



HAL
open science

Mixture Models with Missing data Classification of Satellite Image Time Series

Serge Iovleff, Mathieu Fauvel, Stéphane Girard, Cristian Preda, Vincent
Vandewalle

► **To cite this version:**

Serge Iovleff, Mathieu Fauvel, Stéphane Girard, Cristian Preda, Vincent Vandewalle. Mixture Models with Missing data Classification of Satellite Image Time Series. Journées Science des Données MaDICS 2017, Jun 2017, Marseille, France. pp.1-60. hal-01649206

HAL Id: hal-01649206

<https://hal.science/hal-01649206v1>

Submitted on 27 Nov 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Mixture Models with Missing data

Classification of Satellite Image Time Series

QUALIMADOS: Atelier Qualité des masses de données scientifiques

S. Iovleff, Mathieu Fauvel, Stéphane Girard, Cristian Preda, Vincent Vandewalle

Laboratoire Paul Painlevé

23 Juin 2017



Sommaire

Clustering using Mixture Models

- What is Clustering ?

- Example

- Mixture Models

- EM Algorithm and variations

- Mixture Model and Mixed Data

Classification of Satellite Image Time Series

Clustering is the cluster building process

Cluster analysis

From Wikipedia, the free encyclopedia

For the supervised learning approach, see [Statistical classification](#).

Cluster analysis or **clustering** is the task of grouping a set of objects in such a way that objects in the same group (called a **cluster**) are more similar (in some sense or another) to each other than to those in other groups (clusters). It is a main task of exploratory [data mining](#), and a common technique for [statistical data analysis](#), used in many fields, including [machine learning](#), [pattern recognition](#), [image analysis](#), [information retrieval](#), [bioinformatics](#), [data compression](#), and [computer graphics](#).

- ▶ The term [Data Clustering](#) first appeared in 1954 (according to JSTOR) in an article dealing with anthropological data,
- ▶ Many, many existing methods (https://en.wikipedia.org/wiki/Category:Data_clustering_algorithms)

Clustering is the cluster building process

Cluster analysis

From Wikipedia, the free encyclopedia

For the supervised learning approach, see [Statistical classification](#).

Cluster analysis or **clustering** is the task of grouping a set of objects in such a way that objects in the same group (called a **cluster**) are more similar (in some sense or another) to each other than to those in other groups (clusters). It is a main task of exploratory [data mining](#), and a common technique for [statistical data analysis](#), used in many fields, including [machine learning](#), [pattern recognition](#), [image analysis](#), [information retrieval](#), [bioinformatics](#), [data compression](#), and [computer graphics](#).

- ▶ The term [Data Clustering](#) first appeared in 1954 (according to JSTOR) in an article dealing with anthropological data,
- ▶ Many, many existing methods
(https://en.wikipedia.org/wiki/Category:Data_clustering_algorithms)

Clustering is the cluster building process

Cluster analysis

From Wikipedia, the free encyclopedia

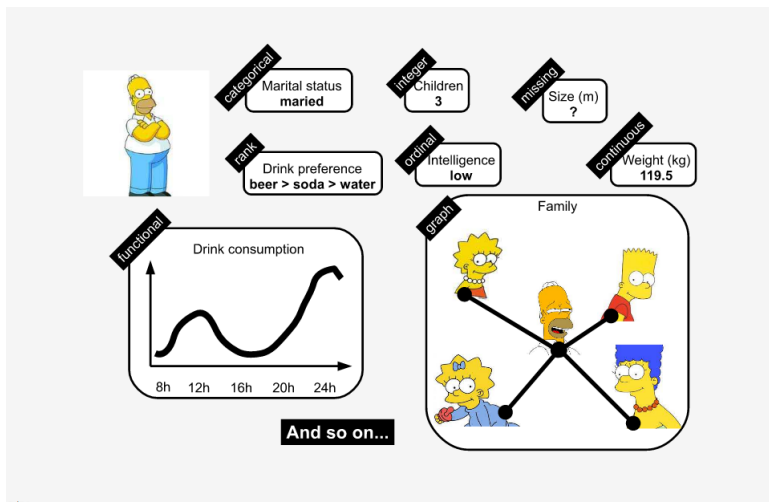
For the supervised learning approach, see [Statistical classification](#).

Cluster analysis or **clustering** is the task of grouping a set of objects in such a way that objects in the same group (called a **cluster**) are more similar (in some sense or another) to each other than to those in other groups (clusters). It is a main task of exploratory [data mining](#), and a common technique for [statistical data analysis](#), used in many fields, including [machine learning](#), [pattern recognition](#), [image analysis](#), [information retrieval](#), [bioinformatics](#), [data compression](#), and [computer graphics](#).

- ▶ The term [Data Clustering](#) first appeared in 1954 (according to JSTOR) in an article dealing with anthropological data,
- ▶ Many, many existing methods (https://en.wikipedia.org/wiki/Category:Data_clustering_algorithms)

New challenges

Need to **algorithms** for Big-Data and Complex Data. In particular mixed features and missing values



An example

Joint works with Christophe Biernacki (head of the Inria Modal team), Vincent Vandewalle, Komi Nagbe,...

Contract for a large lingerie store: "*Clustering cash receipts of the Customers with a loyalty card*"

- ▶ 28 variables related to products,
- ▶ 6 variables related to costumers,
- ▶ 8 variables related to stores,
- ▶ $n = 2,899,030$ receipts.

Some meaningful variables with missing values.



An example

Joint works with Christophe Biernacki (head of the Inria Modal team), Vincent Vandewalle, Komi Nagbe,...

Contract for a large lingerie store: "*Clustering cash receipts of the Customers with a loyalty card*"

- ▶ 28 variables related to products,
- ▶ 6 variables related to costumers,
- ▶ 8 variables related to stores,
- ▶ $n = 2,899,030$ receipts.



Some meaningful variables with missing values.

An example

Joint works with Christophe Biernacki (head of the Inria Modal team), Vincent Vandewalle, Komi Nagbe,...

Contract for a large lingerie store: "*Clustering cash receipts of the Customers with a loyalty card*"

- ▶ 28 variables related to products,
- ▶ 6 variables related to costumers,
- ▶ 8 variables related to stores,
- ▶ $n = 2,899,030$ receipts.



Some meaningful variables with missing values.

An example

Joint works with Christophe Biernacki (head of the Inria Modal team), Vincent Vandewalle, Komi Nagbe,...

Contract for a large lingerie store: "*Clustering cash receipts of the Customers with a loyalty card*"

- ▶ 28 variables related to products,
- ▶ 6 variables related to costumers,
- ▶ 8 variables related to stores,
- ▶ $n = 2,899,030$ receipts.

