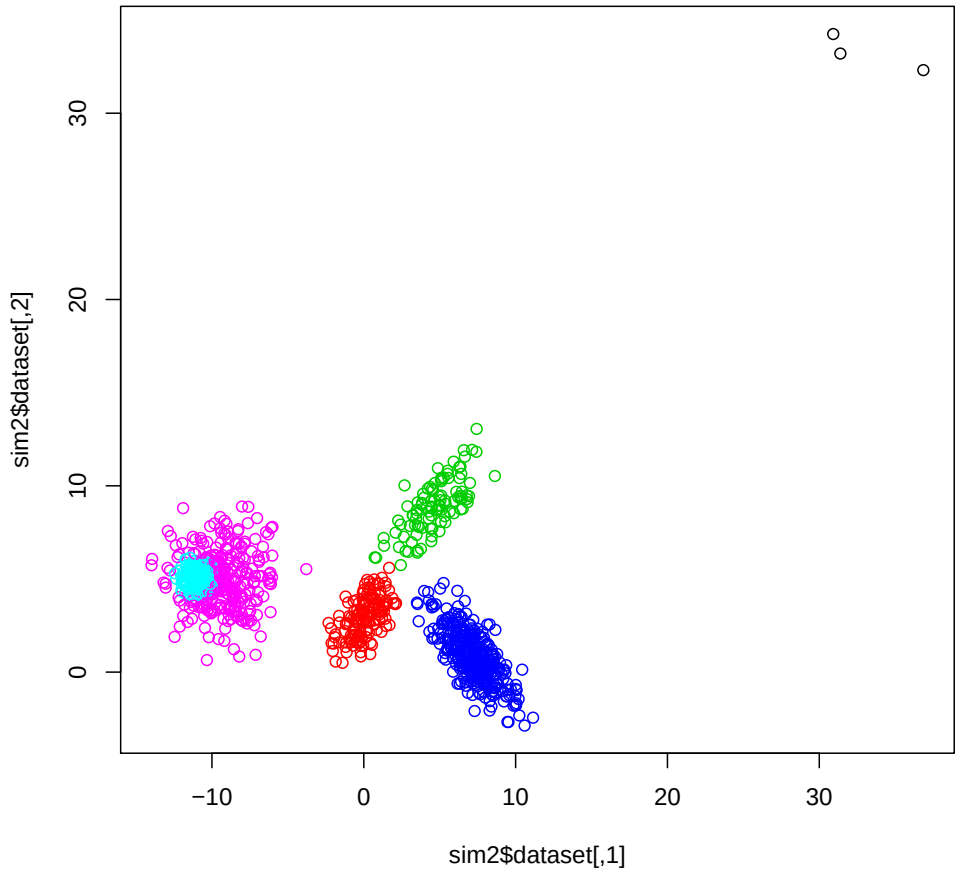
UCL

Gaussian and not-so-Gaussian clustering with robustness against outliers and a stab at the number of clusters

Christian Hennig and Pietro Coretto

1.1 Introduction - Challenges to Gaussian clustering



Standard Gaussian model-based clustering

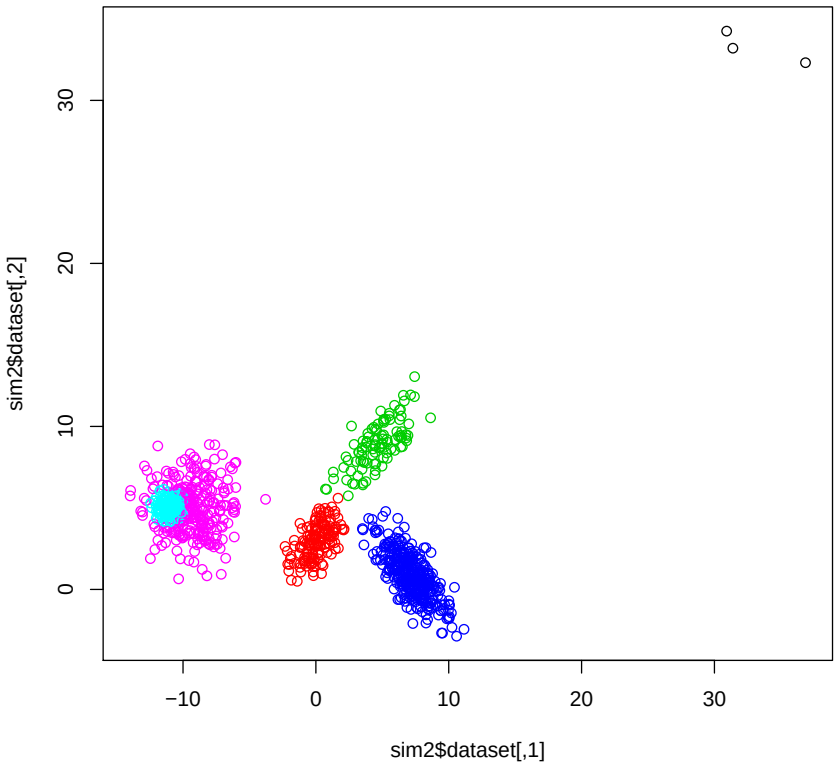
$$f(x) = \sum_{j=1}^{G} \pi_j \varphi_{\mathbf{a}_j, \Sigma_j}(x)$$

