# Survival analysis with complex covariates: a model-based clustering preprocessing step

Vincent Vandewalle, Christophe Biernacki

# Survival analysis with complex covariates: a model-based clustering preprocessing step

Vincent Vandewalle[1,2,3] & Christophe Biernacki[3,4,5]

IEEE PHM
Dallas
June 19th, 2017

1 Université de Lille 2 Droit et Santé
2 équipe d'accueil EA 2694
3 Inria informatics mathematics
4 cnrs dépasser les frontières
5 Université de Lille 1 Sciences et Technologies

# Overview

# Overview

# Survival analysis framework without covariates

- In survival analysis the goal is often to predict/explain $T$ the elapsed time before the occurence of some phenomenon (failure, death, ...).
- Most of the models are interested in the **survival function**

$$S(t) = \mathrm{P}(T > t)$$

- Or equivalently in the **hazard function**

$$\lambda(t) = \lim_{h \to 0} \frac{1}{h} P(t + h \geq T > t | T > t) = \frac{f(t)}{S(t)}$$

  with $f(t)$ the probability density function of the failure time.
- Let $\Lambda(t) = \int_0^t \lambda(u)du$ the **cumulative hazard function**, the survival function is thus

$$S(t) = \exp(-\Lambda(t))$$

- The estimation of $\lambda(t)$ can be performed using:
  - parametric models (exponential, Weibull, Gamma, ...)
  - non parametric model (Kaplan-Meier, ...)

# Survival analysis with standard covariates

## Covariates $\mathbf{x}$ available

- Thus one is interested in $S(t|\mathbf{x})$ the **conditional survival function** $S(t|\mathbf{x}) = P(T > t|\mathbf{x})$.
- Or alternatively in $\lambda(t|\mathbf{x})$ the **conditional hazard function**.

## Cox Proportional Hazard model

Let $\mathbf{x} \in \mathbb{R}^d$, $\boldsymbol{\beta} \in \mathbb{R}^d$ the conditional hazard function is

$$\lambda(t|\mathbf{x}) = \lambda_0(t) \exp(\boldsymbol{\beta}^T \mathbf{x}).$$

- $\boldsymbol{\beta}$ estimated without assumptions on $\lambda_0(t)$ (Cox partial likelihood)
- $\lambda_0(t)$ estimated in second time, in a parametric or non parametric way

## Cox model limitations

- Cannot handle missing data
- Cannot handle non-standard covariates

# Non-standard covariates (1/2)

Many variables collected:

- Heterogeneous data (of various natures)
- Incertain data (by interval)
- Missing data

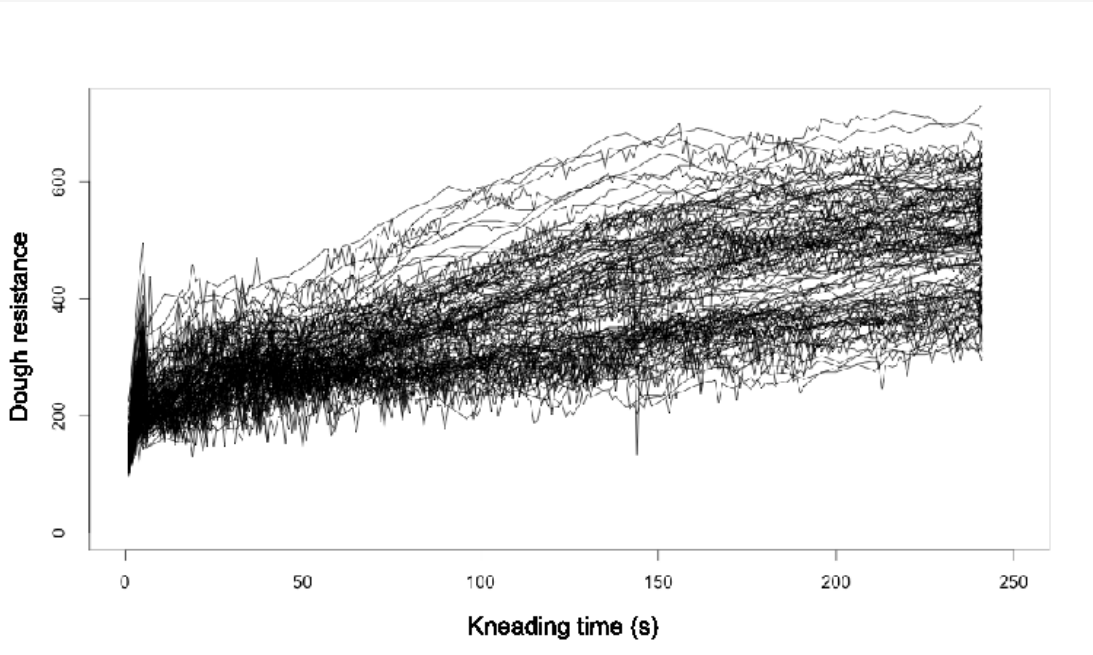
**Heterogeneous, missing, incertain**

| Units $\mathbf{x}^O \in \mathcal{X}$ | | | |
|---|---|---|---|
| ? | 0.5 | ? | 5 |
| 0.3 | 0.1 | green | 3 |
| 0.3 | 0.6 | {red, green} | 3 |
| 0.9 | [0.25 0.45] | red | ? |
| ↓ | ↓ | ↓ | ↓ |
| continuous | continuous | categorical | integer |

# Non-standard covariates (2/2)

And also:

- Rank data
- Directional data
- Ordinal data
- Functional data
- Network data
- ...



# From non-standard to standard covariates

## Idea

Define a mapping $r$ from $\mathcal{X}$ (non-standard) to $\{1, \ldots, K\}$ (standard)

$$\begin{array}{rccc} r: & \mathcal{X} & \longrightarrow & \{1, \ldots, K\} \\ & \mathbf{x} & \longmapsto & r(\mathbf{x}). \end{array}$$

## Application to survival analysis

- Similar $\mathbf{x}$ should have similar survival functions $S(t|\mathbf{x})$.
- $r$ should group together similar values of $\mathbf{x}$.
- Apply standard survival analysis with $r(\mathbf{x})$ as covariate: $S(t|r(\mathbf{x}))$.

## How to define $r$?

- By using **clustering methods** on complex data.
- Summarize the heterogeneity of $\mathbf{x}$ distribution by $K$ homogenous areas.
- For $K$ large enough: $S(t|r(\mathbf{x})) \simeq S(t|\mathbf{x})$.

# Clustering of complex data

## Data

$n$ individuals: $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_n) = (\mathbf{x}^O, \mathbf{x}^M)$ belonging to a space $\mathcal{X}$

- Observed data $\boldsymbol{x}^O$
- Missing data $\boldsymbol{x}^M$

## Aim

Estimation of the partition $\mathbf{z}$ and the number of clusters $K$
Partition in $K$ clusters $G_1, \ldots, G_K$: $\mathbf{z} = (\mathbf{z}_1, \ldots, \mathbf{z}_n)$, $\mathbf{z}_i = (z_{i1}, \ldots, z_{iK})'$

$$\omega_i \in G_k \quad \Leftrightarrow \quad z_{ih} = \mathbb{I}_{\{h=k\}}$$

## Mixed, missing, uncertain

| Individuals $\mathbf{x}^O$ | | | | Partition $\mathbf{z}$ | | | $\Leftrightarrow$ | Group |
|---|---|---|---|---|---|---|---|---|
| ? | 0.5 | red | 5 | ? | ? | ? | $\Leftrightarrow$ | ??? |
| 0.3 | 0.1 | green | 3 | ? | ? | ? | $\Leftrightarrow$ | ??? |
| 0.3 | 0.6 | {red,green} | 3 | ? | ? | ? | $\Leftrightarrow$ | ??? |
| 0.9 | [0.25 0.45] | red | ? | ? | ? | ? | $\Leftrightarrow$ | ??? |
| ↓ | ↓ | ↓ | ↓ | | | | | |
| continuous | continuous | categorical | integer | | | | | |

# Including complex covariates in survival analysis through mixture models

## Model based clustering allow to cluster the data and to compute:

$$P(k|\mathbf{x}) = P(\omega \in G_k|\mathbf{x})$$

## Cluster-specific survival analysis

$$P(T > t|\mathbf{x}) = \sum_{k=1}^{K} P(T > t|\mathbf{x}, k)P(k|\mathbf{x}) = \sum_{k=1}^{K} \underbrace{P(T > t|\cancel{\mathbf{x}}, k)}_{\text{key assumption}} P(k|\mathbf{x}) = \sum_{k=1}^{K} S_k(t)P(k|\mathbf{x})$$

with $S_k(t)$ the survival function specific of cluster $k$.

## Rationale

The clusters found, based on the heterogeneity of $\mathbf{x}$, explain the variability of the survival times $T$.