Some meaningful variables with missing values.



An example (Variables)

Variables liées aux clients

Num	Name_var	Type	Nbre Modal	Modalités	% Manquant	Discretisé
1	CODE_CIVILITE	factor	4	M E M MM E MR	32.55	NON
2	TRANCHE_URBAINE	factor	5	+10000 20000 à 50000 50000 à 100000 5000 à 20000	33.37	NON
3	[REDACTED]	factor	2	NON OUI	0	NON
4	[REDACTED]	factor	2	NON OUI	0	NON
5	S ANNEE_NAISSANCE (Age)	numeric	106		0 37.07	NON
6	[REDACTED]	integer	225		0 32.55	NON

Variables liées aux magasins

Num	Name_var	Type	Nbre Modal	Modalités	% Manquant	Discretisé
1	TYPE_MAGASIN	factor	2	FRANCHISE SUCCURSALE	0	NON
2	SECTEUR_MAGASIN	factor	10	[REDACTED]	0	NON
3	LOCALISATION_MAGASIN	factor	4	CENTRE_CIALAVEC_HYPERMARCHE CENTRE_CIAL_SANS_HYPERMARCHE	0.41	NON
4	MUSIQUE_MAGASIN	factor	2	NON OUI	0	NON
5	magasin_ferme	factor	2	FERME NON_FERME	0	NON
6	[REDACTED]	factor	4	1-Sans 6-10ans 11-15ans plus de 15ans	0	NON
7	[REDACTED]	factor	28		0	NON
8	SUPERFICIE_MAGASIN	integer	53		0 0	NON

An example (Variables)

Variables liées aux Produits

Num	Name_var	Type	Nbre Mod	Modalités	% Manqua	Discrets
1	mois_ticket	factor	12	01 02 03 04 05 06 07 08 09 10 11 12	0	0
2	jours_ticket	factor	7	d m anche jeu di lun di m ardi mercredi samedi ven dredi	0	0
3	TYPE_LIGNE	factor			0	0
5	MONTANT_REMISE	factor	2	REMISE SANS REMISE	0	OUI
6	SOLDE	factor	2	NON OUI	0	0
7	MT_REMISE_OP_COMM	factor	2	REMISE COMM SANS REMISE COMM	0	OUI
		factor	2	NON OUI	0	0
		factor	2	NON OUI	0	0
10	REGLEMENT_CB	factor	2	NON OUI	0	0
11	REGLEMENT_ES	factor	2	NON OUI	0	0
12	REGLEMENT_CH	factor	2	NON OUI	0	0
13	REGLEMENT_AUTRES	factor	2	NON OUI	0	0
14	KKDO_UTILISATION	factor	2	NON OUI	0	0
15		factor	1	NON	0	0
16		factor	2	NON OUI	0	0
17		factor	2	NON OUI	0	0
18	ACT_NELLE_CLIENTE	factor	2	NON OUI	0	0
19		factor	2	NON OUI	0	0
20		factor	2	NON OUI	0	0
21	GROUPE_PRODUIT	factor	5	ACCESSOIRES MODE BALNEAIRE BAS COULEURS LINGERIE DE JOUR LIN	0	0
22	COLLECTION	factor	2		0	0
23	COLORIS_BASE	factor	13	Blanc Bleu Cyan Gris Jaune Marron Noir Orange Peau Rose Rouge Vert Violet	0	0
24	COORDONNE	factor	4		19.02	0
25	STYLE_PORTER	factor	4	CONFORTABLE SEDUCTION SEXY	19.74	0
26	ASPECT_COLORIS	factor	3	BICOLORE IMPRIME UNI	2.35	0
27	CARACT_MATIERE	factor	5	AUTRES DENTELLE MICROFIBRE SATIN TULLE BRODE	0	0
4	PRIX_UNITAIRE	numeric	4811		0	0

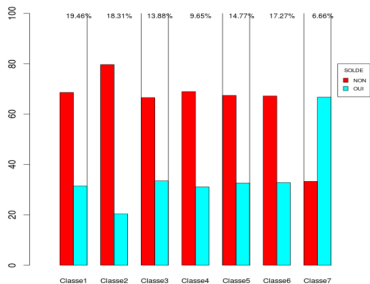
An example (Results)

	Classe1	Classe2	Classe3	Classe4	Classe5	Classe6	Classe7
Proportion		18.31%			14.77%		6.66%
Effectif		527499			427417		205443

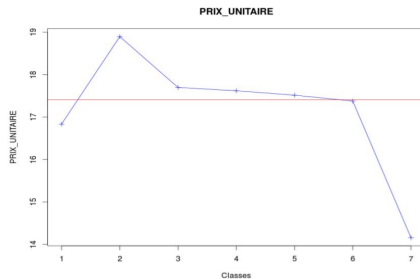
An example (Results)

	Classe1	Classe2	Classe3	Classe4	Classe5	Classe6	Classe7
Proportion		18.31%			14.77%		6.66%
Effectif		527499			427417		205443

Solde



Prix Unitaire

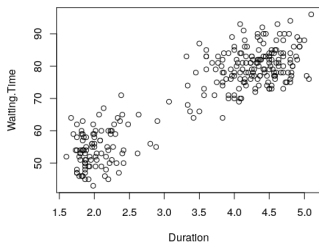


Mixture Models

Main Idea

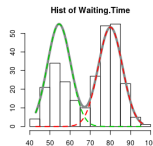
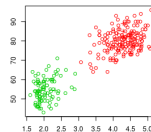
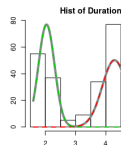
\mathbf{x} in cluster $k \iff \mathbf{x}$ belongs to distribution P_k

$$\mathbf{x} = (\mathbf{x}_1, \dots, \mathbf{x}_n)$$



→
clustering

$$\hat{\mathbf{z}} = (\hat{\mathbf{z}}_1, \dots, \hat{\mathbf{z}}_n), \hat{K} \text{ clusters}$$



Model Based clustering is a probabilistic approach.

