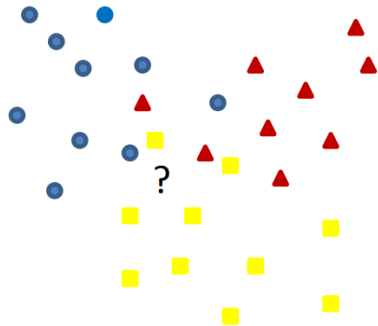
Rappel Classification

Evidential K-NN

Pro-DS

Méthodes de clustering

Rappel classification



- ▶ Les objets sont répartis en c classes avec $\Omega = \{\omega_1, \dots, \omega_c\}$.
- ▶ Chaque objet est décrit par :
 - ▶ un vecteur de caractéristiques \mathbf{X}
 - ▶ une étiquette $y \in \Omega$
- ▶ Problématique : à partir d'un ensemble d'apprentissage $\mathcal{L} = \{(\mathbf{X}_1, y_1), \dots, (\mathbf{X}_n, y_n)\}$ prédire la classe d'un objet \mathbf{X} .

Méthodes de classification

Rappel Classification

Evidential K-NN

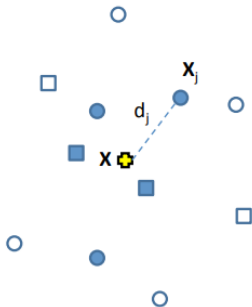
Pro-DS

Méthodes de clustering

Evidential KNN (E-KNN)

Classifieur Evidentiel basé sur les distances

- ▶ Soit $\mathcal{N}_K(\mathbf{X}) \subset \mathcal{L}$ composé des K plus proche voisins d'un objet \mathbf{X} à classer
- ▶ Chaque $\mathbf{X}_j \in \mathcal{N}_K(\mathbf{X})$ peut être considéré comme une part de croyance concernant l'appartenance de \mathbf{X} .
- ▶ Cette part de croyance est décroissante en fonction de la distance d_j entre \mathbf{X} et \mathbf{X}_j .



- ▶ Cadre de discernement : $\Omega = \{\omega_1, \dots, \omega_c\}$
- ▶ La part de croyance de l'objet \mathbf{X}_j , qui a une étiquette y_j , peut être représentée de la manière suivante :

$$m_j(\{\omega_k\}) = \phi_k(d_j)y_{jk}, \quad \forall k = 1, \dots, c$$
$$m_j(\Omega) = 1 - \phi_k(d_j)$$

. où :

- ▶ $y_{jk} = 1$ si $y_j = \omega_k$ et 0 sinon
- ▶ ϕ_k (avec $k = 1, \dots, c$) sont des fonctions décroissantes de $[0, +\infty)$ à $[0, 1]$ telles que $\lim_{d \rightarrow +\infty} \phi_k(d) = 0$

- ▶ Les K voisins fournissent donc K fonctions de croyance qui peuvent être fusionnées avec la règle de Dempster :

$$m = \bigoplus_{\mathbf{x}_j \in \mathcal{N}_K(\mathbf{X})} m_j$$

- ▶ Règle de décision : maximum de plausibilité

Apprentissage des paramètres

- ▶ Les fonctions ϕ_k sont choisies de la forme suivante :

$$\phi_k(d) = \alpha \exp(-\gamma_k d^2)$$

- ▶ Le paramètre α est généralement fixé à 0.95.
- ▶ Les paramètres γ_k peuvent être appris à partir de l'apprentissage en minimisant la fonction suivante :

$$C(\gamma) = \sum_{j=1}^n \sum_{k=1}^c (pl_j(\omega_k) - y_{jk})^2$$

avec pl_j la fonction de contour¹ de la fonction m_j obtenue en utilisant la donnée d'apprentissage \mathbf{X}_j .

¹ Equivalent de la fonction PI réduite aux singletons.