## Strategy

- Cluster complex data, $\mathbf{x}$, with model based clustering
- Learn each cluster specific survival function $S_k(t)$ based on the data classified in cluster $k$

# Overview

# Supervised classification (1/3)

- Data: learning dataset $\mathcal{D} = (\mathbf{x}^O, \mathbf{z})$
  - $n$ individuals: $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_n) = (\mathbf{x}^O, \mathbf{x}^M)$ belonging to a space $\mathcal{X}$
  - Observed data $\boldsymbol{x}^O$
  - Missing data $\boldsymbol{x}^M$
  - Partition in $K$ groups $G_1, \ldots, G_K$: $\mathbf{z} = (\mathbf{z}_1, \ldots, \mathbf{z}_n)$, $\mathbf{z}_i = (z_{i1}, \ldots, z_{iK})'$

$$\mathbf{x}_i \in G_k \quad \Leftrightarrow \quad z_{ih} = \mathbb{I}_{\{h=k\}}$$

- Aim: estimation of an allocation rule $r$ from $\mathcal{D}$

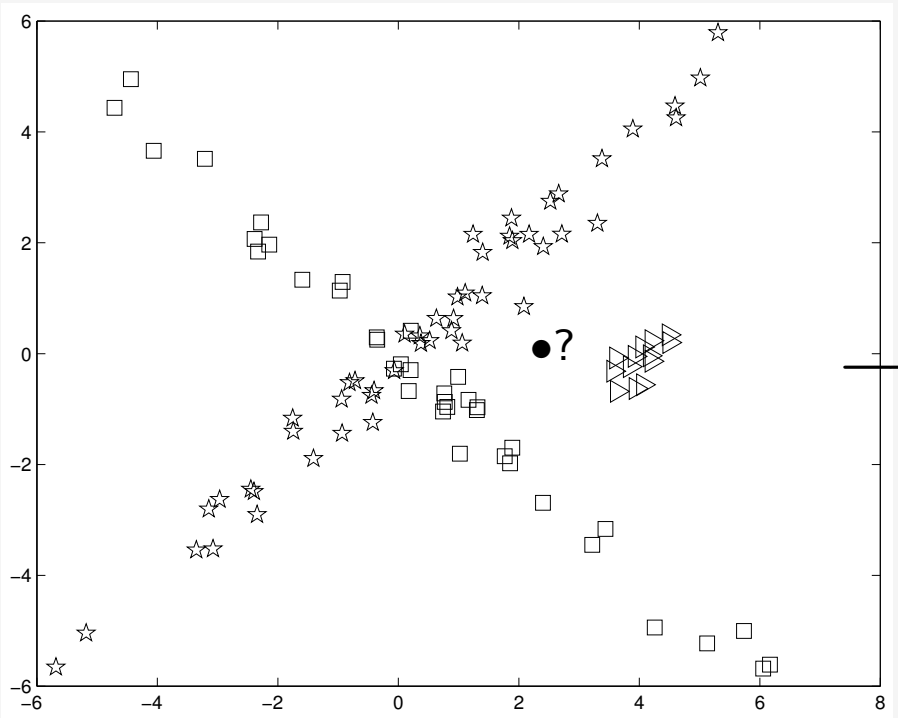
$$\begin{array}{rccc} r: & \mathcal{X} & \longrightarrow & \{1, \ldots, K\} \\ & \mathbf{x}^O_{n+1} & \longmapsto & r(\mathbf{x}^O_{n+1}). \end{array}$$

# Supervised classification (2/3)

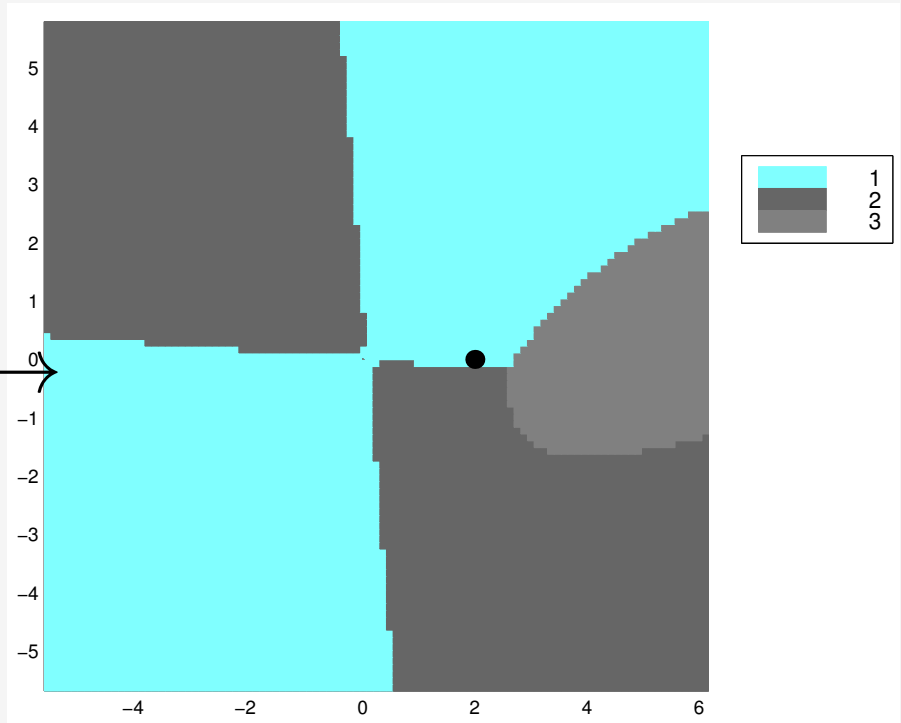
Mixed, missing, uncertain

| Individuals $\mathbf{x}^O$ | | | | Partition $\mathbf{z}$ | | | | Group |
|---|---|---|---|---|---|---|---|---|
| ? | 0.5 | red | 5 | 0 | 1 | 0 | $\Leftrightarrow$ | $G_2$ |
| 0.3 | 0.1 | green | 3 | 1 | 0 | 0 | $\Leftrightarrow$ | $G_1$ |
| 0.3 | 0.6 | {red,green} | 3 | 1 | 0 | 0 | $\Leftrightarrow$ | $G_1$ |
| 0.9 | [0.25 0.45] | red | ? | 0 | 0 | 1 | $\Leftrightarrow$ | $G_3$ |
| ↓ | ↓ | ↓ | ↓ | | | | | |
| continuous | continuous | categorical | integer | | | | | |

# Supervised classification (3/3)




$(\mathbf{x}^O, \mathbf{z}^O)$ and $\mathbf{x}^O_{n+1}$




$\hat{r}$ and $\hat{\mathbf{z}}^0_{n+1}$

# Semi-supervised classification (1/3)

- Data: learning dataset $\mathcal{D} = (\mathbf{x}^O, \mathbf{z}^O)$
  - $n$ individuals: $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_n) = (\mathbf{x}^O, \mathbf{x}^M)$ belonging to a space $\mathcal{X}$
  - Observed data $\boldsymbol{x}^O$
  - Missing data $\boldsymbol{x}^M$
  - Partition: $\mathbf{z} = (\boldsymbol{z}_1, \ldots, \mathbf{z}_n) = (\mathbf{z}^O, \mathbf{z}^M)$
  - Observed partition $\boldsymbol{z}^O$
  - Missing partition $\boldsymbol{z}^M$
- Aim: estimation of an allocation rule $r$ from $\mathcal{D}$

$$\begin{array}{rccc} r: & \mathcal{X} & \longrightarrow & \{1, \ldots, K\} \\ & \mathbf{x}^O_{n+1} & \longmapsto & r(\mathbf{x}^O_{n+1}). \end{array}$$

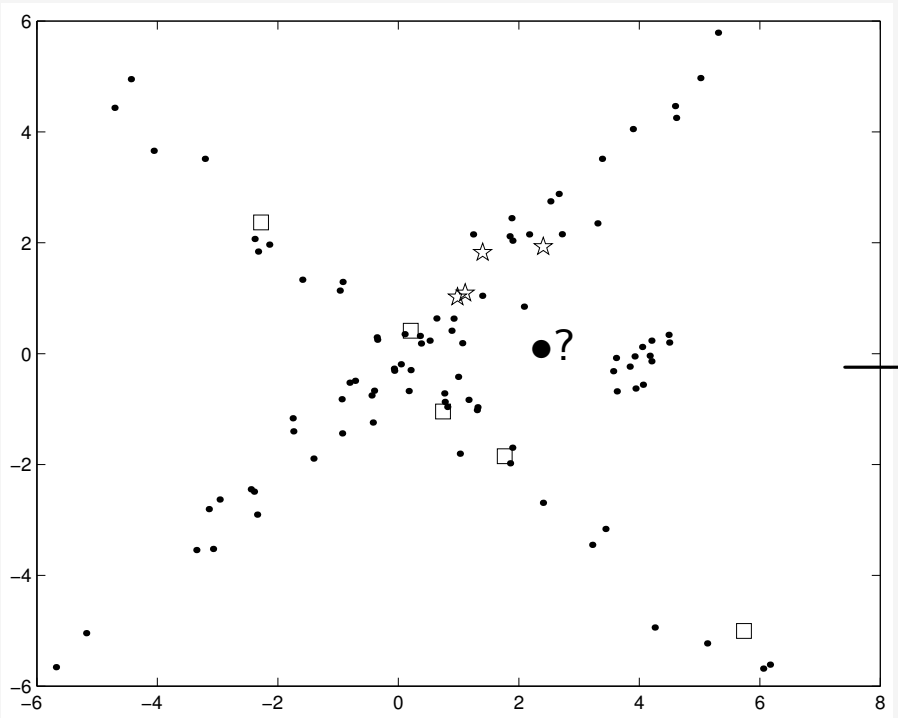
- Idea: $\mathbf{x}$ is cheaper than $\mathbf{z}$ so $n \gg n^l$