R package MixAll and SaaS MixtComp

Two softwares available

► R package **MixAll**

```
> library(MixAll)
> data(geyser)
> ## add 10 missing values as random
> x = as.matrix(geyser); n <- nrow(x); p <- ncol(x);
> indexes <- matrix(c(round(runif(5,1,n)), round(runif(5,1,p))), ncol=2);
> x[indexes] <- NA;
> ## estimate model
> model <- clusterDiagGaussian( data=x, nbCluster=2:3, models=c( "gaussian_pk_sjk" ))
> plot(model)
> missingValues(model)
  row col  value
1  133   1 2.029661
2   42   2 54.569144
3   49   2 79.970973
4  209   2 54.569144
5  213   2 54.569144
```

► SaaS software **MixtComp** <https://massiccc.lille.inria.fr/>

Hypothesis of mixture of parametric distributions

- ▶ Cluster k is modeled by a **parametric distribution**

$$\mathbf{x}_j | z = k \sim p(\cdot | \alpha_k)$$

- ▶ Cluster k has probability π_k

$$z_i \sim \mathcal{M}(\mathbf{1}, \pi_1, \dots, \pi_K).$$

Mixture model

The model parameters are $\theta = (\pi_1, \dots, \pi_K, \alpha_1, \dots, \alpha_K)$ and

$$p(\mathbf{x}_i) = \sum_{k=1}^K \pi_k p(\mathbf{x}_i; \alpha_k)$$

EM Algorithm

Starting from an initial arbitrary parameter θ^0 , the m th iteration of the **EM** algorithm consists of repeating the following **I**, **E** and **M** steps.

- ▶ **I step:** *Impute* by using **expectation** of the missing values \mathbf{x}^m using \mathbf{x}^o , θ^{r-1} , t_{ik}^{r-1} .
- ▶ **E step:** Compute **conditional probabilities** $z_i = k | \mathbf{x}_i$ using current value θ^{r-1} of the parameter:

$$t_{ik}^r = t_k^r(\mathbf{x}_i | \theta^{r-1}) = \frac{p_k^{r-1} h(\mathbf{x}_i | \alpha_k^{r-1})}{\sum_{l=1}^K p_l^{r-1} h(\mathbf{x}_i | \alpha_l^{r-1})}. \quad (1)$$

- ▶ **M step:** Update **ML estimate** θ^r using conditional probabilities t_{ik}^r as mixing weights

$$L(\theta | \mathbf{x}_1, \dots, \mathbf{x}_n, \mathbf{t}^r) = \sum_{i=1}^n \sum_{k=1}^K t_{ik}^r \ln [p_k h(\mathbf{x}_i | \alpha_k)],$$

- ▶ Iterate until convergence

EM Algorithm

Starting from an initial arbitrary parameter θ^0 , the m th iteration of the **EM** algorithm consists of repeating the following **I**, **E** and **M** steps.

- ▶ **I step:** *Impute* by using **expectation** of the missing values \mathbf{x}^m using \mathbf{x}^o , θ^{r-1} , t_{ik}^{r-1} .
- ▶ **E step:** Compute **conditional probabilities** $z_i = k | \mathbf{x}_i$ using current value θ^{r-1} of the parameter:

$$t_{ik}^r = t_{ik}^r(\mathbf{x}_i | \theta^{r-1}) = \frac{p_k^{r-1} h(\mathbf{x}_i | \alpha_k^{r-1})}{\sum_{l=1}^K p_l^{r-1} h(\mathbf{x}_i | \alpha_l^{r-1})}. \quad (1)$$

- ▶ **M step:** Update **ML estimate** θ^r using conditional probabilities t_{ik}^r as mixing weights

$$L(\theta | \mathbf{x}_1, \dots, \mathbf{x}_n, \mathbf{t}^r) = \sum_{i=1}^n \sum_{k=1}^K t_{ik}^r \ln [p_k h(\mathbf{x}_i | \alpha_k)],$$

- ▶ Iterate until convergence

EM Algorithm

Starting from an initial arbitrary parameter θ^0 , the m th iteration of the **EM** algorithm consists of repeating the following **I**, **E** and **M** steps.

- ▶ **I step:** *Impute* by using **expectation** of the missing values \mathbf{x}^m using \mathbf{x}^o , θ^{r-1} , t_{ik}^{r-1} .
- ▶ **E step:** Compute **conditional probabilities** $z_i = k|\mathbf{x}_i$ using current value θ^{r-1} of the parameter:

$$t_{ik}^r = t_k^r(\mathbf{x}_i|\theta^{r-1}) = \frac{p_k^{r-1} h(\mathbf{x}_i|\alpha_k^{r-1})}{\sum_{l=1}^K p_l^{r-1} h(\mathbf{x}_i|\alpha_l^{r-1})}. \quad (1)$$

- ▶ **M step:** Update **ML estimate** θ^r using conditional probabilities t_{ik}^r as mixing weights

$$L(\theta|\mathbf{x}_1, \dots, \mathbf{x}_n, \mathbf{t}^r) = \sum_{i=1}^n \sum_{k=1}^K t_{ik}^r \ln [p_k h(\mathbf{x}_i|\alpha_k)],$$

- ▶ Iterate until convergence

EM Algorithm

Starting from an initial arbitrary parameter θ^0 , the m th iteration of the **EM** algorithm consists of repeating the following **I**, **E** and **M** steps.

- ▶ **I step:** *Impute* by using **expectation** of the missing values \mathbf{x}^m using \mathbf{x}^o , θ^{r-1} , t_{ik}^{r-1} .
- ▶ **E step:** Compute **conditional probabilities** $z_i = k | \mathbf{x}_i$ using current value θ^{r-1} of the parameter:

$$t_{ik}^r = t_k^r(\mathbf{x}_i | \theta^{r-1}) = \frac{p_k^{r-1} h(\mathbf{x}_i | \alpha_k^{r-1})}{\sum_{l=1}^K p_l^{r-1} h(\mathbf{x}_i | \alpha_l^{r-1})}. \quad (1)$$

- ▶ **M step:** Update **ML estimate** θ^r using conditional probabilities t_{ik}^r as mixing weights

$$L(\theta | \mathbf{x}_1, \dots, \mathbf{x}_n, \mathbf{t}^r) = \sum_{i=1}^n \sum_{k=1}^K t_{ik}^r \ln [p_k h(\mathbf{x}_i | \alpha_k)],$$

- ▶ Iterate until convergence

EM Algorithm

Starting from an initial arbitrary parameter θ^0 , the m th iteration of the **EM** algorithm consists of repeating the following **I**, **E** and **M** steps.