Références et Codes

- ▶ Denœux, T. 1995. “A k-Nearest Neighbor Classification Rule Based on Dempster-Shafer Theory.” IEEE Trans. on Systems, Man and Cybernetics 25 (05): 804–13.
- ▶ Zouhal, L. M., and T. Denœux. 1998. “An Evidence-Theoretic k-NN Rule with Parameter Optimization.” IEEE Trans. on Systems, Man and Cybernetics C 28 (2): 263–71.
- ▶ Package R : evclass
(<https://cran.r-project.org/web/packages/evclass/index.html>)
 - ▶ EkNNinit
 - ▶ EkNNfit
 - ▶ EkNNval

Exemples

```
data(iris)
train<-sample(150,100)
xtrain<-iris[train,1:4]
ytrain<-iris[train,5]
param<-EkNNinit(xtrain,ytrain)
K<-5
fit<-EkNNfit(xtrain,ytrain,K,param)
xtst<-iris[-train,1:4]
ytst<-iris[-train,5]
test<-EkNNval(xtrain,ytrain,xtst,K,ytst,fit$param)
```

Exercice n°1

- ▶ **Enoncé** : Visualiser les valeurs de γ avant et après l'optimisation et l'erreur de classification commise après l'optimisation.

- ▶ **Correction** :

```
data(iris)
train<-sample(150,100)
xtrain<-iris[train,1:4]
ytrain<-iris[train,5]
param<-EkNNinit(xtrain,ytrain)
K<-5
fit<-EkNNfit(xtrain,ytrain,K,param)
xtst<-iris[-train,1:4]
ytst<-iris[-train,5]
test<-EkNNval(xtrain,ytrain,xtst,K,ytst,fit$param)
cat("Valeur de gamma avant optimisation : ", param$gamma,"\n")
cat("Valeur de gamma apres optimisation : ", fit$param$gamma,"\n")
cat("Taux d'erreur : ",test$err,"\n")
```

Exercice n°2

- ▶ **Enoncé :** Reprendre l'exemple précédent sur les données "Bananas" ($n=1000$, $r=5$ et $s=1$). Visualiser le résultat de la classification dans l'espace des caractéristiques.

- ▶ **Correction :**

```
data<-bananas(1000,5,1)
train<-sample(1000,500)
xtrain<-data$x[train,]
ytrain<-data$y[train]
param<-EkNNinit(xtrain,ytrain)
K<-5
fit<-EkNNfit(xtrain,ytrain,K,param)
xtst<-data$x[-train,]
ytst<-data$y[-train]
test<-EkNNval(xtrain,ytrain,xtst,K,ytst,fit$param)
plot(xtst,pch=ytst,col=ytst)
cat("Taux d'erreur : ", test$err,"\n")
cat("Matrice de confusion :")
print(table(ytst,test$ypred))
```

Méthodes de classification

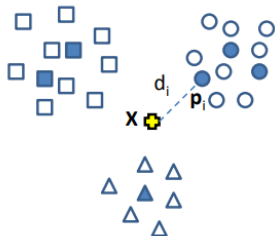
Rappel Classification

Evidential K-NN

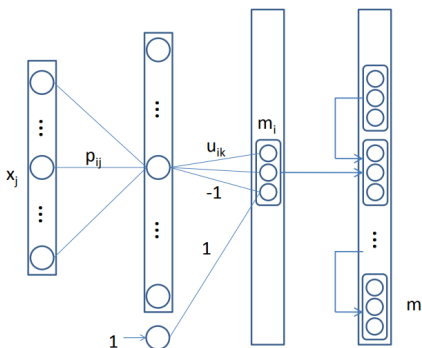
Pro-DS

Méthodes de clustering

- ▶ L'ensemble d'apprentissage est résumé à r prototypes.
- ▶ Chaque prototype \mathbf{p}_i a un degré d'appartenance $\mu_{ik} \in [0, 1]$ à chaque classe ω_k avec $\sum_{k=1}^c \mu_{ik} = 1$.
- ▶ Chaque prototype \mathbf{p}_i est une part de croyance sur la classe de \mathbf{X} qui décroît en fonction de la distance d_i entre \mathbf{X} et le prototype \mathbf{p}_i .