# Semi-supervised classification (2/3)

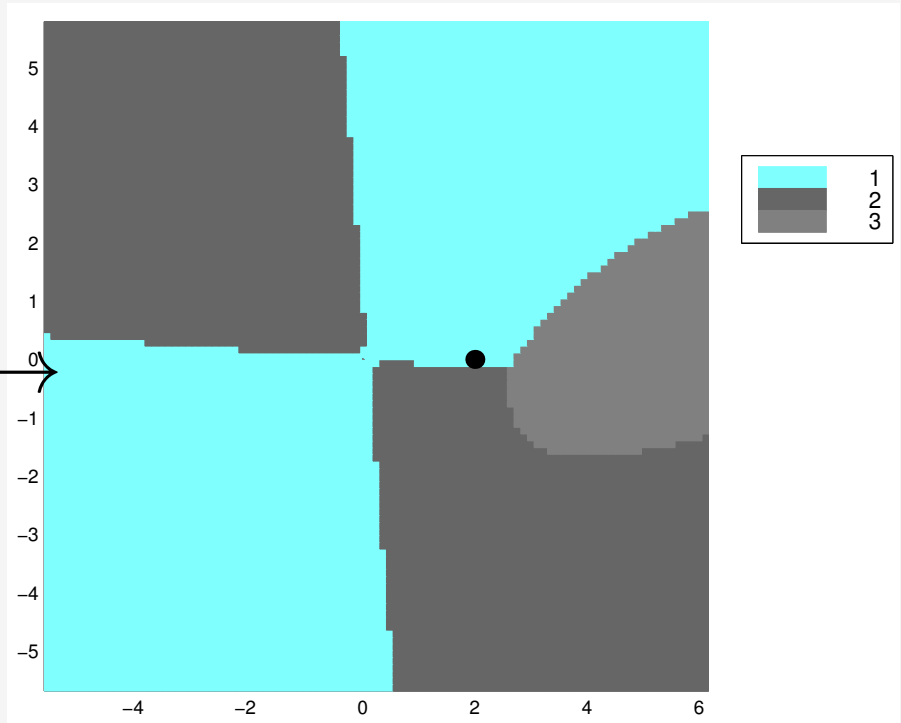
Mixed, missing, uncertain

| Individuals $\mathbf{x}^O$ | | | | Partition $\mathbf{z}^O$ | | | $\Leftrightarrow$ | Group |
|---|---|---|---|---|---|---|---|---|
| ? | 0.5 | red | 5 | 0 | **?** | **?** | $\Leftrightarrow$ | $G_2$ or $G_3$ |
| 0.3 | 0.1 | green | 3 | 1 | 0 | 0 | $\Leftrightarrow$ | $G_1$ |
| 0.3 | 0.6 | {red,green} | 3 | **?** | **?** | **?** | $\Leftrightarrow$ | ??? |
| 0.9 | [0.25 0.45] | red | ? | 0 | 0 | 1 | $\Leftrightarrow$ | $G_3$ |
| ↓ | ↓ | ↓ | ↓ | | | | | |
| continuous | continuous | categorical | integer | | | | | |

# Semi-supervised classification (3/3)




$(\mathbf{x}^O, \mathbf{z}^O)$ and $\mathbf{x}^O_{n+1}$




$\hat{r}$ and $\hat{\mathbf{z}}^O_{n+1}$

# Unsupervised classification (1/3)

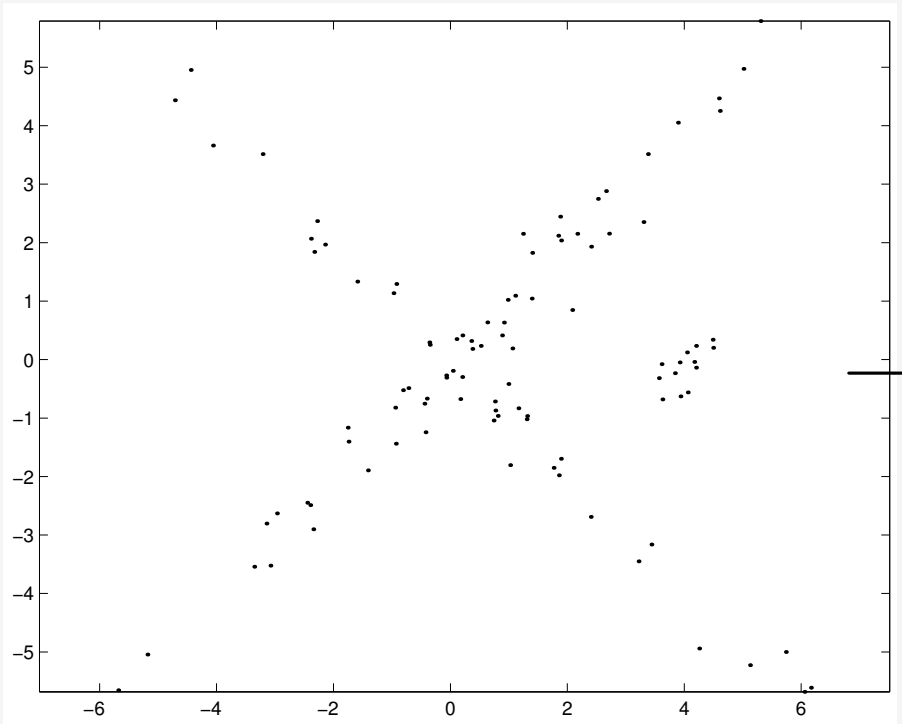
- Data: learning dataset $\mathcal{D} = \mathbf{x}^O$, so $\mathbf{z}^O = \emptyset$
- Aim: estimation of the partition $\mathbf{z}$ and the number of groups $K$
- Also known as: clustering

# Unsupervised classification (2/3)

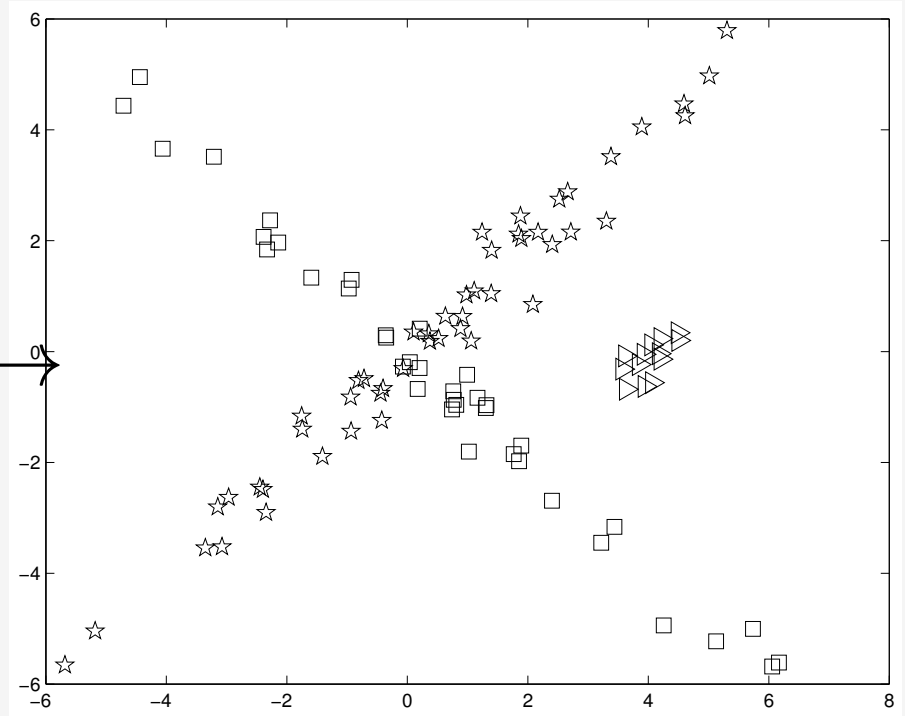
Mixed, missing, uncertain

| Individuals $\mathbf{x}^O$ | | | | Partition $\mathbf{z}^O$ | | | $\Leftrightarrow$ | Group |
|---|---|---|---|---|---|---|---|---|
| ? | 0.5 | red | 5 | **?** | **?** | **?** | $\Leftrightarrow$ | ??? |
| 0.3 | 0.1 | green | 3 | **?** | **?** | **?** | $\Leftrightarrow$ | ??? |
| 0.3 | 0.6 | {red,green} | 3 | **?** | **?** | **?** | $\Leftrightarrow$ | ??? |
| 0.9 | [0.25 0.45] | red | ? | **?** | **?** | **?** | $\Leftrightarrow$ | ??? |
| ↓ | ↓ | ↓ | ↓ | | | | | |
| continuous | continuous | categorical | integer | | | | | |

# Unsupervised classification (3/3)




$\mathbf{x}^O$




$(\mathbf{x}^O, \hat{\mathbf{z}})$

# Traditional solutions (1/3)

Two main frameworks

- Generative models
  - Model $p(\boldsymbol{x}, \boldsymbol{z})$
  - Thus direct model for $p(\boldsymbol{x}) = \sum_{\boldsymbol{z}} p(\boldsymbol{x}, \boldsymbol{z})$
  - Easy to take into account some missing $\boldsymbol{z}$ and $\boldsymbol{x}$
- Predictive models
  - Model $p(\boldsymbol{z}|\boldsymbol{x})$ or sometimes $\mathbf{1}_{\{p(\boldsymbol{z}|\boldsymbol{x})>1/2\}}$ or also ranking on $p(\boldsymbol{z}|\boldsymbol{x})$
  - Avoid asumptions on $p(\boldsymbol{x})$, thus avoids associated error model
  - difficult to take into account some missing $\boldsymbol{z}$ and $\boldsymbol{x}$

# Traditional solutions (2/3)

No mixed, missing or uncertain data:

- Supervised classification[1]
  - Generative models: linear/quadratic discriminant analysis
  - Predictive models: logistic regression, support vector machines (SVM), $k$ nearest neighbourhood, classification trees...
- Semi-supervised classification[2]
  - Generative models: mixture models
  - Predictive models: low density separation (transductive SVM), graph-based methods...
- Unsupervised classification[3]
  - Generative models: $k$-means like criteria, hierarchical clustering, mixture models
  - Predictive models: -

[1] Govaert *et al.*, Data Analysis, Chap.6, 2009

[2] Chapelle *et al.*, Semi-supervised learning, 2006

[3] Govaert *et al.*, Data Analysis, Chap.7-9, 2009

# Traditional solutions (3/3)

But more complex with mixed, missing or uncertain data...

