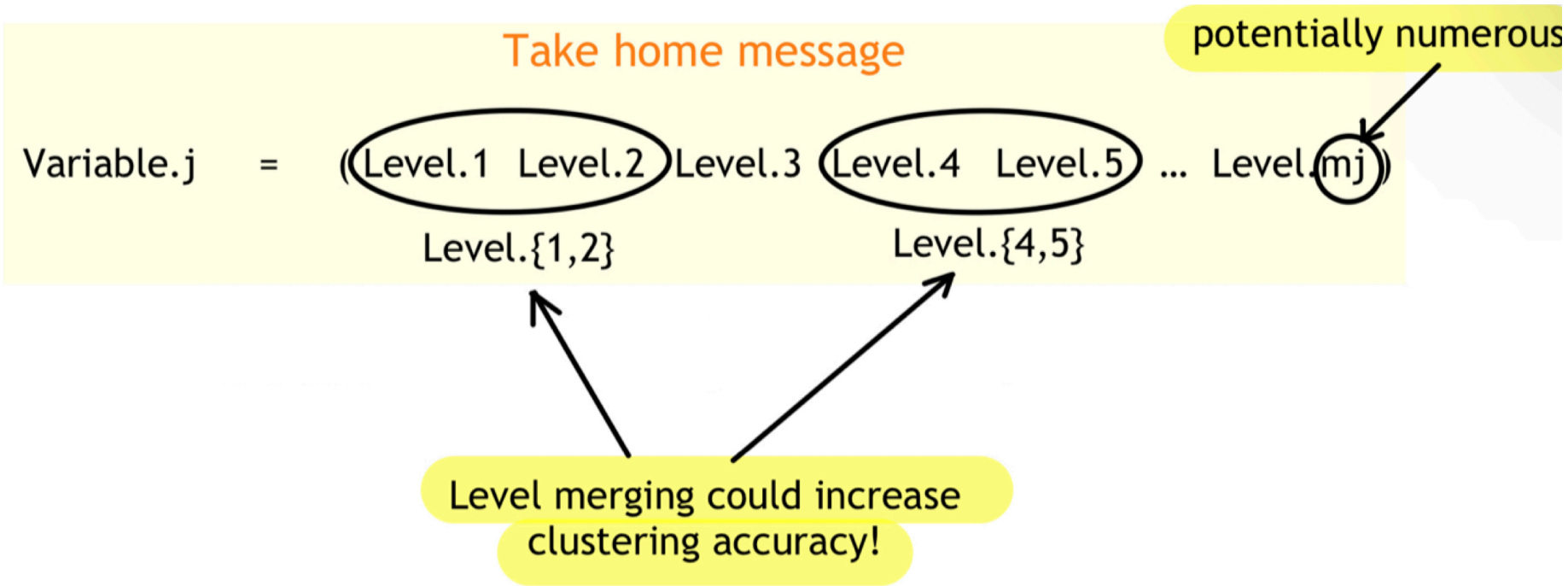
Levels Merging in the Latent Class Model

Christophe Biernacki

Levels Merging in the Latent Class Model

C. Biernacki

CFE-CMStatistics 2023, December 16-18 2023, Berlin (Germany)


Take home message
potentially numerous
Variable.j = ((Level.1 Level.2) Level.3 (Level.4 Level.5) ... Level.mj))
Level.{1,2}
Level.{4,5}
Level merging could increase clustering accuracy!

Outline

Variety of high dimensional (HD) situations

Let a data set $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_n) \in \mathcal{X}^d$.

Well explored HD cases essentially rely on d $\rightarrow$ "HD-d"

- Marketing: $d \sim 10^2$ (any kind of variables)
- microarray gene expression: $d \sim 10^2$–10^4
- Text mining $d \sim 10^4$ (count or categorical data)
- SNP data: $d \sim 10^6$
- Curves: depends on discretization but can be very high
- ...

A less explored HD case: numerous levels for categorical data... $\rightarrow$ "HD-m_j"

- Categorical data case: the jth variable has m_j response levels
- An underestimated HD case: some m_js can be large
- Ex: professional social categories ($m_j \sim 30$), US states ($m_j \sim 50$), French departments ($m_j \sim 100$), medical nomenclature ($m_j \sim 1000$)...