Compute $\hat{\pi}_j, \hat{\mathbf{a}}_j, \hat{\Sigma}$ by ML/EM-algorithm, classify points by

$$\hat{\gamma}(i) = \arg\max_k \frac{\hat{\pi}_k \varphi_{\hat{\mathbf{a}}_k, \hat{\Sigma}_k}(x_i)}{\sum_{j=1}^{G} \hat{\pi}_j \varphi_{\hat{\mathbf{a}}_j, \hat{\Sigma}_j}(x)}.$$

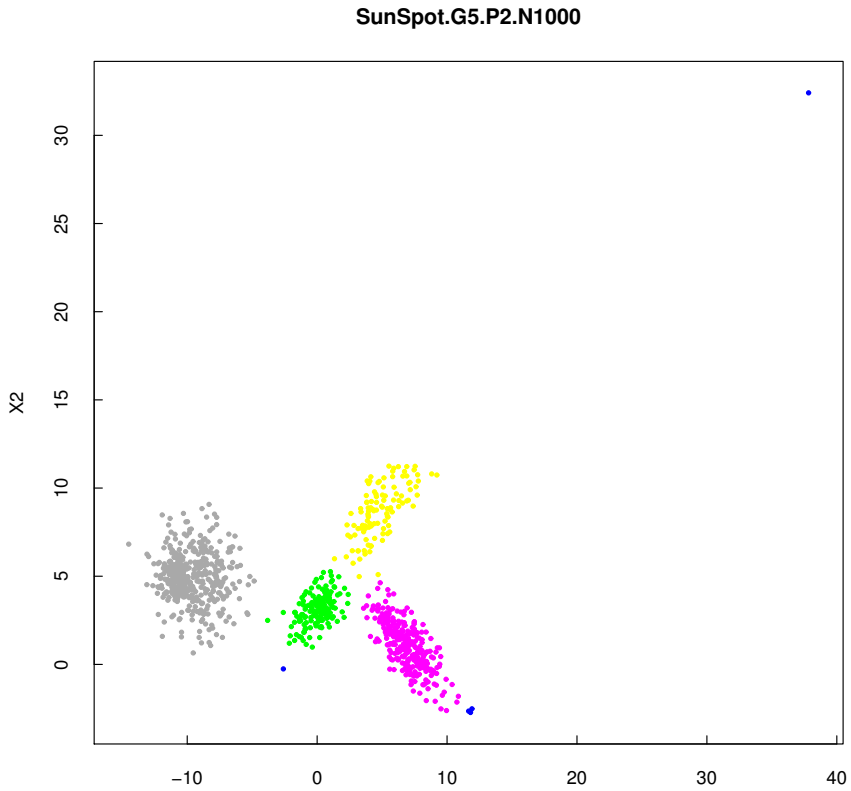
(Bayes rule, used for all mixture-based methods.)
R-mclust package, Fraley and Raftery

Estimating G by BIC (mclust) gives $\hat{G} = 6$ and this...