- Missing/uncertain data: multiple imputation is possible but it should ideally take into account the classification purpose at hand
- Mixed data: some heuristic methods with recoding

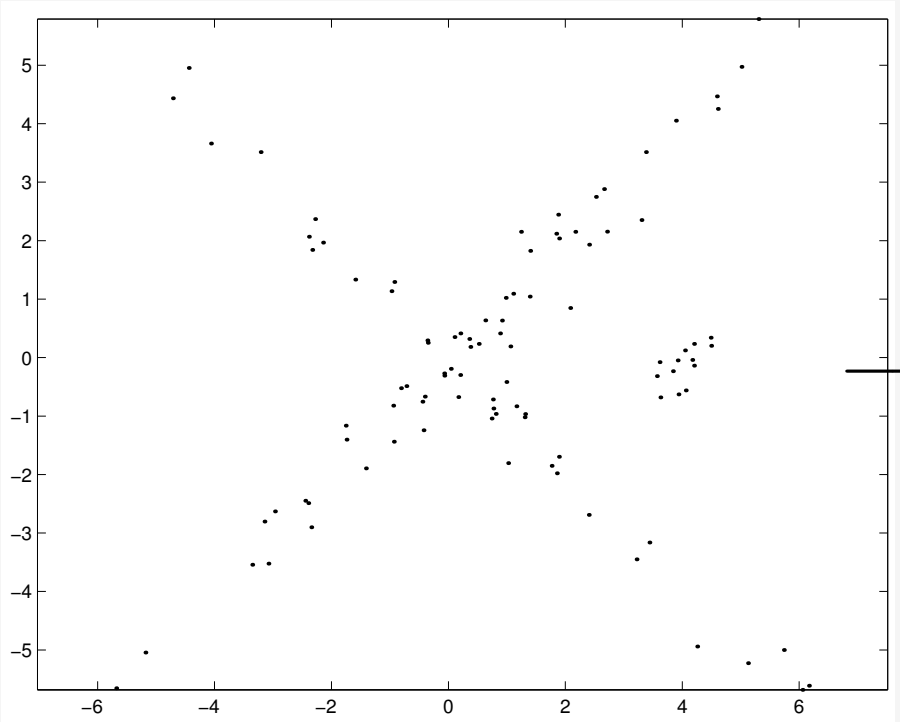
How to marry the classification aim with mixed, missing or uncertain data?

# Overview

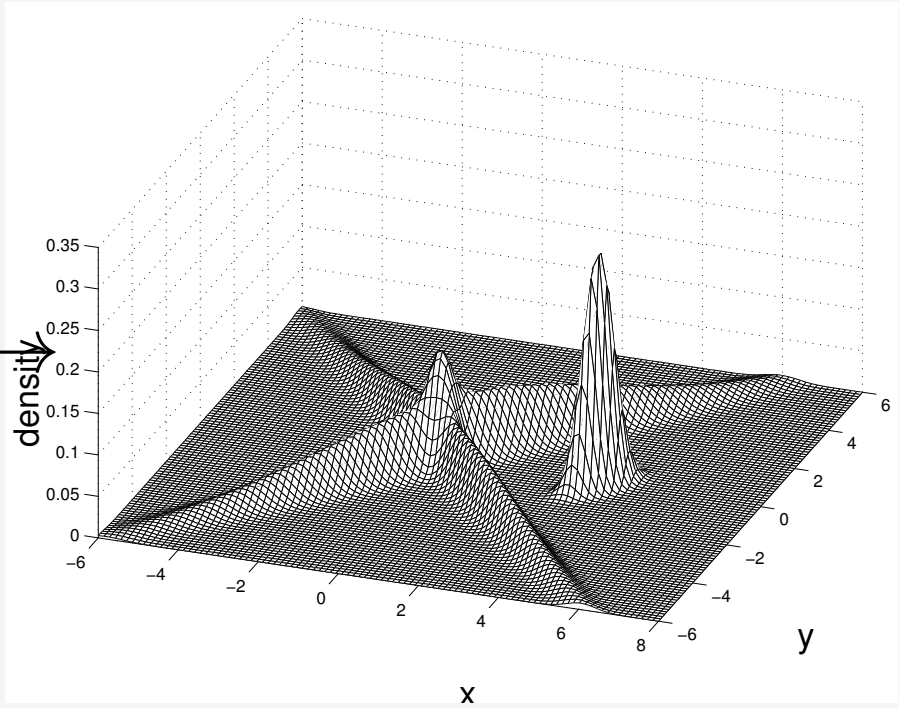
# Density estimation (1/2)

- Data: learning dataset $\mathcal{D} = \mathbf{x}^O$, so $\boldsymbol{z}^O = \emptyset$
- Aim: estimation of the distribution $\mathrm{p}(\mathbf{x})$
- Extension easy to: $\mathcal{D} = (\mathbf{x}^O, \mathbf{z}^O)$ with $\boldsymbol{z}^O \neq \emptyset$
- Useful for: data imputation and multi-purpose classification!

# Density estimation (2/2)




$\mathbf{x}^O$




$\hat{p}(\mathbf{x})$

# The mixture model answer in {∅,semi,un} classification

- Rigorous definition of a group:

$$\mathbf{x}_1 \in G_k \quad \Leftrightarrow \quad \mathbf{x}_1 \text{ is a realization of } \mathbf{X}_1 \sim \mathrm{p}_k(\mathbf{x}_1)$$

- Generative formulation:

$$\begin{aligned} \mathbf{Z}_1 &\sim \mathrm{Mult}_K(1, \underbrace{\pi_1, \ldots, \pi_K}_{\boldsymbol{\pi}}) \\ \mathbf{X}_{1|Z_{1k}=1} &\sim \mathrm{p}_k(\mathbf{x}_1) \end{aligned}$$

- Joint and marginal (or mixture) distributions:

$$\begin{aligned} (\mathbf{X}_1, \mathbf{Z}_1) &\sim \prod_{k=1}^{K} [\pi_k \mathrm{p}_k(\mathbf{x}_1)]^{z_{1k}} \\ \mathbf{X}_1 &\sim \mathrm{p}(\mathbf{x}_1) = \sum_{k=1}^{K} \pi_k \mathrm{p}_k(\mathbf{x}_1) \end{aligned}$$

- Maximum *a posteriori* (MAP): with $t_k(\mathbf{x}_1^O) = \mathrm{p}(Z_{1k} = 1|\mathbf{x}_1^O) = \frac{\pi_k \mathrm{p}_k(\mathbf{x}_1^O)}{\mathrm{p}(\mathbf{x}_1^O)}$

$$r(\mathbf{x}_1) = \arg \max_{k=\{1,\ldots,K\}} t_k(\mathbf{x}_1^O)$$

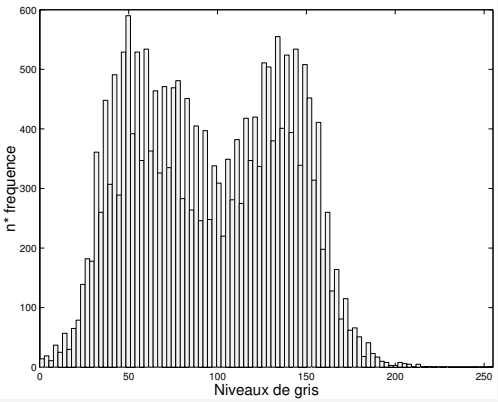
# The mixture model answer for imputation

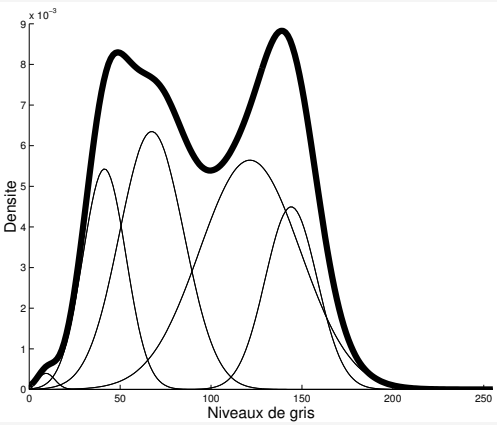
Straightforward also

$$\hat{\boldsymbol{x}}^M = \arg\max_{\boldsymbol{x}^M} \mathrm{p}(\boldsymbol{x}^M | \boldsymbol{x}^O)$$

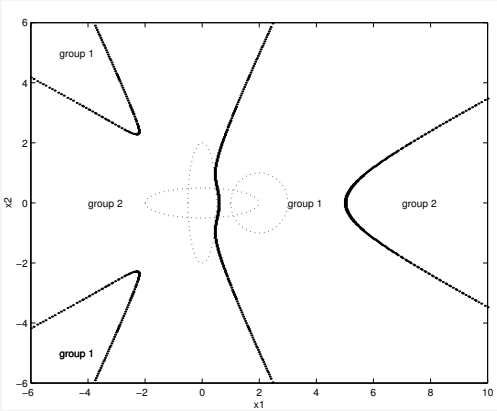
# The mixture model answer in density estimation

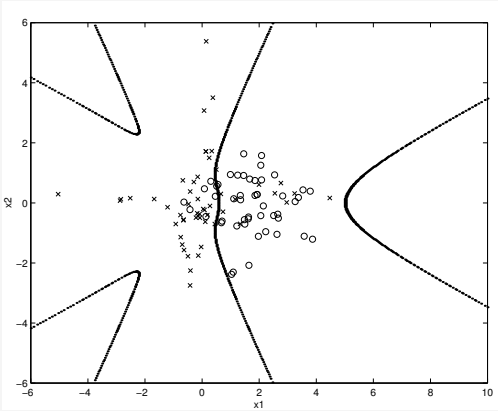
- Mixture models: extremely flexible family of distributions




- Mixture of mixture models: flexibility for groups also



# Parametric mixture model

- Parametric assumption:

$$p_k(\mathbf{x}_1) = p(\mathbf{x}_1; \boldsymbol{\alpha}_k)$$

thus

$$p(\mathbf{x}_1) = p(\mathbf{x}_1; \boldsymbol{\theta}) = \sum_{k=1}^{K} \pi_k p(\mathbf{x}_1; \boldsymbol{\alpha}_k)$$

- Mixture parameter:

$$\boldsymbol{\theta} = (\boldsymbol{\pi}, \boldsymbol{\alpha}) \text{ with } \boldsymbol{\alpha} = (\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_K)$$