- ▶ **I step:** *Impute* by using **expectation** of the missing values \mathbf{x}^m using \mathbf{x}^o , θ^{r-1} , t_{ik}^{r-1} .
- ▶ **E step:** Compute **conditional probabilities** $z_i = k|\mathbf{x}_i$ using current value θ^{r-1} of the parameter:

$$t_{ik}^r = t_k^r(\mathbf{x}_i|\theta^{r-1}) = \frac{p_k^{r-1} h(\mathbf{x}_i|\alpha_k^{r-1})}{\sum_{l=1}^K p_l^{r-1} h(\mathbf{x}_i|\alpha_l^{r-1})}. \quad (1)$$

- ▶ **M step:** Update **ML estimate** θ^r using conditional probabilities t_{ik}^r as mixing weights

$$L(\theta|\mathbf{x}_1, \dots, \mathbf{x}_n, \mathbf{t}^r) = \sum_{i=1}^n \sum_{k=1}^K t_{ik}^r \ln [p_k h(\mathbf{x}_i|\alpha_k)],$$

- ▶ Iterate until convergence

SEM/SemiSEM Algorithms

Drawbacks

- ▶ The I step may be difficult
- ▶ EM algorithm may converge slowly and is slowed down by the imputation step
- ▶ Biased estimators

Solution: Use Monte Carlo

- ▶ Replace I step by a simulation step
- ▶ IS step: simulate missing values \mathbf{x}^m using \mathbf{x}^o , θ^{r-1} , t_{ik}^{r-1} .
- ▶ Replace E step by a simulation step (Optional)
- ▶ S step: generate labels $\mathbf{z}^r = \{z_1^r, \dots, z_n^r\}$ according to the categorical distribution $(t_{ik}^r, 1 \leq k \leq K)$.

SEM and SemiSEM does not converge point wise. It generates a Markov chain.

- ▶ $\bar{\theta} = (\theta^r)_{r=1, \dots, R}$
- ▶ missing values imputed using empirical MAP value (or expectation)

SEM/SemiSEM Algorithms

Drawbacks

- ▶ The I step may be difficult
- ▶ EM algorithm may converge slowly and is slowed down by the imputation step
- ▶ Biased estimators

Solution: Use Monte Carlo

- ▶ Replace I step by a simulation step
- ▶ IS step: simulate missing values \mathbf{x}^m using \mathbf{x}^o , θ^{r-1} , t_{ik}^{r-1} .
- ▶ Replace E step by a simulation step (Optional)
- ▶ S step: generate labels $\mathbf{z}^r = \{z_1^r, \dots, z_n^r\}$ according to the categorical distribution $(t_{ik}^r, 1 \leq k \leq K)$.

SEM and SemiSEM does not converge point wise. It generates a Markov chain.

- ▶ $\bar{\theta} = (\theta^r)_{r=1, \dots, R}$
- ▶ missing values imputed using empirical MAP value (or expectation)

SEM/SemiSEM Algorithms

Drawbacks

- ▶ The I step may be difficult
- ▶ **EM** algorithm may converge slowly and is slowed down by the imputation step
- ▶ **Biased** estimators

Solution: Use Monte Carlo

- ▶ Replace I step by a simulation step
- ▶ **IS step**: **simulate** missing values \mathbf{x}^m using \mathbf{x}^o , θ^{r-1} , t_{ik}^{r-1} .
- ▶ Replace **E** step by a simulation step (Optional)
- ▶ **S step**: **generate** labels $\mathbf{z}^r = \{z_1^r, \dots, z_n^r\}$ according to the categorical distribution $(t_{ik}^r, 1 \leq k \leq K)$.

SEM and **SemiSEM** does not converge point wise. It generates a **Markov chain**.

- ▶ $\bar{\theta} = (\theta^r)_{r=1, \dots, R}$
- ▶ missing values imputed using empirical MAP value (or expectation)

SEM/SemiSEM Algorithms

Drawbacks

- ▶ The I step may be difficult
- ▶ **EM** algorithm may converge slowly and is slowed down by the imputation step
- ▶ **Biased** estimators

Solution: Use Monte Carlo

- ▶ Replace I step by a simulation step
- ▶ **IS step**: simulate missing values \mathbf{x}^m using \mathbf{x}^o , θ^{r-1} , t_{ik}^{r-1} .
- ▶ Replace **E** step by a simulation step (Optional)
- ▶ **S step**: generate labels $\mathbf{z}^r = \{z_1^r, \dots, z_n^r\}$ according to the categorical distribution $(t_{ik}^r, 1 \leq k \leq K)$.

SEM and **SemiSEM** does not converge point wise. It generates a **Markov chain**.

- ▶ $\bar{\theta} = (\theta^r)_{r=1, \dots, R}$
- ▶ missing values imputed using empirical MAP value (or expectation)

SEM/SemiSEM Algorithms

Drawbacks

- ▶ The I step may be difficult
- ▶ **EM** algorithm may converge slowly and is slowed down by the imputation step
- ▶ **Biased** estimators

Solution: Use Monte Carlo

- ▶ Replace I step by a simulation step
- ▶ **IS step**: simulate missing values \mathbf{x}^m using \mathbf{x}^o , θ^{r-1} , t_{ik}^{r-1} .
- ▶ Replace E step by a simulation step (Optional)
- ▶ **S step**: generate labels $\mathbf{z}^r = \{z_1^r, \dots, z_n^r\}$ according to the categorical distribution $(t_{ik}^r, 1 \leq k \leq K)$.

SEM and **SemiSEM** does not converge point wise. It generates a **Markov chain**.

- ▶ $\bar{\theta} = (\theta^r)_{r=1, \dots, R}$
- ▶ missing values imputed using empirical MAP value (or expectation)

SEM/SemiSEM Algorithms

Drawbacks

- ▶ The I step may be difficult
- ▶ **EM** algorithm may converge slowly and is slowed down by the imputation step
- ▶ **Biased** estimators

Solution: Use Monte Carlo

- ▶ Replace I step by a simulation step
- ▶ **IS step:** simulate missing values \mathbf{x}^m using \mathbf{x}^o , θ^{r-1} , t_{ik}^{r-1} .
- ▶ Replace E step by a simulation step (Optional)
- ▶ **S step:** generate labels $\mathbf{z}^r = \{z_1^r, \dots, z_n^r\}$ according to the categorical distribution $(t_{ik}^r, 1 \leq k \leq K)$.

SEM and **SemiSEM** does not converge point wise. It generates a **Markov chain**.

- ▶ $\bar{\theta} = (\theta^r)_{r=1, \dots, R}$
- ▶ missing values imputed using empirical MAP value (or expectation)

SEM/SemiSEM Algorithms

Drawbacks

- ▶ The I step may be difficult
- ▶ **EM** algorithm may converge slowly and is slowed down by the imputation step
- ▶ **Biased** estimators

Solution: Use Monte Carlo

- ▶ Replace I step by a simulation step
- ▶ **IS step:** simulate missing values \mathbf{x}^m using \mathbf{x}^o , θ^{r-1} , t_{ik}^{r-1} .
- ▶ Replace **E** step by a simulation step (Optional)
- ▶ **S step:** generate labels $\mathbf{z}^r = \{z_1^r, \dots, z_n^r\}$ according to the categorical distribution $(t_{ik}^r, 1 \leq k \leq K)$.