HD issues in clustering

- Cluster analysis is one of the main data analysis method. It aims at estimating a K groups partition $\mathbf{z} = (\mathbf{z}_1, \ldots, \mathbf{z}_n)$ of a data set $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_n) \in \mathcal{X}^d$
- Model-based clustering allows to reformulate cluster analysis as a well-posed estimation problem both for the partition $\mathbf{z}$ and for the number K of groups [McLachlan and Basford, 88]
- Model-based clustering has led practical successes, even in the challenging HD-d case [Biernacki and Maugis, 17]
- But less works on model-based clustering dedicated to the HD-m_j case

Have now a look at

- Classical model-based clustering for HD-d methods
- The need of model-based clustering for HD-m_j methods

The general trade-off bias/variance as a guideline

The fundamental statistical principle

Always minimize an error err between truth ($\mathbf{z}$) and estimate ($\hat{\mathbf{z}}$)

- Gap between true ($\mathbf{z}$) and model-based ($\mathcal{Z}_\mathsf{p}$) partitions:

$$\mathbf{z}^* = \arg \min_{\tilde{\mathbf{z}} \in \mathcal{Z}_\mathsf{p}} \mathrm{err}(\mathbf{z}, \tilde{\mathbf{z}})$$

- Estimation $\hat{\mathbf{z}}$ of $\mathbf{z}^*$ in $\mathcal{Z}_\mathsf{p}$: any relevant method (bias, consistency, efficiency...)
- Fundamental decomposition of the observed error $\mathrm{err}(\mathbf{z}, \hat{\mathbf{z}})$:

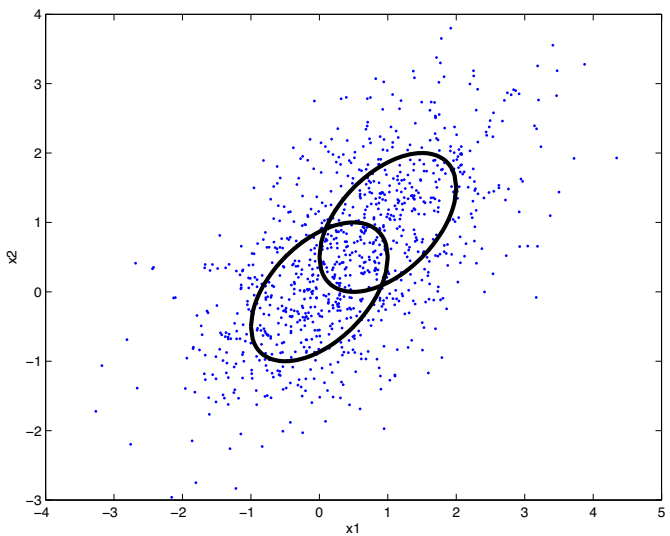
$$\begin{aligned} \mathrm{err}(\mathbf{z}, \hat{\mathbf{z}}) &= \Big\{\mathrm{err}(\mathbf{z}, \mathbf{z}^*) - \mathrm{err}(\mathbf{z}, \mathbf{z})\Big\} + \Big\{\mathrm{err}(\mathbf{z}, \hat{\mathbf{z}}) - \mathrm{err}(\mathbf{z}, \mathbf{z}^*)\Big\} \\ &= \Big\{\text{bias}\Big\} + \Big\{\text{variance}\Big\} \\ &= \Big\{\text{error of approximation}\Big\} + \Big\{\text{error of estimation}\Big\} \end{aligned}$$

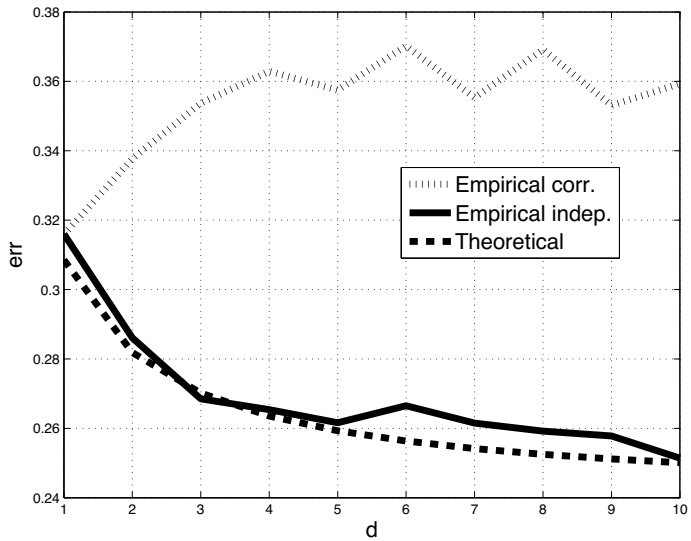
Example for HD-d: reduce variance, accept bias

