greed: model-based hierarchical clustering with the exact ICL

Based on an ADAC article by **E. Côme, N. Jouvin, P. Latouche, C. Bouveyron**

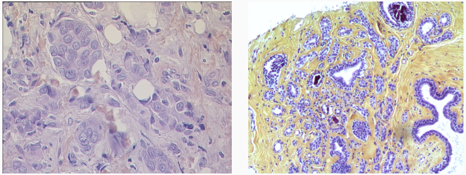
Happy R - Friday 24, 2022

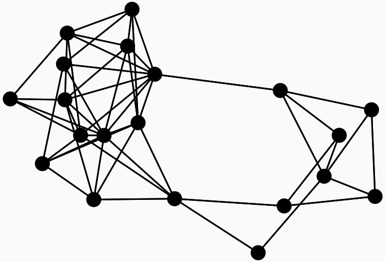
Université Gustave Eiffel

INRAE

Clustering in a nutshell

It's a data's world...

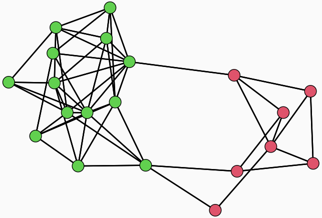


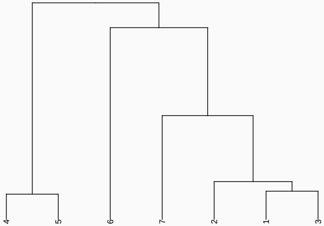


Clustering

Clustering is the task of grouping objects together into classes or *clusters*, in an unsupervised fashion based on some criterion.

Doc 1	"Lésions cancéreuses (. . .) carcinome canalaire"
Doc 2	"Lésions cancéreuses (. . .) carcinome lobulaire"
. . .	. . .
Doc n	"Lésions bénignes (. . .) métaplasie"





Organization

The framework: dicrete latent variable models

The exact integrated classification likelihood

Greedy maximimization of ICLex: a genetic algorithm

A quick introduction to the *greed* package

The framework: dicrete latent variable models

The rationale of model-based approaches

Observe $\boldsymbol{X}$ related to n objects

Search for $\boldsymbol{z}_i \in \{0,1\}^K$ the cluster assignment of object i

Assume $\boldsymbol{Z} = \{\boldsymbol{z}_i\}$ contains independent and identically distributed (*i.i.d.*) discrete latent variables

$$p(\boldsymbol{Z} \mid \boldsymbol{\pi}) = \prod_{i=1}^{n} \mathcal{M}_K(\boldsymbol{z}_i \mid 1, \boldsymbol{\pi})$$

Posit a statistical model on $\boldsymbol{X} \mid \boldsymbol{Z}, \boldsymbol{\theta}$

Discrete latent variable models (DLVMs)

Conditional independence of the observations given $\boldsymbol{Z}$:

$$p(\boldsymbol{X} \mid \boldsymbol{Z}, \boldsymbol{\theta}) = \prod_{\boldsymbol{x} \in \boldsymbol{X}} p(\boldsymbol{x} \mid \boldsymbol{Z}, \boldsymbol{\theta}) \qquad \text{(DLVMs)}$$

Example 1: Finite Mixture Models (FMM)

Observations $\boldsymbol{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$ are *i.i.d.* inside a cluster

$$\forall i, \quad \boldsymbol{x}_i \mid \{z_{ik} = 1\} \sim p(\cdot \mid \boldsymbol{\theta}_k)$$

- Gaussian mixture model: $p(\boldsymbol{x}_i \mid \boldsymbol{\theta}_k) = \mathcal{N}_p(\boldsymbol{x}_i \mid \boldsymbol{m}_k, \boldsymbol{S}_k)$
- Mixture of multinomials: $p(\boldsymbol{x}_i \mid \boldsymbol{\theta}_k) = \mathcal{M}_p(\boldsymbol{x}_i \mid \boldsymbol{\theta}_k)$

$$p(\boldsymbol{X} \mid \boldsymbol{\pi}, \boldsymbol{\theta}) = \prod_{i=1}^n p(\boldsymbol{x}_i \mid \boldsymbol{\pi}, \boldsymbol{\theta}) = \prod_{i=1}^n \sum_{k=1}^K \pi_k p(\boldsymbol{x}_i \mid \boldsymbol{\theta}_k)$$

Discrete latent variable models (DLVMs)

Conditional independence of the observations given $\boldsymbol{Z}$:

$$p(\boldsymbol{X} \mid \boldsymbol{Z}, \boldsymbol{\theta}) = \prod_{\boldsymbol{x} \in \boldsymbol{X}} p(\boldsymbol{x} \mid \boldsymbol{Z}, \boldsymbol{\theta}) \tag{DLVMs}$$

Example 2: Stochastic Block Model (SBM)

Observe n^2 edges $\boldsymbol{X} = \{x_{ij}\}_{ij}$, cluster n nodes

$$\forall (i,j), \quad x_{ij} \mid \{z_{ik} z_{jl} = 1\} \sim p(\cdot \mid \boldsymbol{\theta}_{kl})$$