Formalisation



- Information apportée par \mathbf{p}_i :

$$m_i(\{\omega_k\}) = \alpha_i \mu_{ik} \exp(-\gamma_i d_i^2)$$

$$\forall k = 1, \dots, c$$

$$m_i(\Omega) = 1 - \alpha_i \exp(-\gamma_i d_i^2)$$

- Combinaison des r fonctions de croyance :

$$m = \bigoplus_{i=1}^r m_i$$

Apprentissage des paramètres

- ▶ Paramètres du modèle :
 - ▶ Prototypes \mathbf{p}_i , $i = 1, \dots, r$.
 - ▶ Degrés d'appartenance μ_{ik} . $i = 1, \dots, r$, $k = 1, \dots, c$.
 - ▶ α_i et γ_i , $i = 1, \dots, r$.
- ▶ Fonction de coût :

$$J(\theta) = \sum_{i=1}^n \sum_{k=1}^c (Pl_{ik} - y_{ik})^2 + \lambda \sum_{i=1}^r \alpha_i$$

avec Pl_{ik} la plausibilité de ω_k pour l'objet \mathbf{X}_i , $y_{ik} = 1$ si $y_i = \omega_k$ 0 sinon et λ un hyperparamètre.

- ▶ Denœux, T. 2000. “A Neural Network Classifier Based on Dempster-Shafer Theory.” IEEE Trans. on Systems, Man and Cybernetics A 30 (2): 131–50.
- ▶ Package R : evclass
(<https://cran.r-project.org/web/packages/evclass/index.html>)
 - ▶ ProDSinit
 - ▶ ProDSfit
 - ▶ ProDSval

Exemples

```
data(iris)
train<-sample(150,100)
xtrain<-iris[train,1:4]
ytrain<-iris[train,5]
param<-proDSinit(xtrain,ytrain,3,nprotoPerClass=TRUE,crisp=TRUE)
fit<-proDSfit(xtrain,ytrain,param)
xtst<-iris[-train,1:4]
ytst<-iris[-train,5]
test<-proDSval(xtst,fit$param,ytst)
```

Exercice n°3

► Enoncé :

1. Appliquer l'exemple précédent sur les données Bananas ($n=500$).
2. Faire varier le nombre de prototypes (1, 3 et 5)
3. Afficher pour chaque valeur : la position des prototypes, la classification obtenue et les résultats de classification.

Correction Exercice n°3

```
data<-bananas(1000,5,1)
train<-sample(1000,500)
xtrain<-data$x[train,]
ytrain<-data$y[train]
nb_proto<-3 ### à modifier en fonction du nombre de proto
param<-proDSinit(xtrain,ytrain,nb_proto,nprotoPerClass=TRUE,crisp=TRUE)
fit<-proDSfit(xtrain,ytrain,param)
xtst<-data$x[-train,]
ytst<-data$y[-train]
test<-proDSval(xtst,fit$param,ytst)
plot(xtrain,col=ytrain)
plot(xtst,col=ytst)
points(fit$param$W,col="green",pch=5)
cat("Taux d'erreur : ",test$err,"\n")
cat("Matrice de confusion :")
print(table(test$ypred,ytst))
```

Méthodes de classification

Méthodes de clustering

Rappel Clustering

Clustering évidentiel

ECM

EVCLUS

Autres approches

Méthodes de classification

Méthodes de clustering

Rappel Clustering

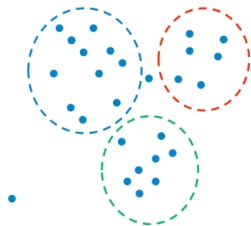
Clustering évidentiel

ECM

EVCLUS

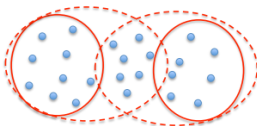
Autres approches

- ▶ Soit n objets décrits par :
 - ▶ des vecteurs attributs $\mathbf{X}_1, \dots, \mathbf{X}_n$ (informations de type attribut) ou
 - ▶ des dissimilarités (informations proximité)
- ▶ Objectif :
 - ▶ Découvrir des groupes parmi les données
 - ▶ Evaluer l'incertitude dans les groupes