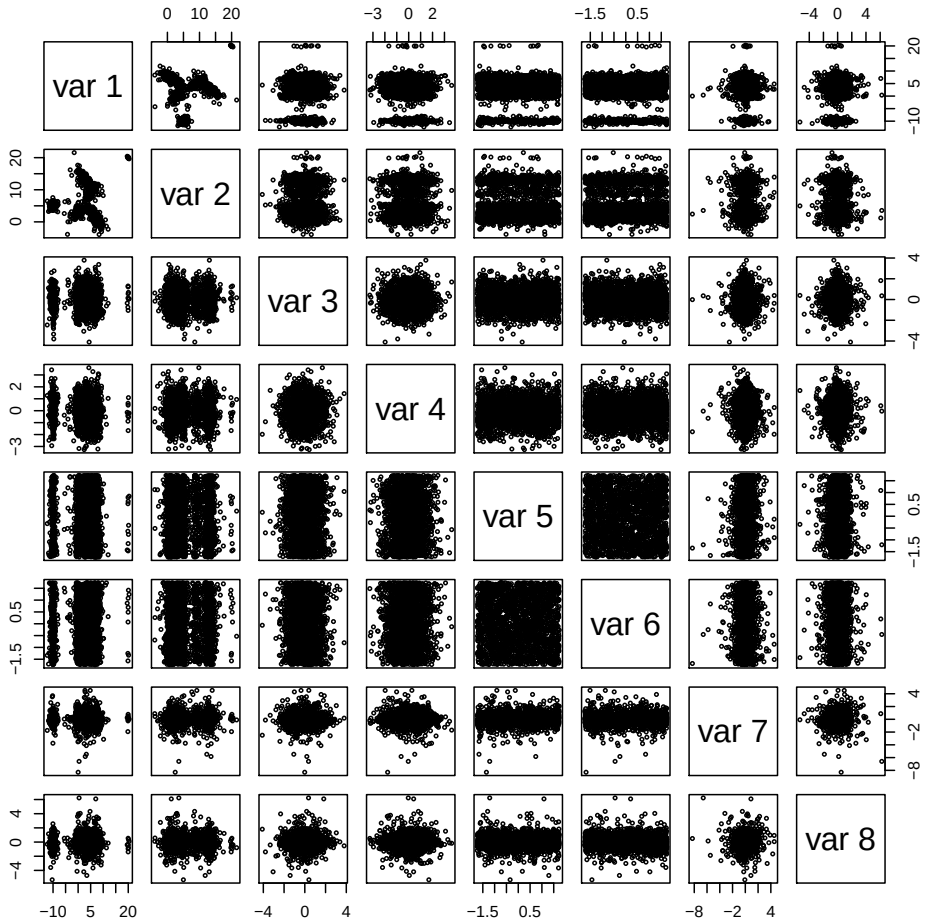
...which is actually fine.

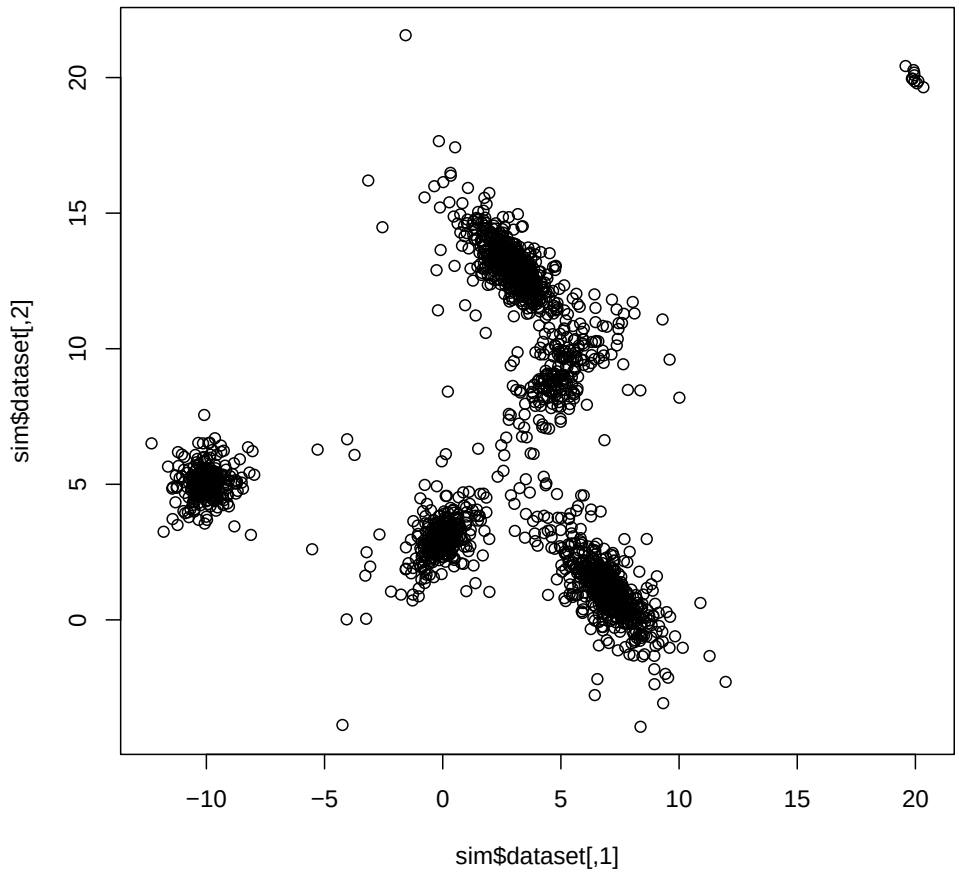
With only one outlier, get $\hat{G} = 5$
and a different cov-matrix model.