SEM and **SemiSEM** does not converge point wise. It generates a **Markov chain**.

- ▶ $\bar{\theta} = (\theta^r)_{r=1, \dots, R}$
- ▶ missing values imputed using empirical MAP value (or expectation)

SEM/SemiSEM Algorithms

Drawbacks

- ▶ The I step may be difficult
- ▶ **EM** algorithm may converge slowly and is slowed down by the imputation step
- ▶ **Biased** estimators

Solution: Use Monte Carlo

- ▶ Replace I step by a simulation step
- ▶ **IS step**: **simulate** missing values \mathbf{x}^m using \mathbf{x}^o , θ^{r-1} , t_{ik}^{r-1} .
- ▶ Replace **E** step by a simulation step (Optional)
- ▶ **S step**: **generate** labels $\mathbf{z}^r = \{z_1^r, \dots, z_n^r\}$ according to the categorical distribution $(t_{ik}^r, 1 \leq k \leq K)$.

SEM and **SemiSEM** does not converge point wise. It generates a **Markov chain**.

- ▶ $\bar{\theta} = (\theta^r)_{r=1, \dots, R}$
- ▶ missing values imputed using empirical MAP value (or expectation)

SEM/SemiSEM Algorithms

Drawbacks

- ▶ The I step may be difficult
- ▶ **EM** algorithm may converge slowly and is slowed down by the imputation step
- ▶ **Biased** estimators

Solution: Use Monte Carlo

- ▶ Replace I step by a simulation step
- ▶ **IS step**: **simulate** missing values \mathbf{x}^m using \mathbf{x}^o , θ^{r-1} , t_{ik}^{r-1} .
- ▶ Replace **E** step by a simulation step (Optional)
- ▶ **S step**: **generate** labels $\mathbf{z}^r = \{z_1^r, \dots, z_n^r\}$ according to the categorical distribution $(t_{ik}^r, 1 \leq k \leq K)$.

SEM and **SemiSEM** does not converge point wise. It generates a **Markov chain**.

- ▶ $\bar{\theta} = (\theta^r)_{r=1, \dots, R}$
- ▶ missing values imputed using empirical MAP value (or expectation)

SEM/SemiSEM Algorithms

Drawbacks

- ▶ The I step may be difficult
- ▶ **EM** algorithm may converge slowly and is slowed down by the imputation step
- ▶ **Biased** estimators

Solution: Use Monte Carlo

- ▶ Replace I step by a simulation step
- ▶ **IS step**: **simulate** missing values \mathbf{x}^m using \mathbf{x}^o , θ^{r-1} , t_{ik}^{r-1} .
- ▶ Replace **E** step by a simulation step (Optional)
- ▶ **S step**: **generate** labels $\mathbf{z}^r = \{z_1^r, \dots, z_n^r\}$ according to the categorical distribution $(t_{ik}^r, 1 \leq k \leq K)$.

SEM and **SemiSEM** does not converge point wise. It generates a **Markov chain**.

- ▶ $\bar{\theta} = (\theta^r)_{r=1, \dots, R}$
- ▶ missing values imputed using empirical MAP value (or expectation)

SEM/SemiSEM Algorithms

Drawbacks

- ▶ The I step may be difficult
- ▶ **EM** algorithm may converge slowly and is slowed down by the imputation step
- ▶ **Biased** estimators

Solution: Use Monte Carlo

- ▶ Replace I step by a simulation step
- ▶ **IS step**: **simulate** missing values \mathbf{x}^m using \mathbf{x}^o , θ^{r-1} , t_{ik}^{r-1} .
- ▶ Replace **E** step by a simulation step (Optional)
- ▶ **S step**: **generate** labels $\mathbf{z}^r = \{z_1^r, \dots, z_n^r\}$ according to the categorical distribution $(t_{ik}^r, 1 \leq k \leq K)$.

SEM and **SemiSEM** does not converge point wise. It generates a **Markov chain**.

- ▶ $\bar{\theta} = (\theta^r)_{r=1, \dots, R}$
- ▶ missing values imputed using empirical MAP value (or expectation)

Mixed Data

Mixed data are handled using **conditional independence** of the variables.

1. Observation space of the form $\mathbf{X} = \mathbf{X}_1 \times \mathbf{X}_2 \times \dots \times \mathbf{X}_L$
2. \mathbf{x}_i arises from a mixture probability distribution with density

$$f(\mathbf{x}_i = (\mathbf{x}_{1i}, \mathbf{x}_{2i}, \dots, \mathbf{x}_{Li}) | \theta) = \sum_{k=1}^K \pi_k \prod_{l=1}^L h^l(\mathbf{x}_{li} | \alpha_{lk}).$$

3. The density functions (or probability distribution functions) $h^l(\cdot | \alpha_{lk})$ can be any implemented model.

MixAll implements Gaussian, Poisson, Categorical, Gamma distributions.

MixComp implements Gaussian, Poisson, Categorical and specific distributions for rank and ordinal data.

Sommaire

Clustering using Mixture Models

Classification of Satellite Image Time Series

- Cube of Data

- Missing Data/Noisy Data/Sampling

- (Long term) Objective

- Modeling

- Missing Values ?

- ▶ Défi Mastodons: Appel à Projet 2016 "Qualité des données"
- ▶ Creation of the CloHe (CLustering Of Heterogeneous Data with applications to satellite data records) project
- ▶ Members: Mathieu Fauvel (INRA), Stéphane Girard (Inria Grenoble), Vincent vandewalle (Lille2), Crisitan Preda (Université Lille 1)

<https://modal.lille.inria.fr/CloHe/>

Formosat-2 is no more operational



Figure: Formosat-2 furnished multi-spectral data (R, G, B, NIR) with a 8 meter resolution. 17 complete images of France by year

Sentinel-2A start service in 2016.



Figure: Sentinel-2 furnish 13 spectral bandwidths with 4 bandwidths with a 10 meters resolution and 6 bandwidths with a 20 meters resolution. A complete image of France every 5 days

Data Cube

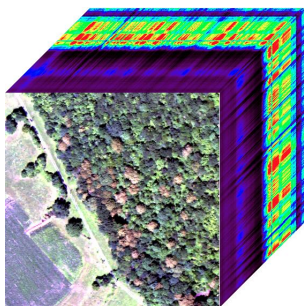


Figure: Sentinel-2 furnish approximately 20TB of images/year, and cover the entire France in 5 days with 1.6 milliard de pixels.