A two-component d-variate Gaussian mixture with intra-dependency:

$$\pi_1 = \pi_2 = \frac{1}{2}, \quad \mathbf{X}_1|z_{11} = 1 \sim \mathrm{N}_d(\mathbf{0}, \mathbf{\Sigma}), \quad \mathbf{X}_1|z_{12} = 1 \sim \mathrm{N}_d(\mathbf{1}, \mathbf{\Sigma})$$

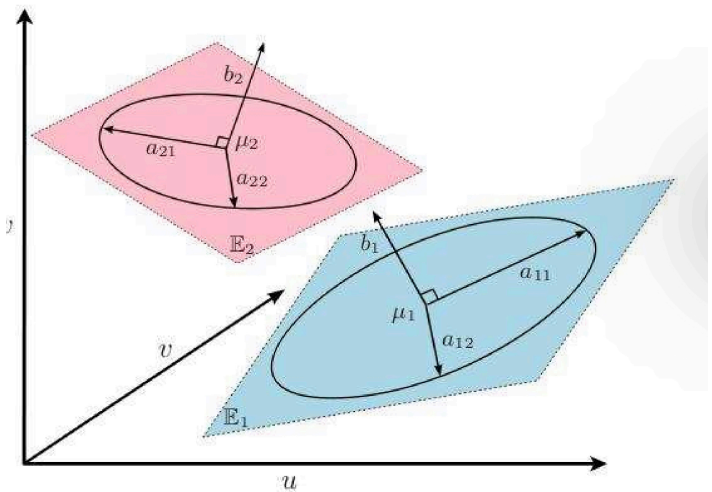
- Each variable provides equal and own separation information
- Theoretical error decreases when d grows: $\mathrm{err}_{theo} = \Phi(-\|\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1\|_{\mathbf{\Sigma}^{-1}}/2)$
- Empirical error rate with the (true) intra-correlated model worse with d
- Empirical error rate with the (false) intra-independent model better with d!



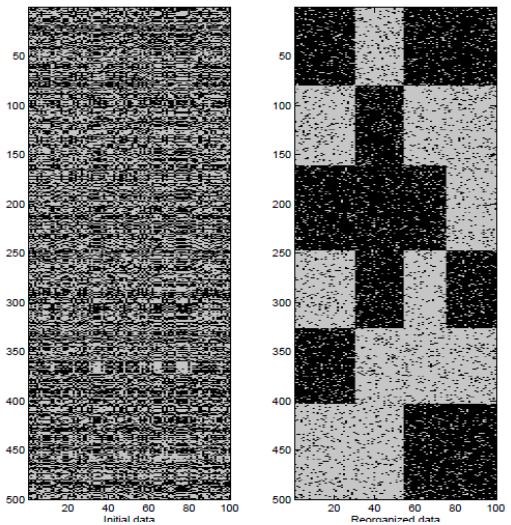


Other classical ways for reducing variance in HD-d

- Dimension reduction
  - In non-canonical space (PCA-like typically): Gaussian mixture of factor analysers [Ghahramani and Hinton, 97], [McLachlan et al., 03], [McNicholas and Murphy, 10], HD Gaussian models [Bouveyron et al., 07]
  - In the canonical space (variable selection): Gaussian "variable selection" [Raftery and Dean, 06], [Maugis et al., 09a], [Maugis et al., 09b], [Sedki et al., 14]
- Model parsimony (constraints on model parameters): previous example, co-clustering [Govaert, 2011], [Biernacki et al., 23]



Continuous case



Categorical case

HD-m_j: limit of the classical Latent Class Model (LCM)