Edges are *i.i.d.* inside a *block* of clusters, **not marginally**

- Binary SBM: $p(x_{ij} \mid \boldsymbol{\theta}_{kl}) = \mathcal{B}(x_{ij} \mid \theta_{kl})$
- Poisson SBM: $p(x_{ij} \mid \boldsymbol{\theta}_{kl}) = \mathcal{P}(x_{ij} \mid \theta_{kl})$

Clustering and model selection in DLVMs

Standard approaches use a two-stage procedure

1. **Inference**: Fix K
   - Estimate $\hat{\boldsymbol{\pi}}, \hat{\boldsymbol{\theta}}$, *e.g.* by maximum-likelihood
   - $\boldsymbol{Z}$ is estimated by some $\hat{\boldsymbol{Z}}$ (*e.g.* MAP estimation)
2. **Model selection**: choose $K^\star$ maximizing a given criterion, *e.g.* AIC, BIC...

Clustering and model selection in DLVMs

Standard approaches use a two-stage procedure

1. **Inference**: Fix K
   - Estimate $\hat{\boldsymbol{\pi}}, \hat{\boldsymbol{\theta}}$, *e.g.* by maximum-likelihood
   - $\boldsymbol{Z}$ is estimated by some $\hat{\boldsymbol{Z}}$ (*e.g.* MAP estimation)
2. **Model selection**: choose $K^\star$ maximizing a given criterion, *e.g.* AIC, BIC...

Clustering context: Integrated Classification Likelihood (ICL, Biernacki et al. 2000)

$$\log p(\boldsymbol{X}, \boldsymbol{Z} \mid K) = \log \int_{\boldsymbol{\pi}} \int_{\boldsymbol{\theta}} p(\boldsymbol{X}, \boldsymbol{Z}, \boldsymbol{\theta}, \boldsymbol{\pi} \mid K) \, \mathrm{d}\boldsymbol{\theta} \, \mathrm{d}\boldsymbol{\pi} \tag{1}$$

À la BIC criterion via a combination of Laplace and Stirling approximations

$$\mathrm{ICL}_{BIC}(K) = \log p(\boldsymbol{X}, \hat{\boldsymbol{Z}} \mid \hat{\boldsymbol{\pi}}, \hat{\boldsymbol{\theta}}, K) - \text{penalty}(K)$$

The exact integrated classification likelihood

Exact integrated classification likelihood

Proposition (Fubini)

With a factorized prior: $p(\boldsymbol{\theta}, \boldsymbol{\pi}) = p(\boldsymbol{\theta} \mid \boldsymbol{\beta})\, p(\boldsymbol{\pi} \mid \boldsymbol{\alpha})$

$$\mathrm{ICL}_{ex}(\boldsymbol{Z}, \boldsymbol{\alpha}, \boldsymbol{\beta}) = \underbrace{\log p(\boldsymbol{X} \mid \boldsymbol{Z}, \boldsymbol{\beta})}_{(1)} + \underbrace{\log p(\boldsymbol{Z} \mid \boldsymbol{\alpha})}_{(2)}$$

(1) *Conjugate* prior for exact available in standard DLVMs, *e.g.*

- **MoM** or **LCA** (Biernacki et al. 2010; Tessier et al. 2006)
- **Binary SBM** (Côme et al. 2015), **dc-SBM** (**come2021hierarchical**)
- **GMM** (Bertoletti et al. 2015), modulo *informative* prior

Exact integrated classification likelihood

Proposition (Fubini)

With a factorized prior: $p(\boldsymbol{\theta}, \boldsymbol{\pi}) = p(\boldsymbol{\theta} \mid \boldsymbol{\beta})\, p(\boldsymbol{\pi} \mid \boldsymbol{\alpha})$

$$\mathrm{ICL}_{ex}(\boldsymbol{Z}, \boldsymbol{\alpha}, \boldsymbol{\beta}) = \underbrace{\log p(\boldsymbol{X} \mid \boldsymbol{Z}, \boldsymbol{\beta})}_{(1)} + \underbrace{\log p(\boldsymbol{Z} \mid \boldsymbol{\alpha})}_{(2)}$$

(1) *Conjugate* prior for exact available in standard DLVMs, *e.g.*

- **MoM** or **LCA** (Biernacki et al. 2010; Tessier et al. 2006)
- **Binary SBM** (Côme et al. 2015), **dc-SBM** (**come2021hierarchical**)
- **GMM** (Bertoletti et al. 2015), modulo *informative* prior

(2) Common part to all DLVMs - Exact expression with universal prior

$$p(\boldsymbol{\pi} \mid \boldsymbol{\alpha}) = \mathcal{D}_K\left(\boldsymbol{\pi} \mid \boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_K)\right)$$

Set $\alpha_k = \alpha, \forall k$ — *e.g.* Uniform ($\alpha = 1$) or Jeffreys ($\alpha = 1/2$)