Data Cube

$$X = (X_{ikt}), \quad i \in I, \quad k \in \{r,v,b,ir\},$$

$$Y = (Y_i), \quad i \in J \subset I.$$

with

- ▶ $i = (x, y)$ geographic position,
- ▶ k spectral band,
- ▶ t dates,
- ▶ missing values (clouds, ported shadows) at some dates and some positions,
- ▶ noisy data (undetected shadows, cloud veil, etc...),
- ▶ mixel (mixture of pixel)

Data Cube

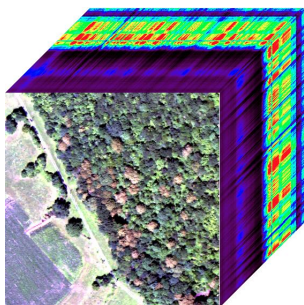


Figure: Sentinel-2 furnish approximately 20TB of images/year, and cover the entire France in 5 days with 1.6 milliard de pixels.

Data Cube

$$\mathbf{X} = (X_{ikt}), \quad i \in I, \quad k \in \{r,v,b,ir\},$$

$$\mathbf{Y} = (Y_i), \quad i \in J \subset I.$$

with

- ▶ $i = (x, y)$ geographic position,
- ▶ k spectral band,
- ▶ t dates,
- ▶ missing values (clouds, ported shadows) at some dates and some positions,
- ▶ noisy data (undetected shadows, cloud veil, etc...).
- ▶ mixel (mixture of pixel)

Data Cube

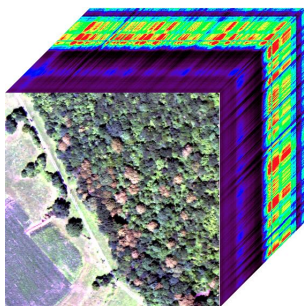


Figure: Sentinel-2 furnish approximately 20TB of images/year, and cover the entire France in 5 days with 1.6 milliard de pixels.

Data Cube

$$\mathbf{X} = (X_{ikt}), \quad i \in I, \quad k \in \{r,v,b,ir\},$$

$$\mathbf{Y} = (Y_i), \quad i \in J \subset I.$$

with

- ▶ $i = (x, y)$ geographic position,
- ▶ k spectral band,
- ▶ t dates,
- ▶ **missing values** (clouds, ported shadows) at some dates and some positions,
- ▶ **noisy data** (undetected shadows, cloud veil, etc...).
- ▶ **mixel** (mixture of pixel)

Data Cube

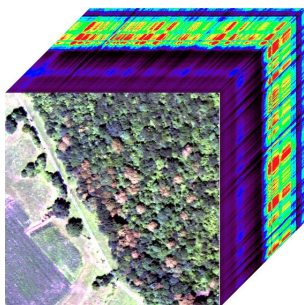


Figure: Sentinel-2 furnish approximately 20TB of images/year, and cover the entire France in 5 days with 1.6 milliard de pixels.

Data Cube

$$\mathbf{X} = (X_{ikt}), \quad i \in I, \quad k \in \{r,v,b,ir\},$$

$$\mathbf{Y} = (Y_i), \quad i \in J \subset I.$$

with

- ▶ $i = (x, y)$ geographic position,
- ▶ k spectral band,
- ▶ t dates,
- ▶ **missing values** (clouds, ported shadows) at some dates and some positions,
- ▶ **noisy data** (undetected shadows, cloud veil, etc...).
- ▶ **mixel** (mixture of pixel)

Data Cube

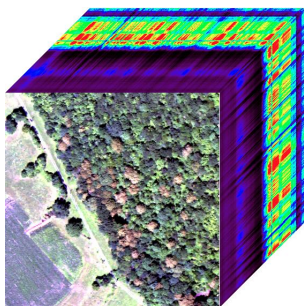


Figure: Sentinel-2 furnish approximately 20TB of images/year, and cover the entire France in 5 days with 1.6 milliard de pixels.

Data Cube

$$\mathbf{X} = (X_{ikt}), \quad i \in I, \quad k \in \{r,v,b,ir\},$$

$$\mathbf{Y} = (Y_i), \quad i \in J \subset I.$$

with

- ▶ $i = (x, y)$ geographic position,
- ▶ k spectral band,
- ▶ t dates,
- ▶ **missing values** (clouds, ported shadows) at some dates and some positions,
- ▶ **noisy data** (undetected shadows, cloud veil, etc...).
- ▶ **mixel** (mixture of pixel)

Missing data

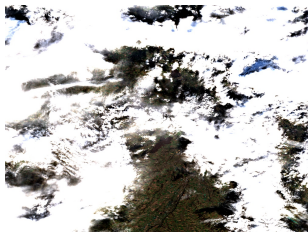


Figure: Very cloudy



Figure: A few number of clouds



Figure: "sheeps"

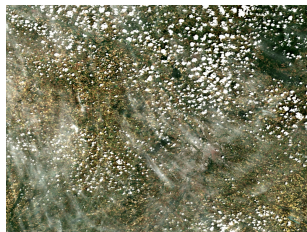


Figure: Some clouds with a veil

Noisy Data

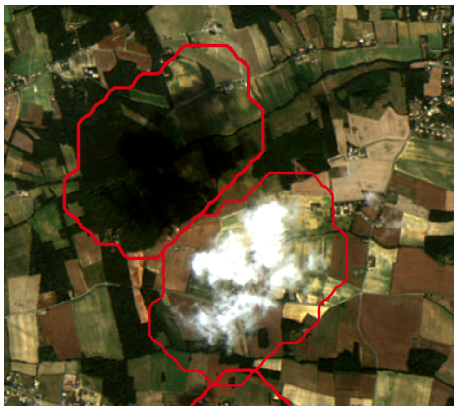


Figure: clouds and their shadows

Non-Uniform sampling

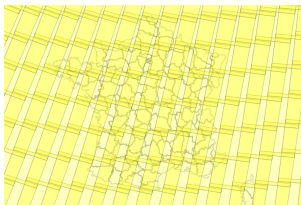


Figure: Path-row grid for Landsat acquisitions. Every path (North-South track) is acquired on the same date every 16 days.

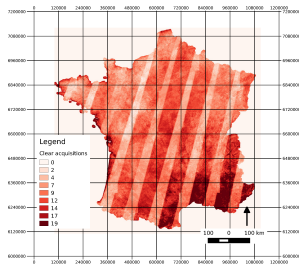


Figure: Map of the number of times that every pixel sees the ground taking into account satellite revisit and cloud cover.