- Model: it includes both the family $p(\cdot; \boldsymbol{\alpha}_k)$ and the number of groups $K$

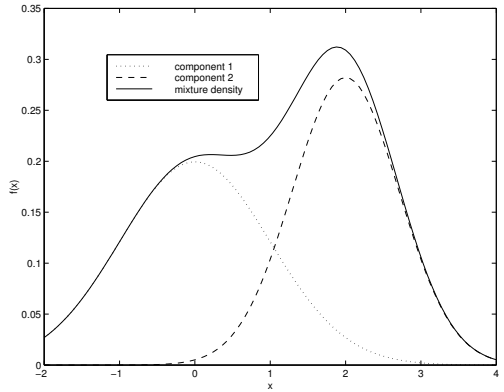
$$\mathbf{m} = \{p(\mathbf{x}_1; \boldsymbol{\theta}) : \boldsymbol{\theta} \in \Theta\}$$

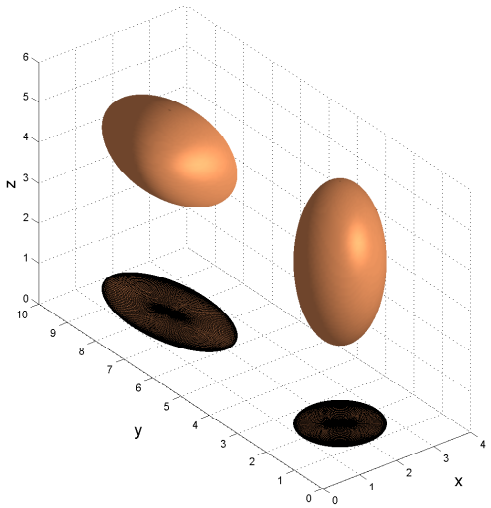
The number of free *continuous* parameters is given by

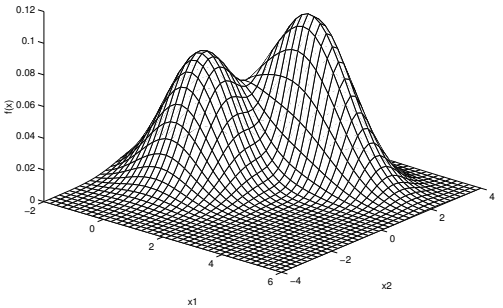
$$\nu = \dim(\Theta)$$

# Continuous: Gaussian mixture model

$$\mathsf{p}(\cdot; \boldsymbol{\alpha}_k^{cont}) = \mathsf{N}_d(\boldsymbol{\mu}_k, \underbrace{\boldsymbol{\Sigma}_k}_{\text{diagonal}})$$



# Categorical: latent class model

- Categorical variables: $d$ variables with $m_j$ modalities each, $\mathbf{x}_i^j \in \{0,1\}^{m_j}$ and

$$\mathbf{x}_i^{jh} = 1 \quad \Leftrightarrow \quad \text{variable } j \text{ of } \mathbf{x}_i \text{ takes modality } h$$

- Intra conditional independence:

$$\mathrm{p}(\mathbf{x}_i^{cat}; \boldsymbol{\alpha}_k^{cat}) = \prod_{j=1}^{d} \prod_{h=1}^{m_j} (\alpha_k^{jh})^{x_i^{jh}}$$

and

$$\alpha_k^{jh} = \mathrm{p}(X_i^{jh} = 1 | Z_{ik} = 1)$$

with $\boldsymbol{\alpha}_k = (\alpha_k^{jh}; j = 1, \ldots, d; h = 1, \ldots, m_j)$

# Integer: Poisson mixture model

- Integer variables: $d$ variables $\mathbf{x}_i^j \in \mathbb{N}$
- Intra conditional independence:

$$p(\mathbf{x}_i^{int}; \boldsymbol{\alpha}_k^{int}) = \prod_{j=1}^{d} \frac{(\alpha_k^j)^{x_i^j}}{\alpha_k^j!} e^{-\alpha_k^j}$$

# Mixed data: conditional independence everywhere

The aim is to combine continuous, categorical and integer data

$$\boldsymbol{x}_1 = (\boldsymbol{x}_1^{cont}, \boldsymbol{x}_1^{cat}, \boldsymbol{x}_1^{int})$$

The proposed solution is to mix all types by inter conditional independence

$$\mathrm{p}(\mathbf{x}_1; \boldsymbol{\alpha}_k) = \mathrm{p}(\mathbf{x}_1^{cont}; \boldsymbol{\alpha}_k^{cont}) \times \mathrm{p}(\mathbf{x}_1^{cat}; \boldsymbol{\alpha}_k^{cat}) \times \mathrm{p}(\mathbf{x}_1^{int}; \boldsymbol{\alpha}_k^{int})$$

In addition, for symmetry between types, intra conditional independence for each type

Also possible to incorporate any type of data by defining its univariate probability distribution $\mathrm{p}(\mathbf{x}_1^{type}; \boldsymbol{\alpha}_k^{type})$.

# Overview

# Sampling assumptions

- True distribution:

$$\mathcal{D} \sim \mathrm{p}(\mathcal{D})$$

- Model distribution:

$$(\mathbf{x}_i, \mathbf{z}_i) \overset{i.i.d.}{\sim} \mathrm{p}(\mathbf{x}_1, \mathbf{z}_1; \boldsymbol{\theta})$$

- Gap between both, but flexibiliy:

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta} \in \Theta} \mathrm{KL}(\mathrm{p}, \mathrm{p}_{\boldsymbol{\theta}})$$

where

$$\mathrm{KL}(\mathrm{p}, \mathrm{p}_{\boldsymbol{\theta}}) = \mathrm{E}_{\mathcal{D}'}[\ln \mathrm{p}(\mathcal{D}') - \ln p(\mathcal{D}'; \boldsymbol{\theta})]$$

# *Observed*-data log-likelihood estimation of $\boldsymbol{\theta}$

- Principle: MLE

$$\hat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}\in\Theta} \ell(\boldsymbol{\theta}; \mathcal{D})$$

with

$$\ell(\boldsymbol{\theta}; \mathcal{D}) = \ln \mathrm{p}(\mathcal{D}; \boldsymbol{\theta}) = \ln \int_{\boldsymbol{x}^M} \sum_{\boldsymbol{z}^M} \mathrm{p}(\boldsymbol{x}, \mathbf{z}; \boldsymbol{\theta}) d\boldsymbol{x}^M$$

- Consistency: we have

$$\hat{\boldsymbol{\theta}} \xrightarrow{a.s.} \boldsymbol{\theta}^*$$

- Algorithm: SEM

# SEM algorithm

- Initialisation: $\boldsymbol{\theta}^{(0)}$
- Iteration nb $q$:
  - E-step: compute conditional probabilities $\mathrm{p}(\boldsymbol{x}^M, \boldsymbol{z}^M | \mathcal{D}; \boldsymbol{\theta}^{(q)})$
  - S-step: draw $(\boldsymbol{x}^{M(q)}, \boldsymbol{z}^{M(q)})$ from $\mathrm{p}(\boldsymbol{x}^M, \boldsymbol{z}^M | \mathcal{D}; \boldsymbol{\theta}^{(q)})$
  - M-step: maximize $\boldsymbol{\theta}^{(q+1)} = \arg\max_{\boldsymbol{\theta}} \ln \mathrm{p}(\mathbf{x}^O, \boldsymbol{z}^O, \boldsymbol{x}^{M(q)}, \boldsymbol{z}^{M(q)}; \boldsymbol{\theta})$
- Stopping rule: iteration number

## Properties

- simplicity because of conditional independence
- classical M steps
- avoids local maxima
- the mean of the sequence $(\boldsymbol{\theta}^{(q)})$ approximates $\hat{\boldsymbol{\theta}}$
- the variance of the sequence $(\boldsymbol{\theta}^{(q)})$ gives confidence intervals

# SE algorithm

A SE algorithm estimates then $(\boldsymbol{x}^M, \boldsymbol{z}^M)$

- Iteration nb $q$:
  - E-step: compute conditional probabilities $\mathrm{p}(\boldsymbol{x}^M, \boldsymbol{z}^M | \mathcal{D}; \hat{\boldsymbol{\theta}})$
  - S-step: draw $(\boldsymbol{x}^{M(q)}, \boldsymbol{z}^{M(q)})$ from $\mathrm{p}(\boldsymbol{x}^M, \boldsymbol{z}^M | \mathcal{D}; \hat{\boldsymbol{\theta}})$
- Stopping rule: iteration number

## Properties

- simplicity because of conditional independence
- the mean/mode of the sequence $(\boldsymbol{x}^{M(q)}, \boldsymbol{z}^{M(q)})$ estimates $(\boldsymbol{x}^M, \boldsymbol{z}^M)$
- confidence intervals are also derived

# Selection on the number of clusters $K$ in the unsupervised setting

Model based clustering allows to use standard statistical tools to perform the selection of the number of cluster $K$

- BIC criterion to find the "true" number of clusters:

$$\text{BIC}(K) = \ell(\hat{\boldsymbol{\theta}}_K; \mathbf{x}^O) - \frac{\nu_K}{2} \log n.$$

- ICL criterion to find "well separated" clusters:

$$\text{ICL}(K) = \ell(\hat{\boldsymbol{\theta}}_K; \mathbf{x}^O, \hat{\mathbf{z}}) - \frac{\nu_K}{2} \log n.$$

# Overview

# Software

## A large variety of clustering packages on the CRAN

https://cran.r-project.org/web/views/Cluster.html

## The MixtComp software

Can deal with:

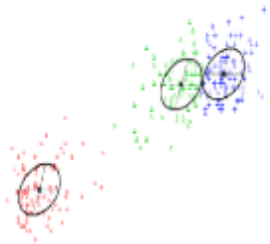
- Mixture models for heterogeneous variables
- Large variety of variable types: continuous, categorical, rank, integer, ordinal, functional ...
- Can deal with missing data, data by interval and partially known categorical varibles
- Main assumption: class conditional independence assumption

## MixtComp available on the MASSICCC platform

https://massiccc.lille.inria.fr : a web platerform to perform clustering.