Overview of (come2021hierarchical)

- Generic approach: applies in the framework of DLVMs
- ICL_{ex} criterion as a clustering objective

Twofold contribution:

1. Genetic algorithm: greedy maximization w.r.t $\boldsymbol{Z}$ and K

$$\boldsymbol{Z}^{(K^\star)} \in \arg\max_{K,\boldsymbol{Z}} \mathrm{ICL}_{ex}(\boldsymbol{Z}, K) \quad (2)$$

   - Jointly performs clustering and model selection
   - Bypass inference of $\boldsymbol{\theta}$ and $\boldsymbol{\pi}$

2. Hierarchical algorithm: start from $\boldsymbol{Z}^{(K^\star)}$ and merge clusters using $\log \alpha$

$$\boldsymbol{Z}^{(K^\star)} \leq \ldots \leq \boldsymbol{Z}^{(1)}$$

   - Produces a **dendrogram**
   - Ordering of the cluster (useful for visualization)

Overview of (come2021hierarchical)

- Generic approach: applies in the framework of DLVMs
- ICL_{ex} criterion as a clustering objective

Twofold contribution:

1. Genetic algorithm: greedy maximization w.r.t $\boldsymbol{Z}$ and K

$$\boldsymbol{Z}^{(K^\star)} \in \arg\max_{K,\boldsymbol{Z}} \mathrm{ICL}_{ex}(\boldsymbol{Z}, K) \quad (2)$$

   - Jointly performs clustering and model selection
   - Bypass inference of $\boldsymbol{\theta}$ and $\boldsymbol{\pi}$

2. Hierarchical algorithm: start from $\boldsymbol{Z}^{(K^\star)}$ and merge clusters using $\log \alpha$

$$\boldsymbol{Z}^{(K^\star)} \leq \ldots \leq \boldsymbol{Z}^{(1)}$$

   - Produces a **dendrogram**
   - Ordering of the cluster (useful for visualization)

Greedy maximimimization of ICLex: a genetic algorithm

Greedy local search

Goal: Optimize ICL_{ex} directly with respect to $\boldsymbol{Z}$

$$\boldsymbol{Z}^{(K^\star)} \in \arg\max_{K,\boldsymbol{Z}} \mathrm{ICL}_{ex}(\boldsymbol{Z}, K) \tag{3}$$

Combinatorial problem: B_n possible partitions (n-th Bell number)

Goal: Optimize ICL_{ex} directly with respect to $\boldsymbol{Z}$

$$\boldsymbol{Z}^{(K^\star)} \in \arg\max_{K,\boldsymbol{Z}} \mathrm{ICL}_{ex}(\boldsymbol{Z}, K) \tag{3}$$

Combinatorial problem: B_n possible partitions (n-th Bell number)

Existing solution: greedy local search (Côme et al. 2015)

1. Starts with an overly segmented $\boldsymbol{Z}^{(K)}$
2. Swap moves: greedily change clusters until convergence (empty clusters)
3. Merge moves: greedily merge clusters until convergence

Output: locally optimal partition $\boldsymbol{Z}^{(K^\star)}$ with $K^\star$ automatically selected

Improving greedy local search

→ **Pros**: Fast & competitive w.r.t alternatives *e.g.* (V)EM

→ **Cons**: Exploitation is good, but **exploration** is hard

- Initialization: ICL_{ex} is highly-multimodal ! seeding is important
- Existence of sub-optimal local maxima (in term of clustering, i.e. **underfitting**)

Our proposition: improve exploration with a genetic algorithm (GA)

→ Grow a set of V candidates

→ Recombination, mutation, natural selection

→ Hybrid GA: use greedy local search on each generation.

Algorithm: Standard genetic algorithm

Input: Population size: V, probability of mutation: pm, $maxgen$

```
// Initialization
```

1. Start with V random partitions

```
// Population evolution
```

for $n = 1$ *to* $maxgen$ **do**

Sample V pairs of candidates (according to ICL_{ex} rank)

for *each pair* $(\boldsymbol{Z}^1, \boldsymbol{Z}^2)$ **do**

- Recombination: $\boldsymbol{Z} = \mathrm{Cross}(\boldsymbol{Z}^1, \boldsymbol{Z}^2)$ (cross-over)
- Apply random splits to $\boldsymbol{Z}$ with proba pm (**mutation**)

end

end

Algorithm: Hybrid genetic algorithm

Input: Population size: V, probability of mutation: pm, $maxgen$

`// Initialization`

1. Start with V random partitions
2. Update each partitions with greedy swapping (delete empty clusters)

`// Population evolution`

for $n = 1$ *to* $maxgen$ **do**

Sample V pairs of candidates (according to ICL_{ex} rank)

for *each pair* $(\boldsymbol{Z}^1, \boldsymbol{Z}^2)$ **do**

- Recombination: $\boldsymbol{Z} = \mathrm{Cross}(\boldsymbol{Z}^1, \boldsymbol{Z}^2)$ (cross-over)
- Use greedy local search on $\boldsymbol{Z}$ (**merges**)
- Apply random splits to $\boldsymbol{Z}$ with proba pm (**mutation**)
- If a random split occurs use greedy local search on $\boldsymbol{Z}$ (**swaps**)

end

end

Crossover operator: cross-partition operator

- Solution space has a particular structure (must handle label switching)
- Integer encoding with single point crossover not well suited for the task