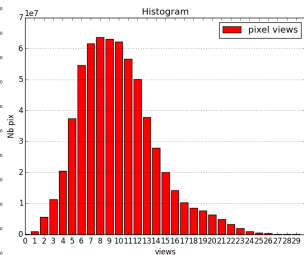
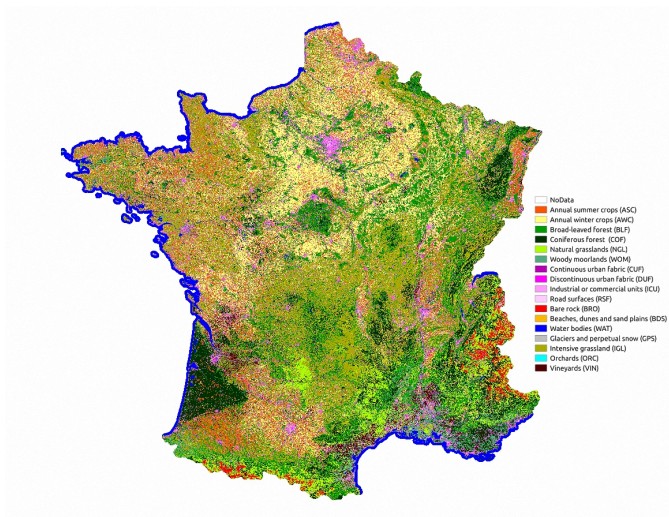


Figure: Histogram of the number of times that every pixel sees the ground taking into account satellite revisit and cloud cover.

Open Access: <http://www.mdpi.com/2072-4292/9/1/95/htm>

Objective

The aim is to be able to cluster the whole France using Sentinel-2 data.



Gaussian modeling

- ▶ $Y_i \in \{1, \dots, G\}$,
- ▶ $\mathcal{L}(\mathbf{X}_i | Y_i = g) = \mathcal{N}(\boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g)$
- ▶ Two kinds of parsimony assumptions on covariance matrices
 - ▶ independence between spectra $\boldsymbol{\Sigma}_{g,k}$ of size $T \times T$, ($T = 17$),
 - ▶ or independences between times, $\boldsymbol{\Sigma}_{g,t}$ of size $K \times K$, ($K = 4$).
- ▶ handle missing values for both models
- ▶ Implementations and tests in a R package.

Gaussian modeling

- ▶ $Y_i \in \{1, \dots, G\}$,
- ▶ $\mathcal{L}(\mathbf{X}_i | Y_i = g) = \mathcal{N}(\boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g)$
- ▶ Two kinds of parsimony assumptions on covariance matrices
 - ▶ independence between spectra $\boldsymbol{\Sigma}_{g,k}$ of size $T \times T$, ($T = 17$),
 - ▶ or independences between times, $\boldsymbol{\Sigma}_{g,t}$ of size $K \times K$, ($K = 4$).
- ▶ handle missing values for both models
- ▶ Implementations and tests in a R package.

Gaussian modeling

- ▶ $Y_i \in \{1, \dots, G\}$,
- ▶ $\mathcal{L}(\mathbf{X}_i | Y_i = g) = \mathcal{N}(\boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g)$
- ▶ Two kinds of parsimony assumptions on covariance matrices
 - ▶ independence between spectra $\boldsymbol{\Sigma}_{g,k}$ of size $T \times T$, ($T = 17$),
 - ▶ or independences between times, $\boldsymbol{\Sigma}_{g,t}$ of size $K \times K$, ($K = 4$).
- ▶ handle missing values for both models
- ▶ Implementations and tests in a R package.

Gaussian modeling

- ▶ $Y_i \in \{1, \dots, G\}$,
- ▶ $\mathcal{L}(\mathbf{X}_i | Y_i = g) = \mathcal{N}(\boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g)$
- ▶ Two kinds of parsimony assumptions on covariance matrices
 - ▶ independence between spectra $\boldsymbol{\Sigma}_{g,k}$ of size $T \times T$, ($T = 17$),
 - ▶ or independences between times, $\boldsymbol{\Sigma}_{g,t}$ of size $K \times K$, ($K = 4$).
- ▶ handle missing values for both models
- ▶ Implementations and tests in a R package.

Method

Missing values formation process

Missing At Random (MAR): Probability for a value to be missing does not depend from its value conditionally to the other observations.

Denote $\mathbf{x}_{ik}^+ = \begin{pmatrix} \mathbf{x}_i^{\text{O}} \\ \mathbf{x}_{ik}^{\text{M}^+} \end{pmatrix}$, $\tilde{\Sigma}_{ik}^+ = \begin{pmatrix} 0_i^{\text{O}} & 0_i^{\text{OM}} \\ 0_i^{\text{MO}} & \tilde{\Sigma}_{ik}^{\text{M}^+} \end{pmatrix}$ with 0 null matrix, and

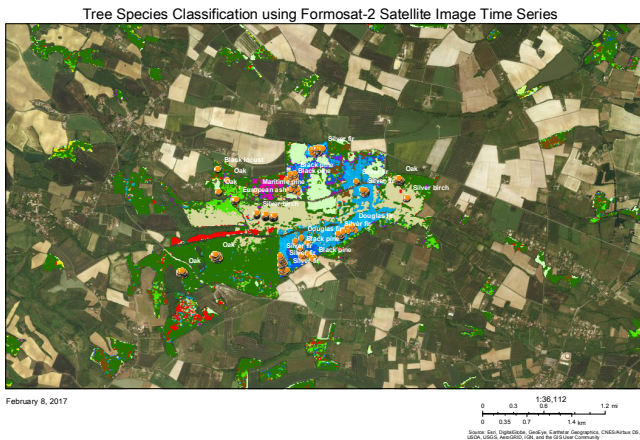
$\tilde{\Sigma}_{ik}^{\text{M}^+} = \Sigma_{ik}^{\text{M}} - \Sigma_{ik}^{\text{MO}} (\Sigma_{ik}^{\text{O}})^{-1} \Sigma_{ik}^{\text{OM}}$. then

$$\Sigma_k^+ = \frac{1}{n_k^+} \sum_{i=1}^n \left[(\mathbf{x}_{ik}^+ - \boldsymbol{\mu}_k^+) (\mathbf{x}_{ik}^+ - \boldsymbol{\mu}_k^+)' + \tilde{\Sigma}_{ik}^+ \right]$$

 $\tilde{\Sigma}_{ik}^{\text{M}^+}$

is correcting the variance due to the imputation by the mean.