- When $\mathcal{X}$ corresponds to d categorical variables (frequent situation), the jth one having m_j response levels,the standard latent class model (LCM, [Goodman, 74]) is used

$$p(\mathbf{x}_i;\boldsymbol{\theta}) = \sum_{k=1}^{K} \pi_k \prod_{j=1}^{d} \prod_{h=1}^{m_j} (\alpha_k^{jh})^{x_i^{jh}}, \quad \text{where } \boldsymbol{\theta} = (\boldsymbol{\pi}, \boldsymbol{\alpha}), \boldsymbol{\pi} = (\pi_k), \boldsymbol{\alpha} = (\boldsymbol{\alpha}_k)$$

- When some m_js are large another kind of dimensionality curse appears

The standard latent class model is affected by such a curse of dimension paradigm since the number of free parameters if equal to $K - 1 + K\sum_{j=1}^{d}(m_j - 1)$

Outline

An early work for parsimony in HD-m_j

[Lebret et al., 15] proposes four parsimonious versions of LCM. They correspond to an extension of the parameterization of Bernoulli distributions used by [Celeux and Govaert, 91] for clustering and also by [Aitchinson and Aitken, 76] for kernel discriminant analysis.

The basic idea is

$$\boldsymbol{\alpha}_k^j = (\alpha_k^{j1}, \ldots, \alpha_k^{jm_j}) = (\beta_k^j, \ldots, \beta_k^j, \gamma_k^j, \beta_k^j, \ldots, \beta_k^j) \quad \text{with} \quad \gamma_k^j > \beta_k^j.$$

Since $\sum_{h=1}^{m_j} \alpha_k^{jh} = 1$, then $(m_j - 1)\beta_k^j + \gamma_k^j = 1$ and, thus, $\beta_k^j = (1 - \gamma_k^j)/(m_j - 1)$.

It leads to a very parsimonious (only $K - 1 + Kd$ parameters) but poor flexible model. We propose to extend this idea by a level partitioning (here an example with 3 level clusters)

$$\boldsymbol{\alpha}_k^j = (\alpha_k^{j1}, \alpha_k^{j2}, \alpha_k^{j3}, \alpha_k^{j4}, \alpha_k^{j5}) = (\beta_k^{j\boxed{1}}, \beta_k^{j\boxed{2}}, \beta_k^{j\boxed{1}}, \beta_k^{j\boxed{3}}, \beta_k^{j\boxed{2}})$$

LCM-LM: a LCM extension for the case of numerous levels

- Idea: constrain some levels to share the same parameter value (level partitionning)
- Notation: for each variable j, L_j clusters of levels $\mathbf{w}_j = (\mathbf{w}_j^1, \ldots, \mathbf{w}_j^{m_j})$
- New model LCM-LM (*Latent Class Model by Levels Merging)*: from the full level clustering $\mathbf{w} = (\mathbf{w}_1, \ldots, \mathbf{w}_d)$, the new LCM is defined by

$$p(\mathbf{x}_i|\mathbf{w};\boldsymbol{\vartheta}) = \sum_{k=1}^{K} \pi_k \prod_{j=1}^{d} \prod_{\ell=1}^{L_j} \left(\frac{\beta_k^{j\ell}}{\sum_{h=1}^{m_j} w_j^{h\ell}} \right)^{\sum_{h=1}^{m_j} w_j^{h\ell} x_i^{jh}}$$

  with $\boldsymbol{\beta} = (\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_K)$, $\boldsymbol{\beta}_k = (\beta_k^{j\ell}; j = 1, \ldots, d; \ell = 1, \ldots, L_j)$ and $\boldsymbol{\vartheta} = (\boldsymbol{\pi}, \boldsymbol{\beta})$
- An interesting extreme case: if $L_j = 1$ then LCM-LM acts as a variable selection (no effect of variable j on the clustering)

LCM-LM is parsimonious with regards to the number of levels since its number of parameters is equal to $K - 1 + K\sum_{j=1}^{d}(L_j - 1)$

Parameter estimation

Hint

Possible to reformulating the log-likelihood of LCM-LM as a classical LCM...