$\Rightarrow$ Work on the space of partitions $\boldsymbol{Z} \Leftrightarrow \mathcal{P} = \{\boldsymbol{C}_1, ..., \boldsymbol{C}_K\}$ a partition of $[n]$.
This space has an interesting operator the cross-partition operator

Cross-partition operator

Let $\mathcal{P}^1 = \{\boldsymbol{C}_1^1, ..., \boldsymbol{C}_{K_1}^1\}$ and $\mathcal{P}^2 = \{\boldsymbol{C}_1^2, ..., \boldsymbol{C}_{K_2}^2\}$ be two partition of $[n]$ the cross-partition operator $\times$ is defined by:

$$\mathcal{P}^1 \times \mathcal{P}^2 := \left\{\boldsymbol{C}_k^1 \cap \boldsymbol{C}_l^2 \,, \forall k \in \{1, ..., K_1\}, \forall l \in \{1, ..., K_2\}\right\} \setminus \{\emptyset\} \,.$$

Crossover operator: cross-partition operator

- Solution space has a particular structure (must handle label switching)
- Integer encoding with single point crossover not well suited for the task

$\Rightarrow$ Work on the space of partitions $\boldsymbol{Z} \Leftrightarrow \mathcal{P} = \{\boldsymbol{C}_1, ..., \boldsymbol{C}_K\}$ a partition of $[n]$.

This space has an interesting operator the cross-partition operator

Cross-partition operator

Example : $\mathcal{P}^1 = \Big\{\{1,2,3\},\{4,5,6,7,8,9\}\Big\}$ and $\mathcal{P}^1 = \Big\{\{1,2,3,4,5,6\},\{7,8,9\}\Big\}$:

$$\mathcal{P}^1 \times \mathcal{P}^2 = \Big\{\{1,2,3\},\{4,5,6\},\{7,8,9\}\Big\}$$

Crossover operator: cross-partition operator

- Solution space has a particular structure (must handle label switching)
- Integer encoding with single point crossover not well suited for the task

$\Rightarrow$ Work on the space of partitions $\boldsymbol{Z} \Leftrightarrow \mathcal{P} = \{\boldsymbol{C}_1, ..., \boldsymbol{C}_K\}$ a partition of $[n]$.
This space has an interesting operator the cross-partition operator

Cross-partition operator

Property:

$\mathcal{P}^1 \times \mathcal{P}^2$ is the coarsest refinement of $\mathcal{P}^1$ and $\mathcal{P}^2$ (both parents may be reconstructed using merge operations).

Crossover operator: cross-partition operator

- Solution space has a particular structure (must handle label switching)
- Integer encoding with single point crossover not well suited for the task

$\Rightarrow$ Work on the space of partitions $\boldsymbol{Z} \Leftrightarrow \mathcal{P} = \{\boldsymbol{C}_1, \ldots, \boldsymbol{C}_K\}$ a partition of $[n]$.
This space has an interesting operator the cross-partition operator

Cross-partition operator

The cross-partition operator is well suited to define a crossover operator

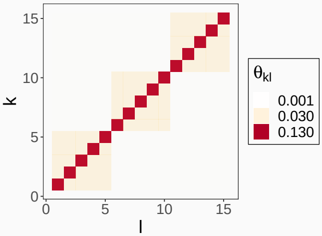
- If both parent partitions are **under-fitted**, crossing them allows the algorithm to go backward (and in the good direction) in the partition lattice, considering finer clustering.
- Synergy with greedy merge operations done afterwards to avoid **over-fitting**

Cross-partition: an illustration

Hierarchical nested SBM with $K = 15$ and $n = 1500$

$$y_{ij} \mid z_{ik} z_{jl} = 1 \sim \mathcal{B}(\theta^\star_{kl}),$$

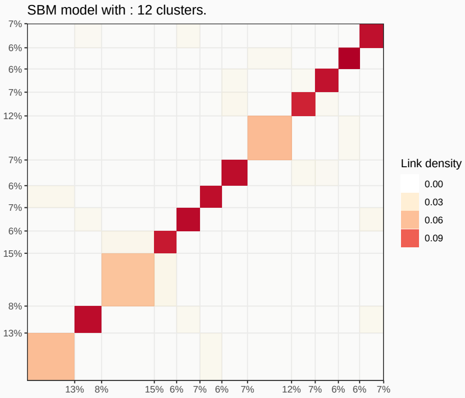
$\boldsymbol{\theta}^\star =$




```
> sbm = rsbm(n, Pi, Theta)
> fit = greed(sbm$x, model=Sbm())
```