# Focus on the MASSICCC platform (1/4)

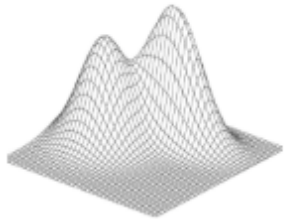
## Some of our Algorithms



### MixMod

Mixmod is a well-established software package for fitting a mixture model of multivariate Gaussian or multinomial probability distribution functions to a given data set. Cluster analysis will partition observations into groups ("clusters") while classification analysis will design a decision function from a learning data set to assign new data to groups a priori known.

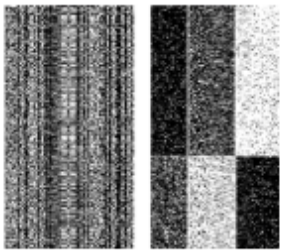
Read more about Mixmod



### MixtComp

MixtComp takes mixture model analysis one step further and deals with mixed, missing or uncertain data which are common in today's data sets.

Read more about MixtComp



### BlockCluster

BlockCluster can estimate the parameters of co-clustering models for binary, contingency and continuous data. Simply put, when considering a set of data as rows and columns, BlockCluster will make simultaneous permutations of rows and columns in order to organise the data into homogenous blocks.

Read more about BlockCluster

# Focus on the MASSICCC platform (2/4)

## How MASSICCC Platform works

MASSICCC Platform lets you upload your data and select from a range of analysis tools to extract meaningful information. No need to install any software or configure some tools. You'll see results straight away.

### Upload your data securely

Upload all or part of your data to Massiccc Platform and run multiple algorithms on them. You will have full control on the data you upload and only you will be able to access them.

### Focus on the data

Minimum configuration is required to use our algorithms since we'll select the most sensible options by default based on the data you provide. No scientific background is required to start working and get results. Advanced configuration options are available if you need specific functions.

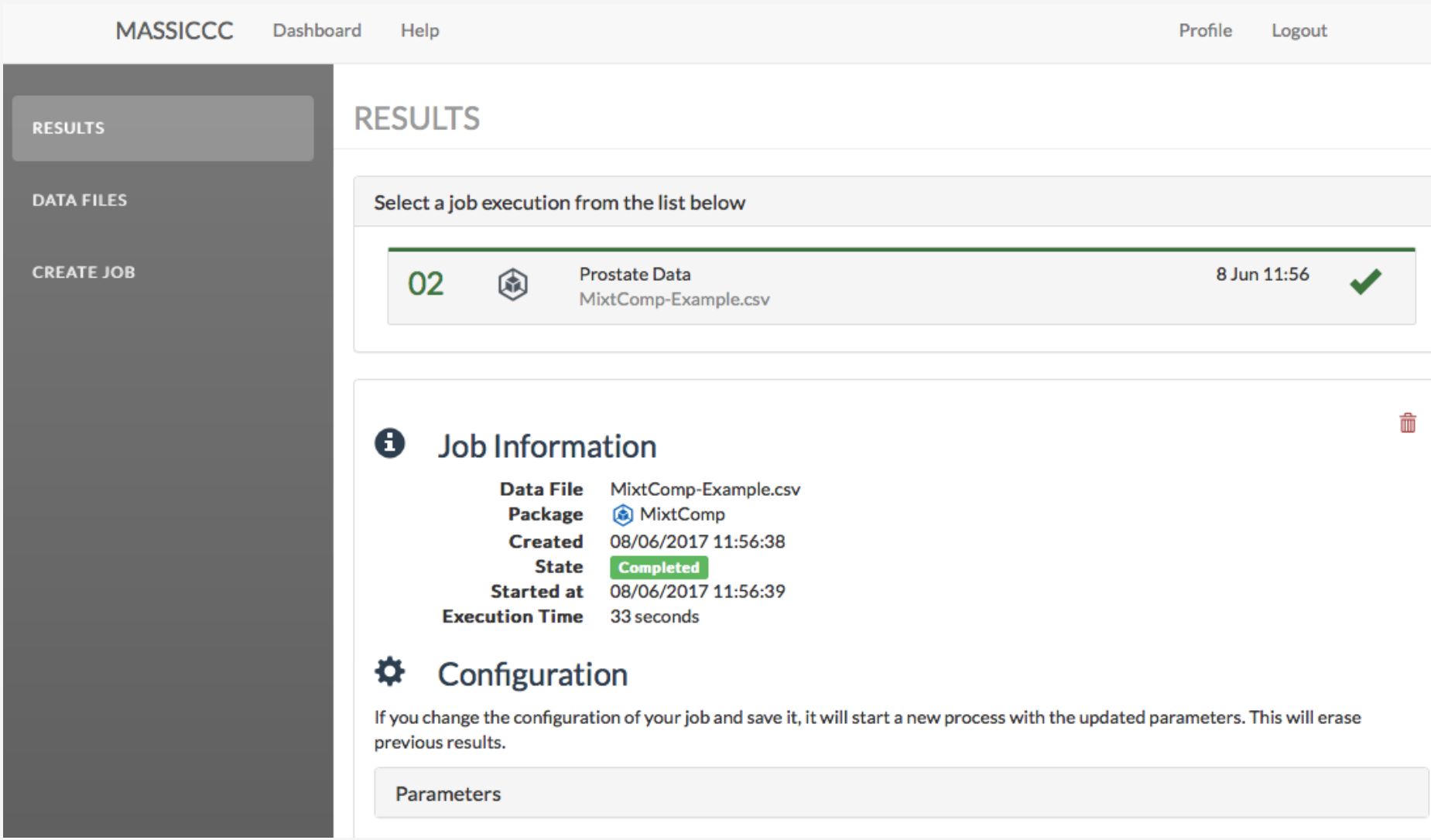
### Choose the best tool for you

Inria is a leading computer science research center and its researchers are working on the most advanced algorithms in Data Science. Massiccc gives you the opportunity to choose from various algorithms that will let you extract and understand the most out of your data.

### Benefit from High Performance Computers

All of the data analysis tasks that you submit on Massiccc are run on a High Performance Computing server. This server, provided by Inria, will let you run algorithms on very large data sets.

# Focus on the MASSICCC platform (3/4)



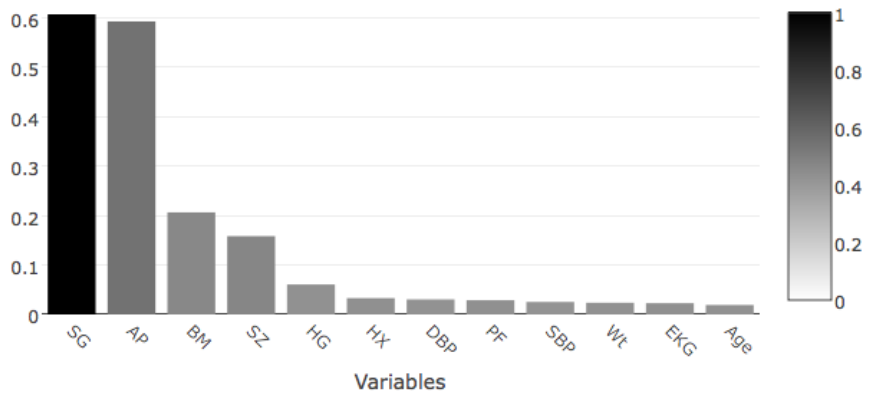
# Focus on the MASSICCC platform (4/4)

Variables | Classes | Criterion | Probabilities

## Variable Importance

This chart represents the **discriminating** level of each variable. A high value (close to one) means that the variable is highly discriminating. A low value (close to zero) means that the variable is poorly discriminating.

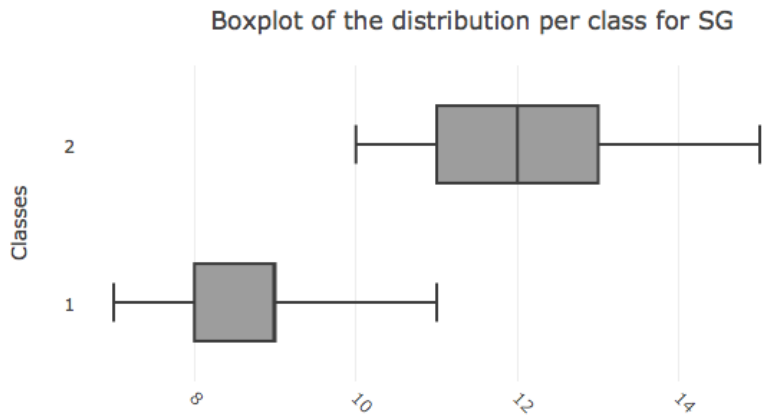
**Sort Variables :** ↓↑




## Variable Parameters

This chart shows the mean and 95%-level confidence interval for each class, for the variable selected.