Différents types de partition

- ▶ Hard clustering : pas de représentation de l'incertitude. Un objet est affecté à une classe seulement ($\mu_{ik} = 1$ si l'objet \mathbf{X}_i appartient à la classe ω_k , $\mu_{ik} = 0$ sinon).
- ▶ Fuzzy clustering : chaque objet \mathbf{X}_i a un degré d'appartenance $\mu_{ik} \in [0, 1]$ à une classe ω_k avec $\sum_{k=1}^c \mu_{ik} = 1$
- ▶ Rough clustering : l'appartenance d'un objet \mathbf{X}_i à une classe ω_k est représentée par un couple $(\underline{\mu}_{ik}, \bar{\mu}_{ik}) \in \{0, 1\}^2$



Méthodes de classification

Méthodes de clustering

Rappel Clustering

Clustering évidentiel

ECM

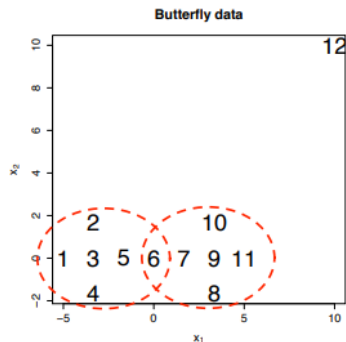
EVCLUS

Autres approches

Clustering évidentiel

- ▶ Soit $\mathcal{X} = \{\mathbf{X}_1, \dots, \mathbf{X}_n\}$ un ensemble de n objets et $\Omega = \{\omega_1, \dots, \omega_c\}$ un ensemble de c classes (clusters).
- ▶ Chaque objet \mathbf{X}_i appartient à au plus un cluster
- ▶ La croyance concernant l'appartenance d'un \mathbf{X}_i est représenté par une fonction de masse m_i sur Ω :
 - ▶ la masse $m_i(A)$ ($A \subseteq \Omega$) représente la croyance sur la proposition " \mathbf{X}_i appartient à l'un des clusters de A ".
 - ▶ $m_i(\emptyset)$ mesure la croyance sur le fait que " \mathbf{X}_i n'appartient à aucun des clusters c ".
- ▶ On appelle partition credale le n -tuple $M = (m_1, \dots, m_n)$

Exemple



	\emptyset	$\{\omega_1\}$	$\{\omega_1\}$	$\{\omega_2, \omega_2\}$
m_3	0	1	0	0
m_5	0	0.5	0	0.5
m_6	0	0	0	1
m_{12}	0.9	0	0.1	0

Méthodes de classification

Méthodes de clustering

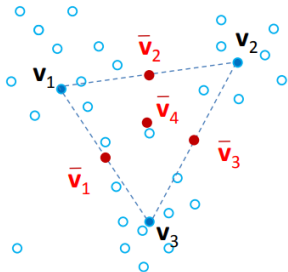
Rappel Clustering

Clustering évidentiel

ECM

EVCLUS

Autres approches



- ▶ Généralisation des approches K-means et Fuzzy C-means
- ▶ Fonctionne avec des données de type attribut
- ▶ Chaque classe ω_k est représentée par un prototype v_k .
- ▶ Les ensembles de classes A_j sont représentés par des prototypes \bar{v}_j définis comme les barycentres des centres v_k tels que $\omega_k \in A_j$.