Cross-partition: an illustration

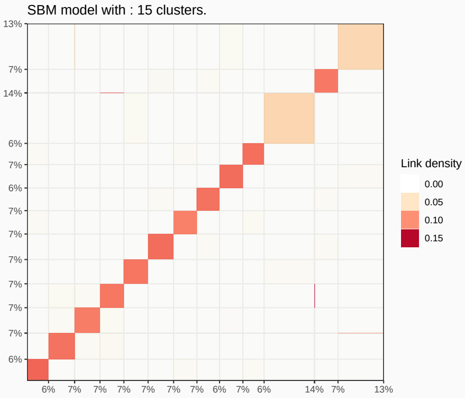
Solution $\mathcal{P}^1$ is a local optimum after greedy swap :



Pb: under-fitting, local swap optima

Cross-partition: an illustration

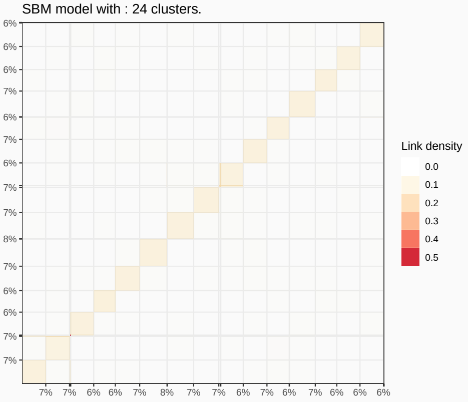
Solution $\mathcal{P}^2$ is another local optimum after greedy swap :



Pb: under-fitting (and over-fitting), local swap optima

Cross-partition: an illustration

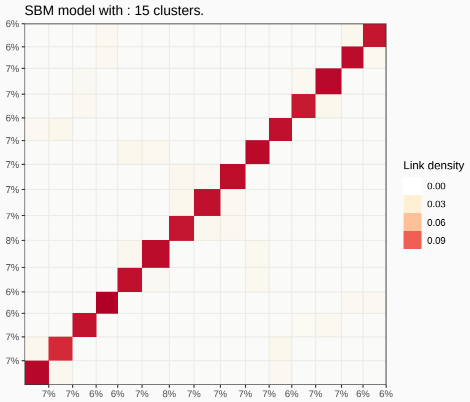
Solution $\mathcal{P}^1 \times \mathcal{P}^2$:



Pb: over-fitting

Crossover operator: cross-partition operator

Solution $\mathcal{P}^1 \times \mathcal{P}^2$ + **greedy merge**:



Simulated partition is recovered

A sketch of the hierarchical algorithm

Goal: hierarchy construction from $\boldsymbol{Z}^{(K^\star)}$

- access to "simpler" partitions and highlight relationship between clusters
- useful E.D.A tools: dendogram, cluster ordering,...

Standard agglomerative method: starts from $\boldsymbol{Z}^{(K^\star)}$

- At stage k, find the best fusion w.r.t ICL_{ex}. Repeat until $k = 1$

A sketch of the hierarchical algorithm

Goal: hierarchy construction from $\boldsymbol{Z}^{(K^\star)}$

- access to “simpler” partitions and highlight relationship between clusters
- useful E.D.A tools: dendogram, cluster ordering,...

Standard agglomerative method: starts from $\boldsymbol{Z}^{(K^\star)}$

- At stage k, find the best fusion w.r.t ICL_{ex}. Repeat until $k = 1$

Problem: fusions are not possible in term of ICL_{ex}

Solution:

- Use α hyper-parameter as a regularization parameter
- Extract a set of dominating *nested* partitions

A quick introduction to the *greed* package

Implementation and API

greed is available on CRAN and it is

- Flexible - can handle categorical, count, continuous, graphs or a combination
- Quick - especially for networks

Implementation and API

greed is available on CRAN and it is

- Flexible - can handle categorical, count, continuous, graphs or a combination
- Quick - especially for networks

Main usage via the `greed()` function

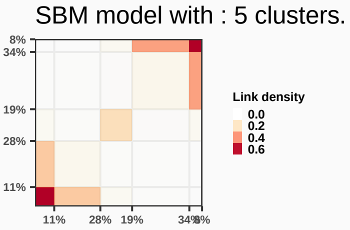
```
> sol <- greed(X,model)
> cl <- clustering(sol)
```

Many generic functionalities such as

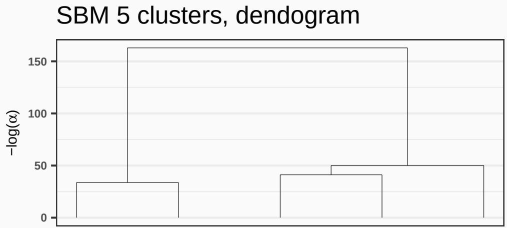
- **Plot** `> plot(sol, type="tree")`
- **Explore** `> sol_K2 <- cut(sol, K=2)`
- **MAP estimate** $\rightarrow \hat{\theta} \mid Z^{(K^\star)}$ `> theta <- coef(sol)`