- LCM-LM $\Leftrightarrow$ LCM with the following new data $\mathbf{y}_{\mathbf{w}} = (\mathbf{y}_{\mathbf{w},1}, \ldots, \mathbf{y}_{\mathbf{w},n})$, and where $\mathbf{y}_{\mathbf{w},i} = (y_{\mathbf{w},i}^{j\ell}; j = 1, \ldots, d; \ell = 1, \ldots, L_j)$ with $y_{\mathbf{w},i}^{j\ell} = \sum_{h=1}^{m_j} w_j^{h\ell} x_i^{jh}$
- Parameter estimation: $\boxed{L(\boldsymbol{\vartheta}|\mathbf{w}; \mathbf{x}) = L(\boldsymbol{\vartheta}; \mathbf{y}_{\mathbf{w}}) + c_{\mathbf{w}}}$, where

$$c_{\mathbf{w}} = -\sum_{i=1}^{n} \sum_{j=1}^{d} \sum_{\ell=1}^{L_j} y_{\mathbf{w},i}^{j\ell} \ln \left(\sum_{h=1}^{m_j} w_j^{h\ell} \right)$$

  is a constant with regards to the parameter $\boldsymbol{\vartheta}$
- Since $c_{\mathbf{w}}$ is independent of $\boldsymbol{\vartheta}$ then use a classical LCM EM

Related EM algorithm

As a consequence of the preeceeding remark, estimating $\boldsymbol{\vartheta}$ can be performed by any classical EM algorithm [Dempster et al., 77] for LCM implemented in any classical existing package (for instance Rmixmod [Lebret et al., 15] or RMixtComp[1]).

Starting from $\boldsymbol{\vartheta}^0 = (\boldsymbol{\pi}^0, \boldsymbol{\beta}^0)$, iteration $r > 0$ of this EM is expressed as:

E step: Conditional probability that individual i arose from cluster k

$$t_{ik}(\boldsymbol{\vartheta}^r) = \frac{\pi_k^r \mathrm{p}(\mathbf{y}_{\mathbf{w},i}; \boldsymbol{\beta}_k^r)}{\mathrm{p}(\mathbf{y}_{\mathbf{w},i}; \boldsymbol{\vartheta}^r)}.$$

M step: Updating the mixture parameter estimates by maximizing the expected complete log-likelihood

$$\pi_k^{r+1} = \frac{\sum_i t_{ik}(\boldsymbol{\vartheta}^r)}{n} \quad \text{and} \quad (\beta_k^{j\ell})^{r+1} = \frac{\sum_{i=1}^n t_{ik}(\boldsymbol{\vartheta}^r) y_{\mathbf{w},i}^{j\ell}}{\sum_{i=1}^n t_{ik}(\boldsymbol{\vartheta}^r)}.$$

The stopping rule can rely on a plateau of the log-likelihood (equally $L(\boldsymbol{\vartheta}; \mathbf{y}_\mathbf{w})$ or $L(\boldsymbol{\vartheta}|\mathbf{w}; \mathbf{x})$) or a given iteration number. Several runs should be performed also from different starting parameters for avoiding local maxima traps.

[1] https://cran.r-project.org/web/packages/RMixtComp/index.html

Model selection: criterion ICL

ICL criterion [Biernacki et al., 00], [Biernacki et al., 11]: just written as a penalization of the classical LCM ICL

$$\boxed{\text{ICL}_{\mathbf{w}}^{\text{LCM-LM}} = \ln p(\mathbf{x}, \hat{\mathbf{z}}_{\mathbf{w}} | \mathbf{w}) = \text{ICL}_{\mathbf{w}}^{\text{LCM}} + \underbrace{c_{\mathbf{w}}}_{!}}$$

where $\text{ICL}_{\mathbf{w}}^{\text{LCM}}$ is the ICL criterion for LCM. This latter is classically defined as follows

$$\text{ICL}_{\mathbf{w}}^{\text{LCM}} = \sum_{k=1}^{K}\sum_{j=1}^{d}\left\{\sum_{\ell=1}^{L_j}\ln\Gamma\left(\hat{n}_{\mathbf{w},k}^{j\ell} + \tfrac{1}{2}\right) - \ln\Gamma(\hat{n}_{\mathbf{w},k} + \tfrac{L_j}{2})\right\} - \ln\Gamma(n + \tfrac{K}{2}) + \ln\Gamma(\tfrac{K}{2})$$

$$+K\sum_{j=1}^{d}\left\{\ln\Gamma(\tfrac{L_j}{2}) - L_j\ln\Gamma(\tfrac{1}{2})\right\} + \sum_{k=1}^{K}\ln\Gamma(\hat{n}_{\mathbf{w},k} + \tfrac{1}{2}) - K\ln\Gamma(\tfrac{1}{2}),$$

where $\hat{n}_{\mathbf{w},k} = \#\{i : \hat{z}_{\mathbf{w},ik} = 1\}$ and $\hat{n}_{\mathbf{w},k}^{j\ell} = \#\{i : \hat{z}_{\mathbf{w},ik} = 1, y_{\mathbf{w},i}^{j\ell} = 1\}$.