- ▶ Idée :
 - ▶ Si \mathbf{x}_i est proche de $\bar{\mathbf{v}}_j$ alors $m_{ij} = m_i(A_j)$ doit être grand.
 - ▶ La distance à l'ensemble vide est considérée fixe et égale à δ .
- ▶ Critère à minimiser :

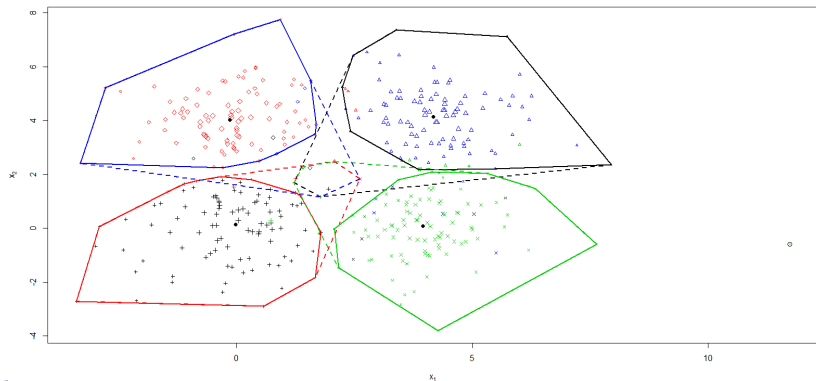
$$J_{ECM}(M, V) = \sum_{i=1}^n \sum_{j/A_j \neq \emptyset, A_j \subseteq \Omega} |A_j|^\alpha m_{ij}^\beta d_{ij}^2 + \sum_{i=1}^n \delta^2 m_{i\emptyset}^\beta$$

- ▶ Paramètres :
 - ▶ α contrôle la spécificité de la fonction de masse
 - ▶ β contrôle la compacité des classes
 - ▶ δ contrôle le nombre d'objets considérés comme aberrants (outliers).
- ▶ Minimiser J_{ECM} en fixant alternativement M et V .

- ▶ Masson, M.-H. et Denœux, T. 2008. "ECM: An evidential version of the fuzzy c-means algorithm." Pattern Recognition, Vol. 41, Issue 4, pages 1384– 1397.
- ▶ Package R : evclust
(<https://cran.r-project.org/web/packages/evclust/index.html>)
- ▶ Extensions de cette approche :
 - ▶ Relationnel Evidentiel C-Means (RECM) : pour les données de dissimilarité (Masson et Denœux, 2009)
 - ▶ Evidentiel C-Means contraints (CECM) : contraintes de liens (ou non) entre objets (Antoine et al., 2011)

Exemple

```
data(fourclass)
x<-fourclass[,1:2]
y<-fourclass[,3]
clus<-ecm(x,c=4,type="simple",disp=TRUE,delta=4.4)
plot(clus,X=x,mfrow=c(2,2),ytrue=y,Outliers=TRUE,Approx=1)
```



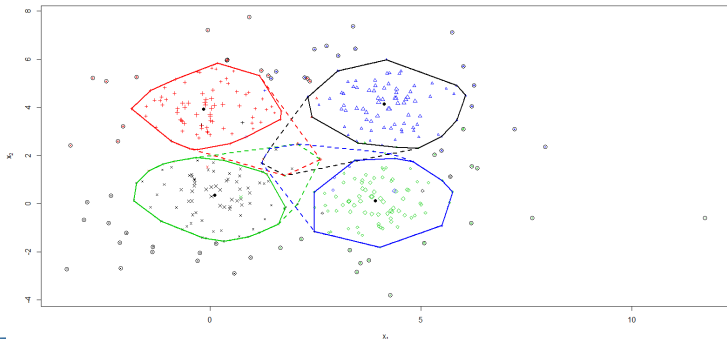
Exercice n°4

- 1. Énoncé :** Regarder l'influence du paramètre δ (en prenant par exemple les valeurs suivantes : 2, 4 et 5).