Graph clustering: the book dataset

```
> sol_sbm = greed(Books$X, model=Sbm())
> plot(sol_sbm, type)
```



`type="blocks"`

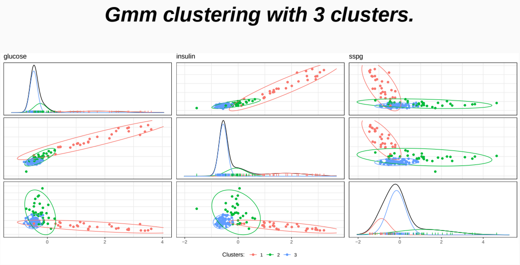


`type="tree"`

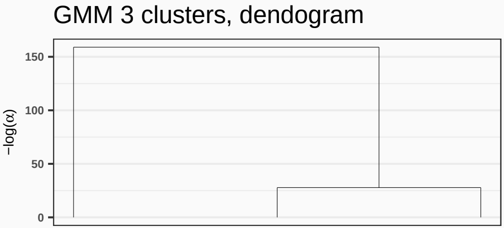
Continuous data: diabetes data

```
> sol_gmm = greed(X, model=Gmm())
```

	1	2	3
Chemical	11	24	1
Normal	73	3	0
Overt	0	6	27



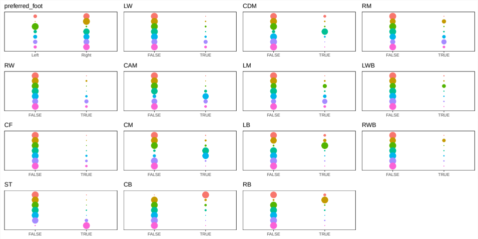

```
gmmpairs(sol_gmm, X)
```



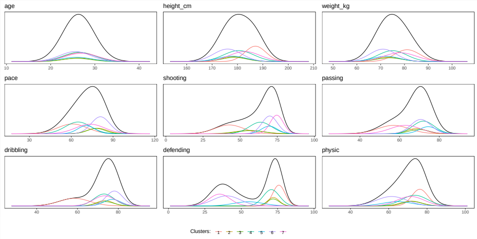

```
plot(sol_gmm, "tree")
```

Heterogeneous data: the fifa dataset

```
> X = list(cat=Xcat, num=Xnum)
> mods <-list(cat=LcaPrior(), num=GmmPrior())
> sol_cb = greed(X, model=CombinedModels(mods))
> submod = extractSubModel(sol, name)
> plot(submod, type="marginals")
```



`name="cat"`



`name="num"`

Conclusion

Model-based approach for clustering and hierarchical clustering

Pros

- Applies to a wide range of data, *e.g.* counts, categorical or graphs
- Handles heterogeneous data
- Efficient algorithms relying on greedy heuristics (bypass inference)
- Uses **random** initializations
- Article also covers the co-clustering case with Latent Block Models

Cons

- Cannot fix the desired number of clusters $K^\star$.
- Needs an **exact** ICL

If you are interested:

- Journal article available here (published in *ADAC*)
- Implementation details and package description here (submitted to *JSS*)

Thank you for your attention !

Questions ?

References

- Bertoletti, Marco, Nial Friel, and Riccardo Rastelli (Aug. 2015). "Choosing the number of clusters in a finite mixture model using an exact integrated completed likelihood criterion". In: *METRON* 73.2, pp. 177–199.
- Biernacki, Christophe, Gilles Celeux, and Gérard Govaert (2000). "Assessing a mixture model for clustering with the integrated completed likelihood". In: *IEEE transactions on pattern analysis and machine intelligence* 22.7, pp. 719–725.
- Biernacki, Christophe, Gilles Celeux, and Gerard Govaert (2010). "Exact and monte carlo calculations of integrated likelihoods for the latent class model". In: *Journal of Statistical Planning and Inference* 140, pp. 2991–3002.
- Côme, Etienne and Pierre Latouche (2015). "Model selection and clustering in stochastic block models based on the exact integrated complete data likelihood". In: *Statistical Modelling* 15.6, pp. 564–589.

Qin, Tai and Karl Rohe (2013). "Regularized Spectral Clustering under the Degree-Corrected Stochastic Blockmodel". In: *Proceedings of Nips*.

Tessier, Damien et al. (2006). "Evolutionary latent class clustering of qualitative data". In.