As a consequence, the $\text{ICL}_{\mathbf{w}}^{\text{LCM-LM}}$ is itself easily calculated also.

Model selection: MICL criterion

MICL criterion (*Maximum Integrated Complete-data Likelihood*) [Marbac and Sedki, 15] is a variant of ICL avoiding multiple parameter estimation

$$\text{MICL}_{\mathbf{w}}^{\text{LCM-LM}} = \ln p(\mathbf{x}, \mathbf{z}_{\mathbf{w}}^*|\mathbf{w}) \quad \text{with} \quad \mathbf{z}_{\mathbf{w}}^* = \arg\max_{\mathbf{z}\in\mathcal{Z}} \ln p(\mathbf{x}, \mathbf{z}|\mathbf{w}).$$

Then, the model $\mathbf{w}^*$ maximizing $\text{MICL}_{\mathbf{w}}^{\text{LCM-LM}}$ is retained: $\mathbf{w}^* = \arg\max_{\mathbf{w}\in\mathcal{W}} \text{MICL}_{\mathbf{w}}$. Finally a specific two step algorithm derived from [Marbac and Sedki, 15] can be used for performing this optimization (not implemented yet).

We do not detail this algorithm here but the idea is the following. Starting from a value $\mathbf{w}^0$ uniformly sampled in the corresponding space and then a value $\mathbf{z}^0$ being deduced from the MAP rule of the associated maximum likelihood estimate, iteration $s > 0$ of the algorithm is composed by the following two steps:

- Partition step: Fix $\mathbf{z}^{s+1}$ such that $\ln p(\mathbf{x}, \mathbf{z}^{s+1}|\mathbf{w}^s) \geq \ln p(\mathbf{x}, \mathbf{z}^s|\mathbf{w}^s)$.
- Model step: Fix $\mathbf{w}^{s+1}$ such that $\ln p(\mathbf{x}, \mathbf{z}^{s+1}|\mathbf{w}^{s+1}) \geq \ln p(\mathbf{x}, \mathbf{z}^{s+1}|\mathbf{w}^s)$.

Outline

Prostate cancer data: description

- Individuals: 506 patients on the basis of petrial variates alone for the prostate cancer clinical trial data of [Byar and Green, 80], described by four categorical variables with various numbers of levels: performance rating (so-called PF, with 4 levels), cardiovascular disease history (HX, 2 levels), electrocardiogram code (EKG, 7 levels) and bone metastases (BM, 2 levels). There exists two stages of the desease (Stage 3 and Stage 4).
- Some missing data: 62 missing values ($\approx 1\%$)

We forget the classes (Stages of the desease) for performing clustering

Prostate cancer data: LCM-LM vs LCM

- EM-like package: RMixtcomp[2] for LCM categorical data and also dealing with missing values
- Levels' distribution: we see below the frequency of each level for all these variables (missing values are denoted by a "?" and appear below as a level, but mathematical speaking it is not a level obviously)

```
    PF       HX            EKG       BM
?:  4    ?:  4    1       :168    ?:  4
1:450    1:289    5       :150    1:420
2: 37    2:213    6       : 75    2: 82
3: 13             3       : 51
4:  2             4       : 26
                  2       : 23
                  (Other): 13
```

- LCM-LM proposal: since level 4 of variable PF is under-represented (only 2 individuals), we propose to merge it with level 3 of the same variable
- Results with $K = 2$: great improvement both in ICL and error values

	LCM	LCM-LM
ICL	-1728.4	**-1618.4**
error	49.2%	**28.8%**

[2] https://cran.r-project.org/web/packages/RMixtComp/index.html

Outline

Conclusion

- Very promizing preliminary numerical results
- Need to still implement MICL
- Extend simultaneously to high dimension both for the number of variables and the number of levels (merge co-clustering and LCM-LM?)