SG




**SG** (Gaussian)

*Hide model parameters*

| Class 1 | Class 2 |
|---|---|
| mean: 8.934, sigma: 1.154 | mean: 12.074, sigma: 1.409 |

# The lung dataset[4]

Survival in 228 patients with advanced lung cancer from the North Central Cancer Treatment Group.
Variables:

- inst: Institution code
- time: Survival time in days
- status: censoring status 1=censored, 2=dead
- age: Age in years (numeric)
- sex: Male=1 Female=2 (categorical)
- ph.ecog: ECOG performance score (0=good 5=dead) (categorical)
- ph.karno: Karnofsky performance score (bad=0-good=100) rated by physician (numeric by interval)
- pat.karno: Karnofsky performance score as rated by patient (numeric by interval)
- meal.cal: Calories consumed at meals (numeric)
- wt.loss: Weight loss in last six months (numeric)

[4] available in the R package survival

# Preprocessing of the lung dataset

- Sample of the original dataset

| | inst | time | status | age | sex | ph.ecog | ph.karno | pat.karno | meal.cal | wt.loss |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 3 | 306.00 | 2.00 | 74.00 | 1.00 | 1 | 90 | 100 | 1175 | NA |
| 2 | 3 | 455.00 | 2.00 | 68.00 | 1.00 | 0 | 90 | 90 | 1225 | 15 |
| 3 | 3 | 1010.00 | 1.00 | 56.00 | 1.00 | 0 | 90 | 90 | NA | 15 |
| 4 | 5 | 210.00 | 2.00 | 57.00 | 1.00 | 1 | 90 | 60 | 1150 | 11 |
| 5 | 1 | 883.00 | 2.00 | 60.00 | 1.00 | 0 | 100 | 90 | NA | 0 |
| 6 | 12 | 1022.00 | 1.00 | 74.00 | 1.00 | 1 | 50 | 80 | 513 | 0 |

- Sample of the dataset imported in MASSICCC[5]

| | age | sex | ph.ecog | ph.karno | pat.karno | meal.cal | wt.loss |
|---|---|---|---|---|---|---|---|
| 1 | 74.00 | 1 | 2 | [85:95] | [95:100] | 1175 | ? |
| 2 | 68.00 | 1 | 1 | [85:95] | [85:95] | 1225 | 15 |
| 3 | 56.00 | 1 | 1 | [85:95] | [85:95] | ? | 15 |
| 4 | 57.00 | 1 | 2 | [85:95] | [55:65] | 1150 | 11 |
| 5 | 60.00 | 1 | 1 | [95:100] | [85:95] | ? | 0 |
| 6 | 74.00 | 1 | 2 | [45:55] | [75:85] | 513 | 0 |

[5] available at `https://goo.gl/FUT2CN`

# Importation of the lung dataset in MASSSIC

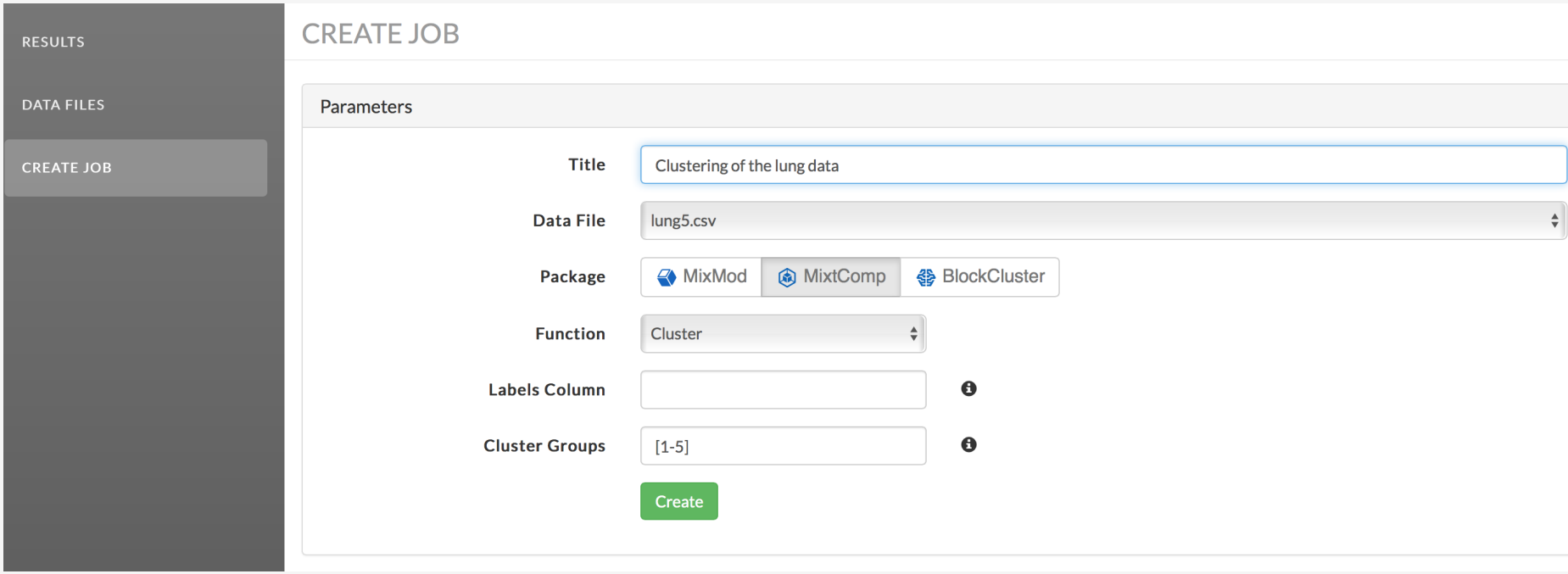
| age | sex | ph.ecog | ph.karno | pat.karno | meal.cal | wt.loss |
|---|---|---|---|---|---|---|
| Continuous | Categorical | Categorical | Continuous | Continuous | Continuous | Continuous |

Save

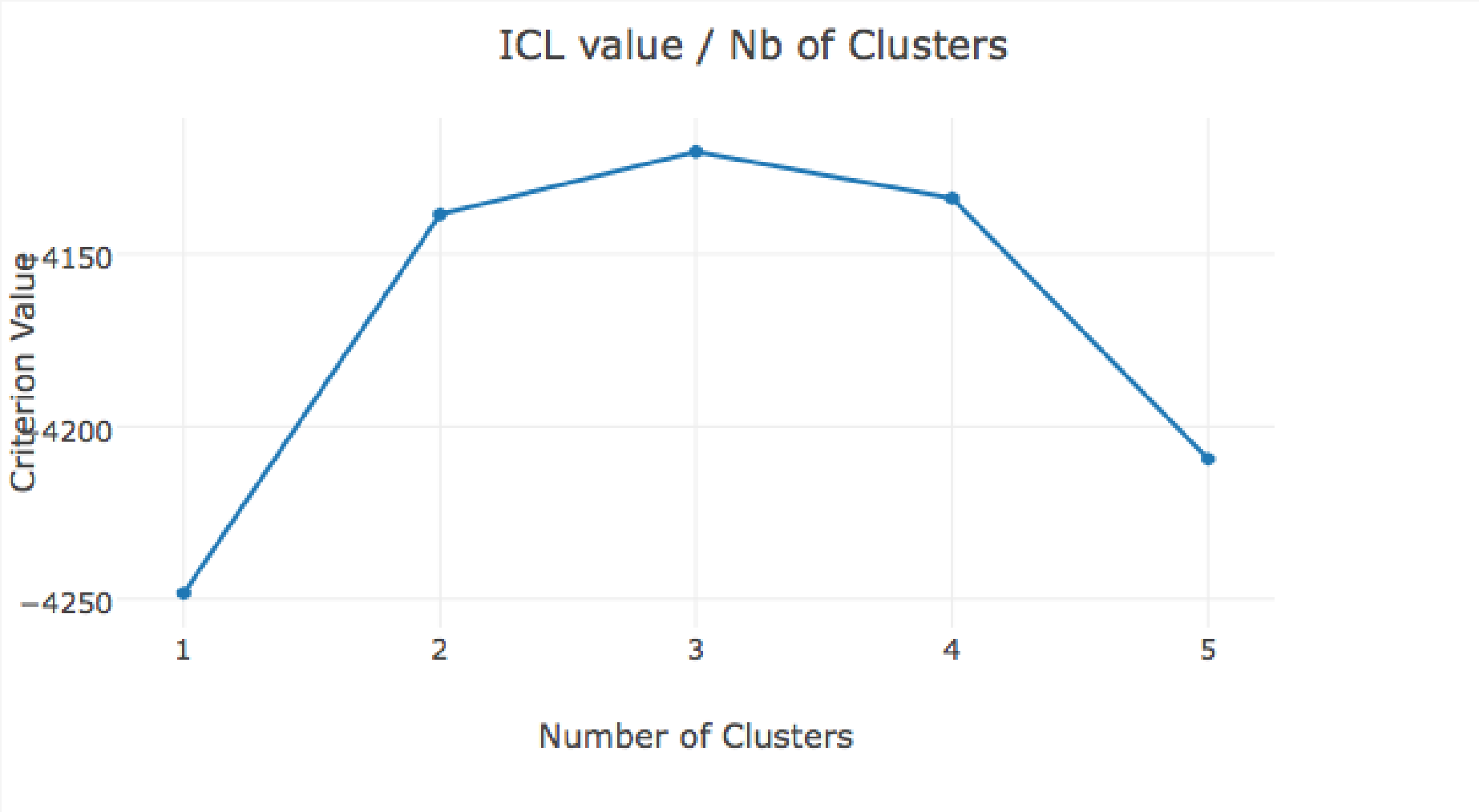
Preview

| | age | sex | ph.ecog | ph.karno | pat.karno | meal.cal | wt.loss |
|---|---|---|---|---|---|---|---|
| **0** | 74 | 1 | 2 | [85:95] | [95:100] | 1175 | ? |
| **1** | 68 | 1 | 1 | [85:95] | [85:95] | 1225 | 15 |
| **2** | 56 | 1 | 1 | [85:95] | [85:95] | ? | 15 |
| **3** | 57 | 1 | 2 | [85:95] | [55:65] | 1150 | 11 |
| **4** | 60 | 1 | 1 | [95:100] | [85:95] | ? | 0 |