Appendix

Combined models

Combined models

Context

- V *views* of the data (*e.g.* Multiplex networks, heterogeneous data)
- $\boldsymbol{X} = \{\boldsymbol{X}_v\}_{v=1,\dots,V}$ — X_v is the v-th views of the data

Stack observational models $\{\mathcal{M}_v\}$ with conditional independence assumption

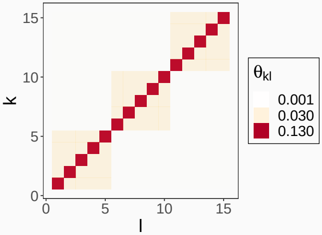
$$p(\boldsymbol{X}_1, \dots, \boldsymbol{X}_V \mid \boldsymbol{Z}) = p(\boldsymbol{X}_1 \mid \mathcal{M}_1, \boldsymbol{Z}) \times \dots \times p(\boldsymbol{X}_V \mid \mathcal{M}_V, \boldsymbol{Z}).$$

ICL_{ex} of the whole dataset is simply the sum of the submodels ICL_{ex}

Experimental results: medium-scale SBM

Hierarchical nested SBM with $K = 15$ and $n = 1500$

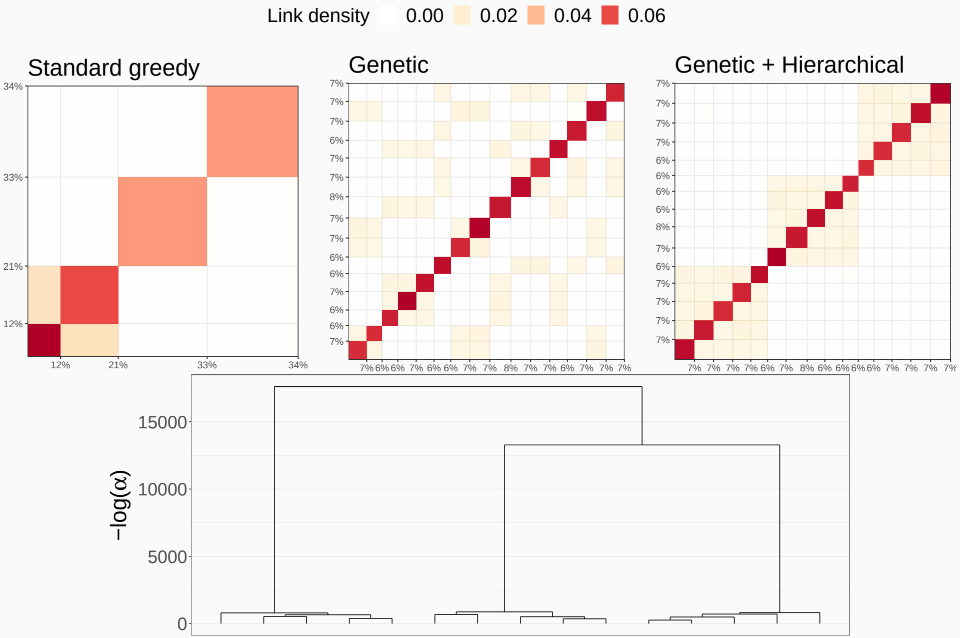
$$y_{ij} \mid z_{ik} z_{jl} = 1 \sim \mathcal{B}(\theta^\star_{kl}), \qquad \theta^\star =$$




```
> sbm = rsbm(n,Pi,Theta)
> fit = greed(sbm$x,model=new("sbm"),alg=new("hybrid",pop_size=40
```

Compare with (implemented in the package)

- Spectral clustering (Qin et al. 2013)
- Greedy local search: unique / multiple / spectral initializations



Link density 0.00 0.02 0.04 0.06
Standard greedy
Genetic
Genetic + Hierarchical
−log(α)
15000
10000
5000
0

Hierarchical model-based clustering in DLVMs

A novel approximation for the ICL_{ex}

$$\mathrm{ICL}_{ex}(\boldsymbol{Z}, \alpha) = D(\boldsymbol{Z}) + \log p(\boldsymbol{Z} \mid \alpha), \quad \log p(\boldsymbol{Z} \mid \alpha) = \log \frac{\Gamma(K\alpha) \prod_k \Gamma(\alpha + n_k)}{\Gamma(\alpha)^K \Gamma(n + \alpha K)}$$

Our proposition: asymptotic of $\log \Gamma$ near 0

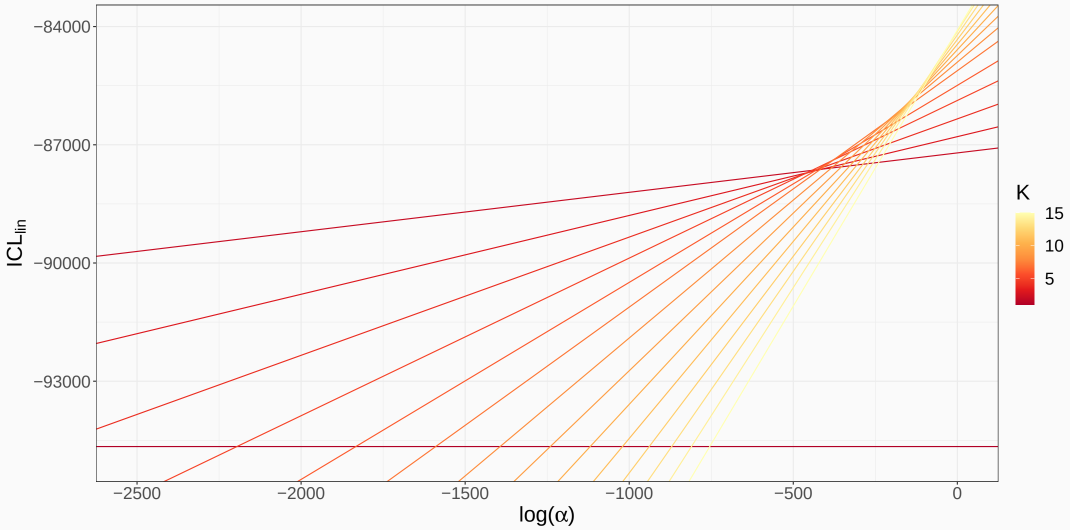
$$\log \Gamma(\alpha) \underset{\alpha \to 0}{\sim} -\log(\alpha)$$