Correction Exercice n°4 ($\delta = 2$)

Code :

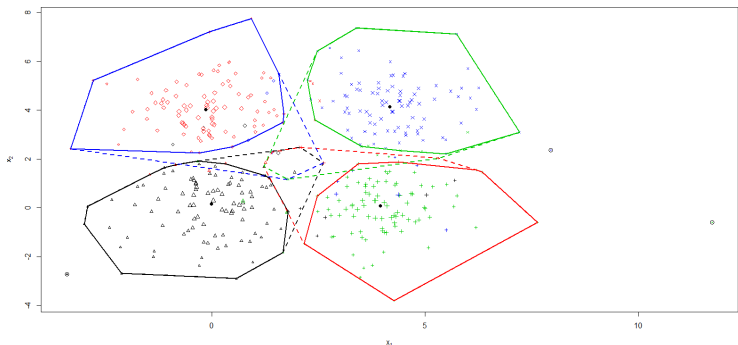
```
data(fourclass)
x<-fourclass[,1:2]
y<-fourclass[,3]
clus<-ecm(x,c=4,disp=TRUE,type="simple",delta=2)
plot(clus,X=x,mfrow=c(1,1),ytrue=y,Outliers=TRUE,Approx=1)
```



Correction Exercice n°4 ($\delta = 4$)

Code :

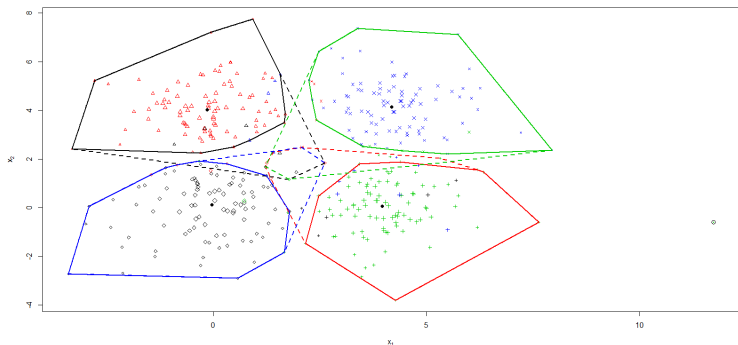
```
data(fourclass)
x<-fourclass[,1:2]
y<-fourclass[,3]
clus<-ecm(x,c=4,disp=TRUE,type="simple",delta=4)
plot(clus,X=x,mfrow=c(1,1),ytrue=y,Outliers=TRUE,Approx=1)
```



Correction Exercice n°4 ($\delta = 5$)

Code :

```
data(fourclass)
x<-fourclass[,1:2]
y<-fourclass[,3]
clus<-ecm(x,c=4,disp=TRUE,type="simple",delta=5)
plot(clus,X=x,mfrow=c(1,1),ytrue=y,Outliers=TRUE,Approx=1)
```



Méthodes de classification

Méthodes de clustering

Rappel Clustering

Clustering évidentiel

ECM

EVCLUS

Autres approches

- ▶ Problème : à partir d'une matrice de dissimilarité $D = (d_{ij})$, comment construire une partition crédale ?
- ▶ Nécessité de construire une relation entre l'appartenance et la matrice de dissimilarité.
- ▶ Idée : "Plus deux objets sont similaires, plus il est possible qu'ils appartiennent au même cluster".

- ▶ Soit m_i et m_j deux fonctions de masse qui représentent le degré d'appartenance de deux objets \mathbf{X}_i et \mathbf{X}_j .
- ▶ Il est possible de montrer que :

$$Pl_{ij}(\{S_{ij}\}) = \sum_{A \cap B \neq \emptyset} m_i(A)m_j(B) = 1 - \kappa_{ij}$$

avec κ_{ij} le degré de conflit entre m_i et m_j .

- ▶ Le problème est alors d'obtenir la partition crédale $M = (m_1, \dots, m_n)$ de manière à ce que les plus grandes valeurs de conflits κ_{ij} correspondent aux plus grandes dissimilarités d_{ij} .