Gaussian modeling

Figure: Tree species classification with $G = 13$

Continuous Model

Main assumption

$$Y|Z = k \sim GP(\mu_k, C_k), \quad k = 1, \dots, K \quad (2)$$

where $GP(\mu_k, C_k)$ is a **Gaussian Process** with mean $\mu_k \in L_2(I)$ and with covariance operator $C_k : I \times I \rightarrow \mathbb{R}$.

- ▶ mean functions belongs to a J -dimensional subspace

$$\mu_k(t) = \sum_{j=1}^J \alpha_{kj} \varphi_j(t),$$

- ▶ Covariance function

$$C_k(s, t)(h_k) = \theta_k Q((t - s)/h_k),$$

- ▶ Spectrum are independents.

Continuous Model

Main assumption

$$Y|Z = k \sim GP(\mu_k, C_k), \quad k = 1, \dots, K \quad (2)$$

where $GP(\mu_k, C_k)$ is a **Gaussian Process** with mean $\mu_k \in L_2(I)$ and with covariance operator $C_k : I \times I \rightarrow \mathbb{R}$.

- ▶ mean functions belongs to a J -dimensional subspace

$$\mu_k(t) = \sum_{j=1}^J \alpha_{kj} \varphi_j(t),$$

- ▶ Covariance function

$$C_k(s, t)(h_k) = \theta_k Q((t - s)/h_k),$$

- ▶ Spectrum are independents.

Continuous Model

Main assumption

$$Y|Z = k \sim GP(\mu_k, C_k), \quad k = 1, \dots, K \quad (2)$$

where $GP(\mu_k, C_k)$ is a **Gaussian Process** with mean $\mu_k \in L_2(I)$ and with covariance operator $C_k : I \times I \rightarrow \mathbb{R}$.

- ▶ mean functions belongs to a J -dimensional subspace

$$\mu_k(t) = \sum_{j=1}^J \alpha_{kj} \varphi_j(t),$$

- ▶ Covariance function

$$C_k(s, t)(h_k) = \theta_k Q((t - s)/h_k),$$

- ▶ Spectrum are independents.

Continuous Model

Main assumption

$$Y|Z = k \sim GP(\mu_k, C_k), \quad k = 1, \dots, K \quad (2)$$

where $GP(\mu_k, C_k)$ is a **Gaussian Process** with mean $\mu_k \in L_2(I)$ and with covariance operator $C_k : I \times I \rightarrow \mathbb{R}$.

- ▶ mean functions belongs to a J -dimensional subspace

$$\mu_k(t) = \sum_{j=1}^J \alpha_{kj} \varphi_j(t),$$

- ▶ Covariance function

$$C_k(s, t)(h_k) = \theta_k Q((t - s)/h_k),$$

- ▶ Spectrum are independents.

Estimation of Continuous Model

For each i , let $B_{\ell,j}^i = \varphi_j(t_\ell^i)$, $m_{ki} = B^i \alpha_k$ and

$$\Sigma_{j,j'}^i(h_k) = \theta_k Q((t_j^i - t_{j'}^i)/h_k) =: \theta_k S_{j,j'}^i(h_k),$$

then

$$y_i | Z_i = k \sim \mathcal{N}_{T_i}(m_{ki}, \theta_k S^i(h_k)), \quad k = 1, \dots, K, \quad i = 1, \dots, n$$

we end up with K independent minimization problems:

$$(\hat{\alpha}_k, \hat{h}_k) = \arg \max_{\alpha_k, h_k, \theta_k} \sum_{Z_i=k} \log \det S^i(h_k) + T_i \log \theta_k \\ + \frac{1}{\theta_k} (y_i - B^i \alpha_k)^\top S^i(h_k)^{-1} (y_i - B^i \alpha_k)$$

Results

About 65% well classified.

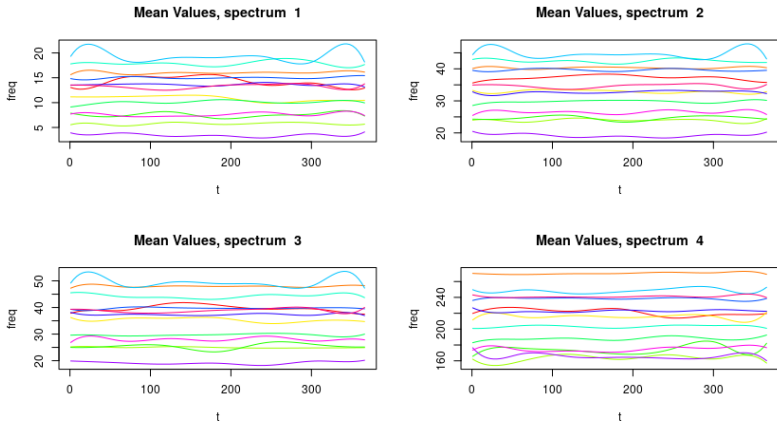


Figure: $G = 13$ spectrum

Mean values

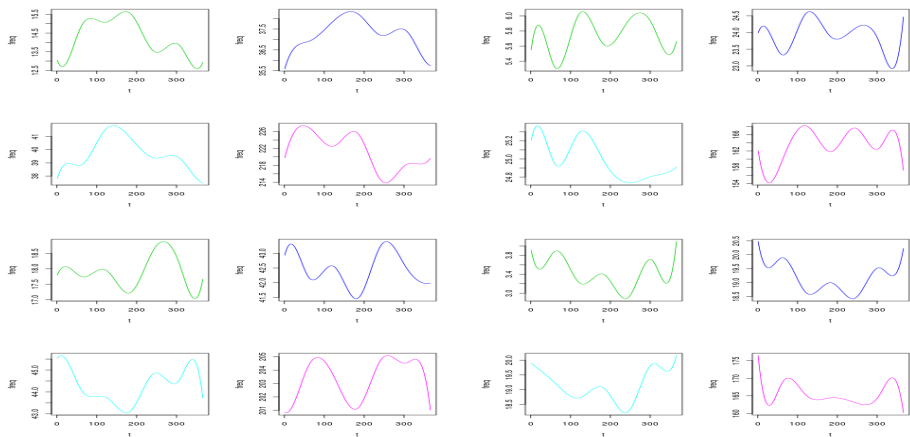


Figure: first, 4th, 7th and 11th classes

Links

- ▶ <https://cran.r-project.org/web/packages/MixAll/>
- ▶ <https://massiccc.lille.inria.fr/>
- ▶ <https://modal.lille.inria.fr/CloHe/>
- ▶ <http://www.mdpi.com/2072-4292/9/1/95/htm>