Log-linear ICL

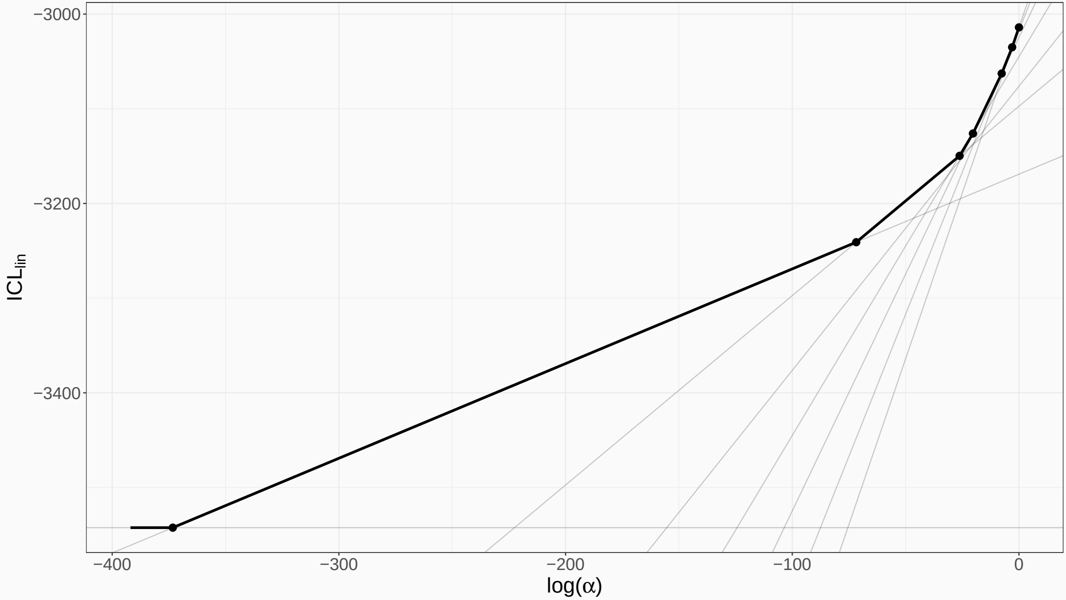
$$\mathrm{ICL}_{lin}(\boldsymbol{Z}^{(k)}, \alpha) := (k-1)\log(\alpha) + I(\boldsymbol{Z}^{(k)})$$

$$I(\boldsymbol{Z}^{(k)}) = D(\boldsymbol{Z}^{(k)}) + \sum_{l=1}^{k} \log \Gamma(n_l) - \log \Gamma(n) - \log(k)$$

ICL_{lin} as lines of increasing slope with K



A discrete Pareto frontier



Fusion opportunity at stage (k)

Fixed partition $\boldsymbol{Z}^{(k)}$ with k clusters

Two clusters (g, h): ICL_{lin} change for $g \cup h$?

$$\Delta_{g\cup h}(\alpha) = \mathrm{ICL}_{lin}\left(\boldsymbol{Z}^{(k)}_{g\cup h}, \alpha\right) - \mathrm{ICL}_{lin}\left(\boldsymbol{Z}^{(k)}, \alpha\right)$$

Proposition

$$\forall g \neq h,\ \Delta_{g\cup h}(\alpha) > 0 \iff \log(\alpha) < I(\boldsymbol{Z}^{(k)}_{g\cup h}) - I(\boldsymbol{Z}^{(k)})$$

Regularization parameter: α unlocks fusions

Question: $k(k-1)/2$ fusions, which one is the best ?

$$(g^\star, h^\star) = \arg\max_{g,h} I(\boldsymbol{Z}^{(k)}_{g\cup h})$$

Hierarchy construction and dendrogram representation

Repeat procedure at each stage $\boldsymbol{Z}^{(k)}$

$$\log \alpha^{(k)} := I(\boldsymbol{Z}^{(k)}_{g^\star \cup h^\star}) - I(\boldsymbol{Z}^{(k)})$$

Outputs a hierarchy of partitions

Dendrogram representation:

- $\alpha^{(k)}$ is the amount of regularization needed for the fusion
- Extract a front of dominating partitions on range $[\alpha^{(k-1)}, \alpha^{(k)}]$