- ▶ Objectif réduire la différence entre la dissimilarité d_{ij} et le conflit κ_{ij} :

$$J(M) = \sum_{i < j} (\kappa_{ij} - \phi(d_{ij}))^2$$

où ϕ est une fonction décroissante

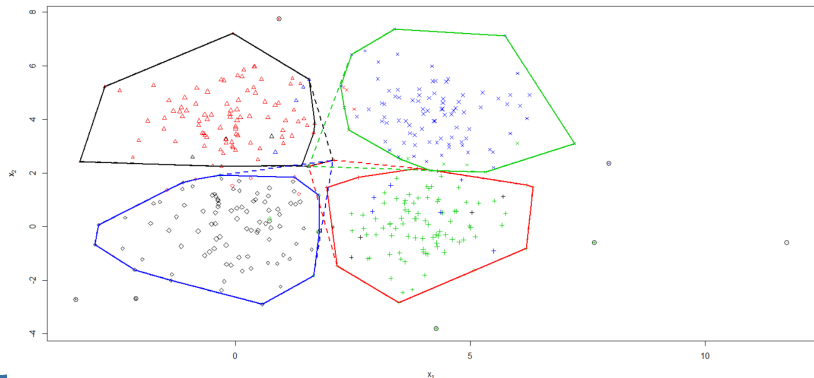
- ▶ Problème : cette approche nécessite le stockage de la matrice de dissimilarité (ce qui est impossible dans le cas d'un grand nombre de données).
- ▶ Solution : sélectionner aléatoirement quelques dissimilarités (k-EVCLUS : 10.000-100.000 objets et une centaine de clusters).

Références et Codes

- ▶ Denœux T. et Masson M.-H. "EVCLUS: Evidential Clustering of Proximity Data." IEEE Transactions on Systems, Man and Cybernetics B, Vol. 34, Issue 1, 95-109, 2004.
- ▶ Denoeux T., Sriboonchitta S. et Kanjanatarakul O. "Evidential clustering of large dissimilarity data". Knowledge-Based Systems, vol. 106, pages 179-195, 2016.
- ▶ Package R : evclust
(<https://cran.r-project.org/web/packages/evclust/index.html>)
- ▶ extensions de cette approche :
 - ▶ EVCLUS constraint : contraintes de liens (ou non) entre objets (Antoine et al., 2014)
 - ▶ K-EVCLUS constraint (Li et al., 2018)

Exemple

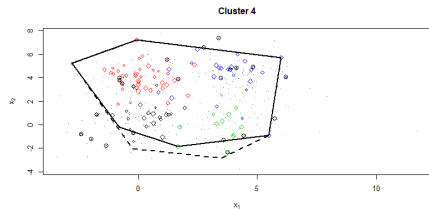
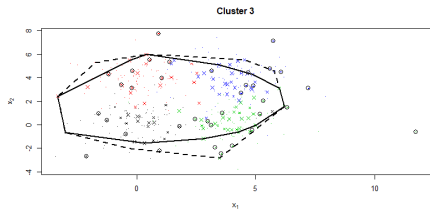
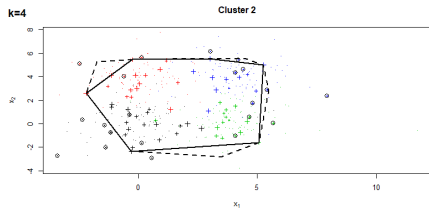
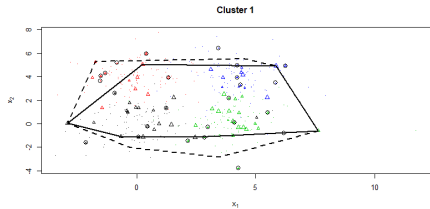
```
data(fourclass)
x<-fourclass[,1:2]
y<-fourclass[,3]
clus<-kevclus(x=x,k=100,c=c,type="simple",disp="TRUE")
plot(clus,X=x,mfrow=c(2,2),ytrue=y,Outliers=TRUE,Approx=1)
```