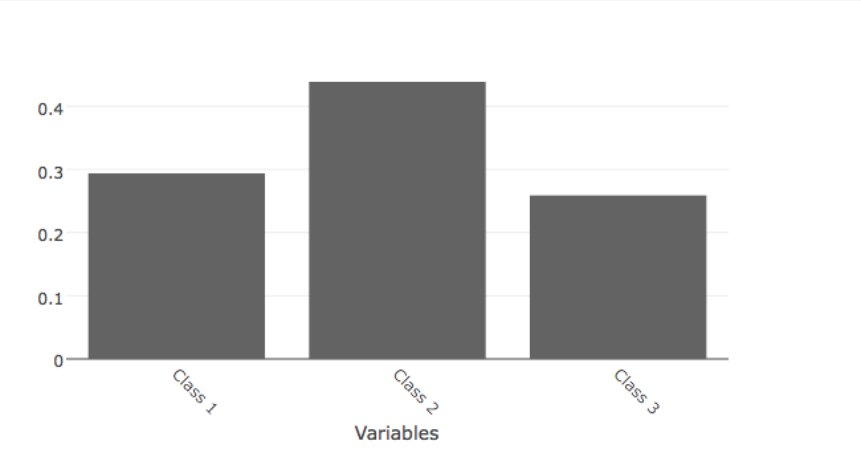
# Defining the job options in MASSSIC



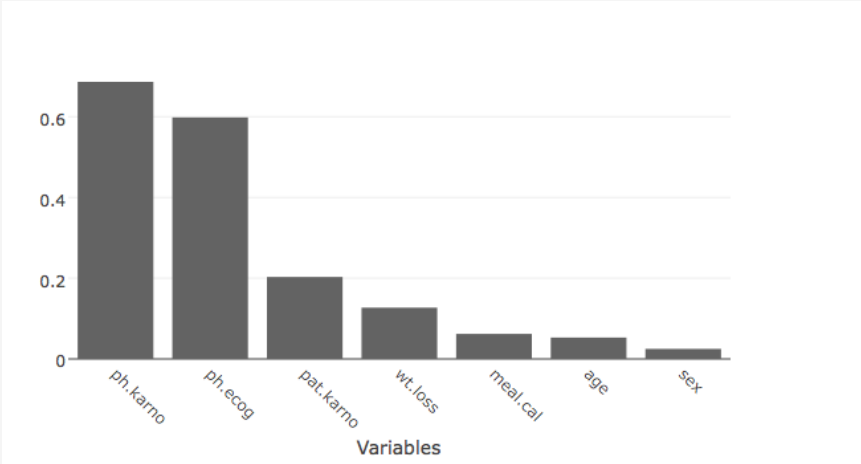
# Selected number of clusters by ICL



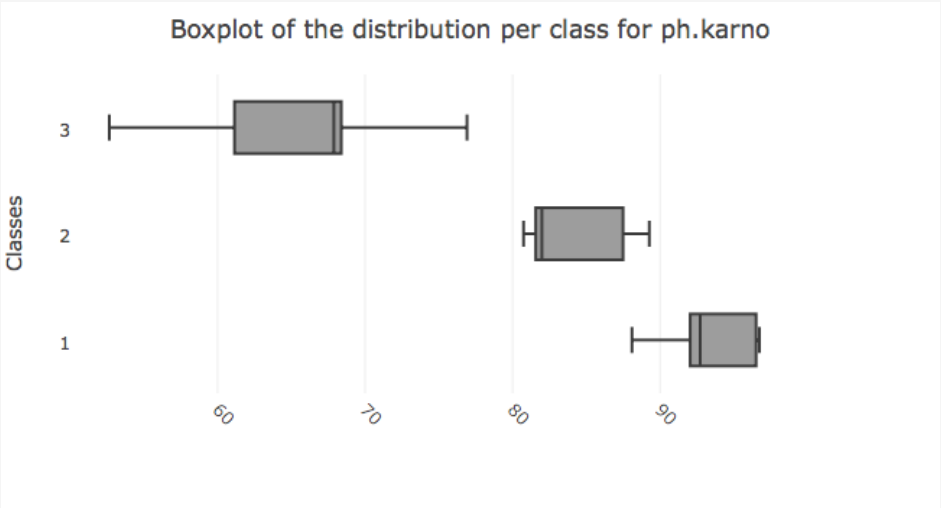
# Interpretation of the clustering in $K = 3$ clusters




Class proportions




Relative importance of the variables




Distribution of the variable ph.karno according to the cluster

# Exportation of the results as an R object

Outputs

Download Results

- **output** : output of the best model
  - strategy: strategy used for the parameters estimation
  - mixture: general characteristics of the model
  - variable: class specific parameters of the model

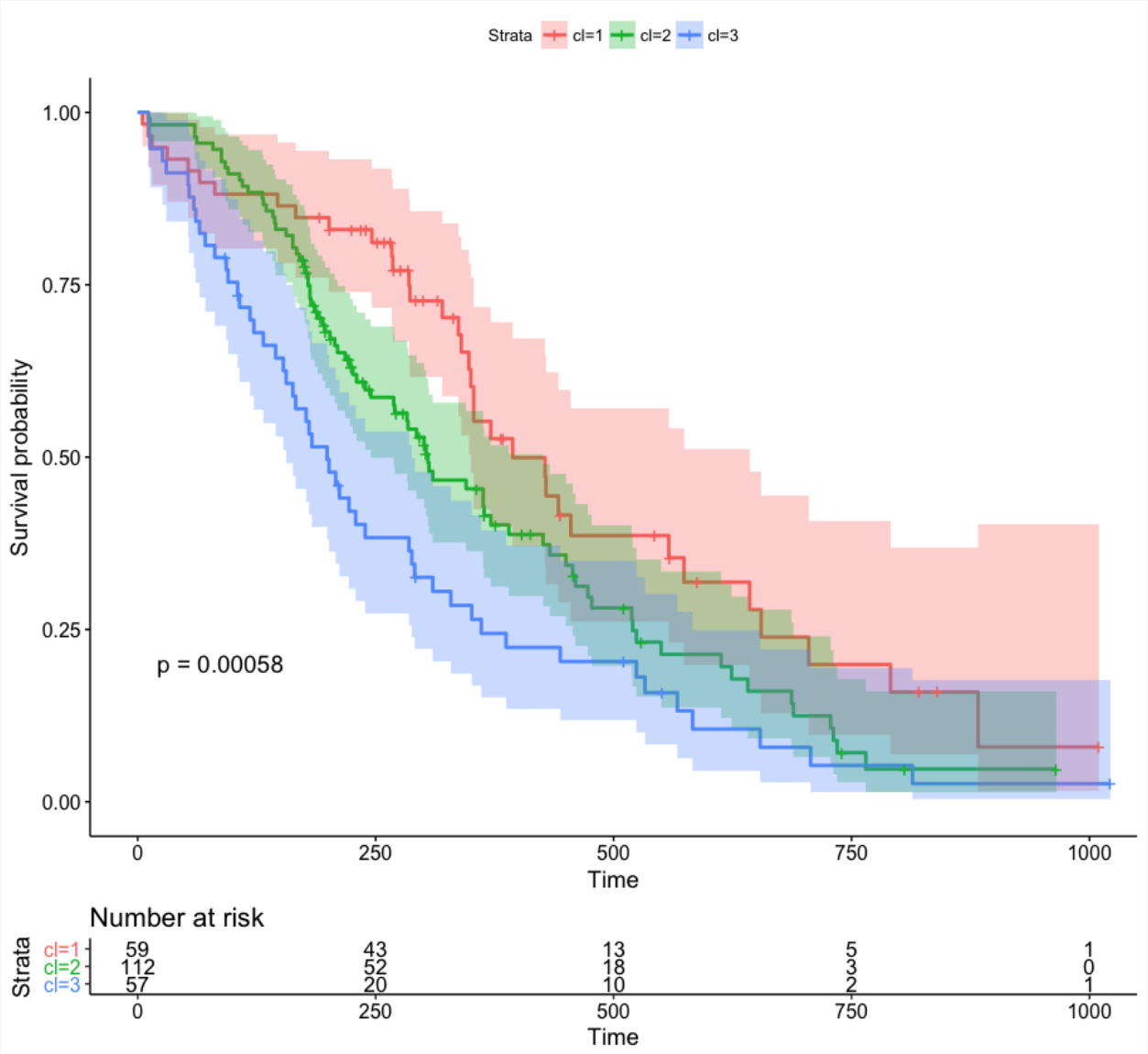
$\Rightarrow$ computation of the class posterior probabilities $P(\omega \in G_k|\mathbf{x})$ (output$variable$data$z_class$stat),

$\Rightarrow$ clustering of the individuals in the most likely cluster $\arg\max_k P(\omega \in G_k|\mathbf{x})$ (output$variable$data$z_class$completed),

$\Rightarrow$ performing a survival analysis per cluster.

# Overview

# Clusterwise survival analysis of the lung dataset (1/2)



# Clusterwise survival analysis of the lung dataset (2/2)

## Possibility to use a Cox model[a] with the cluster variable, `cl`, as explicative variable

[a] R package survival

```
coxph(formula = Surv(time, status) ~ cl, data = lung)

  n= 228, number of events= 165

      coef exp(coef) se(coef)     z Pr(>|z|)
cl2 0.4285    1.5350   0.2037 2.103 0.035447 *
cl3 0.8342    2.3029   0.2211 3.772 0.000162 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

    exp(coef) exp(-coef) lower .95 upper .95
cl2     1.535     0.6515     1.030     2.288
cl3     2.303     0.4342     1.493     3.552

Concordance= 0.604  (se = 0.024 )
Rsquare= 0.062   (max possible= 0.999 )
Likelihood ratio test= 14.54  on 2 df,   p=0.0006958
Wald test            = 14.4  on 2 df,   p=0.0007483
Score (logrank) test = 14.92  on 2 df,   p=0.0005745
```

## Significant effect of the cluster variable

# Overview

- Mixture models a very flexible tool to cluster any type of data with any type of uncertain data
- MASSICCC platform is available to perform clustering in SaaS mode
- Produced clusters can be used in a standard survival analysis