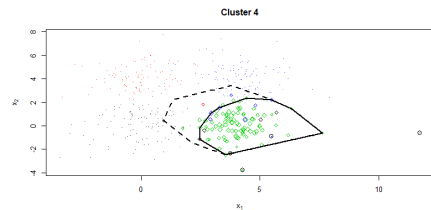
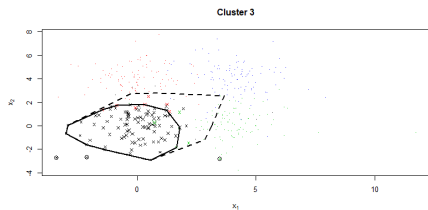
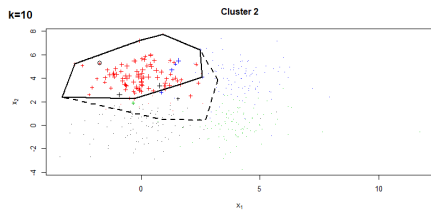
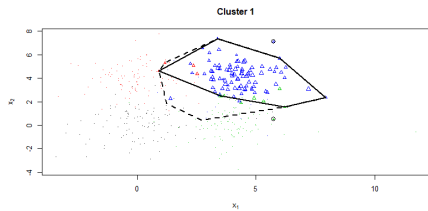
Exercice n°5

- 1. Énoncé :** En reprenant l'exemple précédent, étudier l'influence de la valeur de k ?

Résultats $k = 4$



Résultats $k = 10$



Méthodes de classification

Méthodes de clustering

Rappel Clustering

Clustering évidentiel

ECM

EVCLUS

Autres approches

- ▶ D'autres approches de clustering évidentiel à découvrir dans le package evclust :
 - ▶ EkNNclus (algorithme EkNNclus) : détermine automatiquement le nombre de classes. Démarrage avec autant de classes que d'objets puis E-KNN et enfin calcul de la partition évidentielle (Denœux et al. 2015).
 - ▶ nnevclus (algorithme NN-EVCLUS) : Version réseau de neurones de kevclus (Denœux, 2021).
 - ▶ bpec (Belief Peak Evidential Clustering) : identique à ECM mais les centres sont déterminés par un graphe et sont ensuite fixes (Su et al., 2019).

References

1. T. Denœux, O. Kanjanatarakul et S. Sriboonchitta. EK-NNclus: a clustering procedure based on the evidential K-nearest neighbor rule. Knowledge-Based Systems, Vol. 88, pages 57–69, 2015.
2. T. Denœux. NN-EVCLUS: Neural Network-based Evidential Clustering. Information Sciences, Vol. 572, Pages 297-330, 2021
3. Z.-G. Su et T. Denœux. BPEC: Belief-Peaks Evidential Clustering. IEEE Transactions on Fuzzy Systems, 27(1):111-123, 2019.
4. M.-H. Masson et T. Denœux. RECM: Relational Evidential c-means algorithm. Pattern Recognition Letters, Vol. 30, pages 1015–1026, 2009
5. V. Antoine, B. Quost, M.-H. Masson et T. Denœux. CECM: Constrained Evidential C-Means algorithm. Computational Statistics and Data Analysis, Vol. 56, Issue 4, pages 894–914, 2012.
6. V. Antoine, B. Quost, M.-H. Masson et T. Denœux. CEVCLUS: Evidential clustering with instance-level constraints for relational data. Soft Computing, Volume 18, Issue 7, pp 1321-1335, 2014.
7. F. Li, S. Li et T. Denœux. k-CEVCLUS: Constrained Evidential Clustering of Large Dissimilarity Data. Knowledge-Based Systems, Vol. 142, Pages 29-44, 2018.

Merci pour votre participation à cet atelier