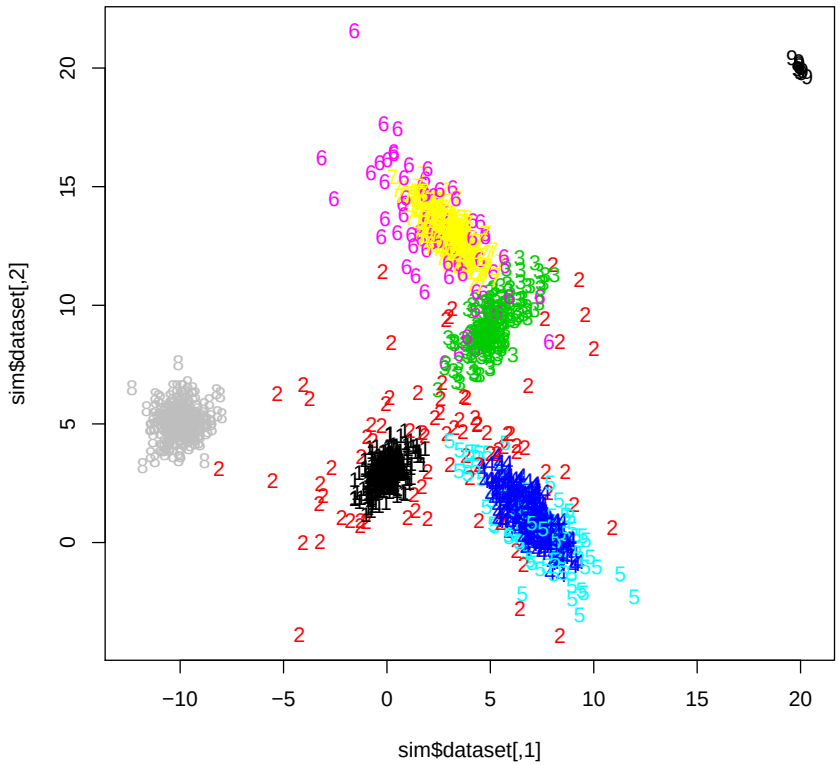


Gaussian clusters and extreme uniform on P=1,2; standard Gaussian on P>2

In reality, we apply Gaussian clusters
to non-Gaussian data.
Want methods that give us something useful
even if clusters are not exactly Gaussian.

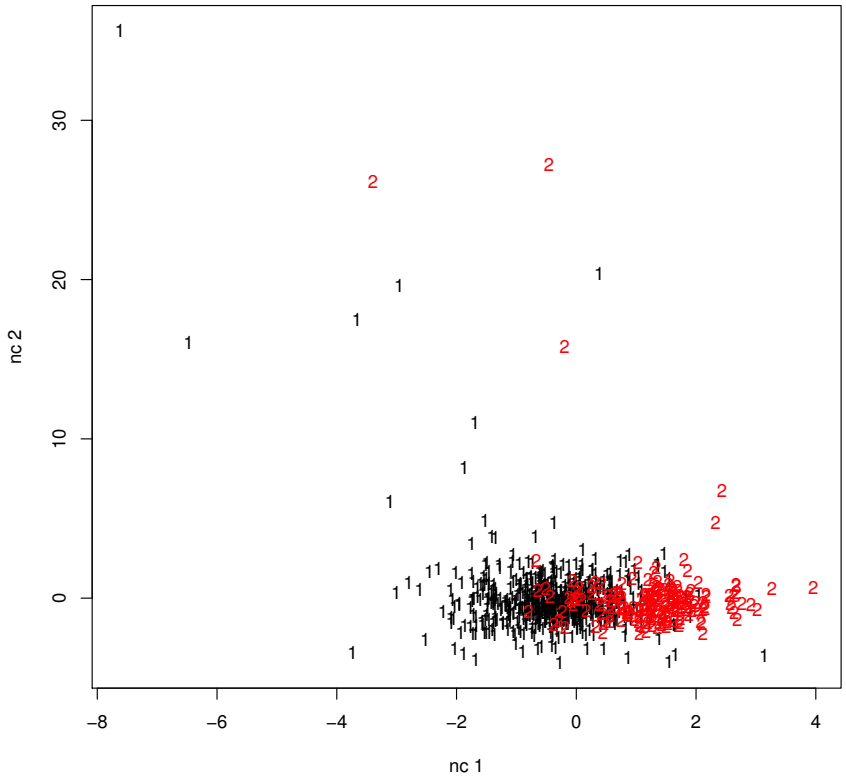

var 1
var 2
var 3
var 4
var 5
var 6
var 7
var 8


sim$dataset[,2]
sim$dataset[,1]
20
15
10
5
0
−10
−5
0
5
10
15
20



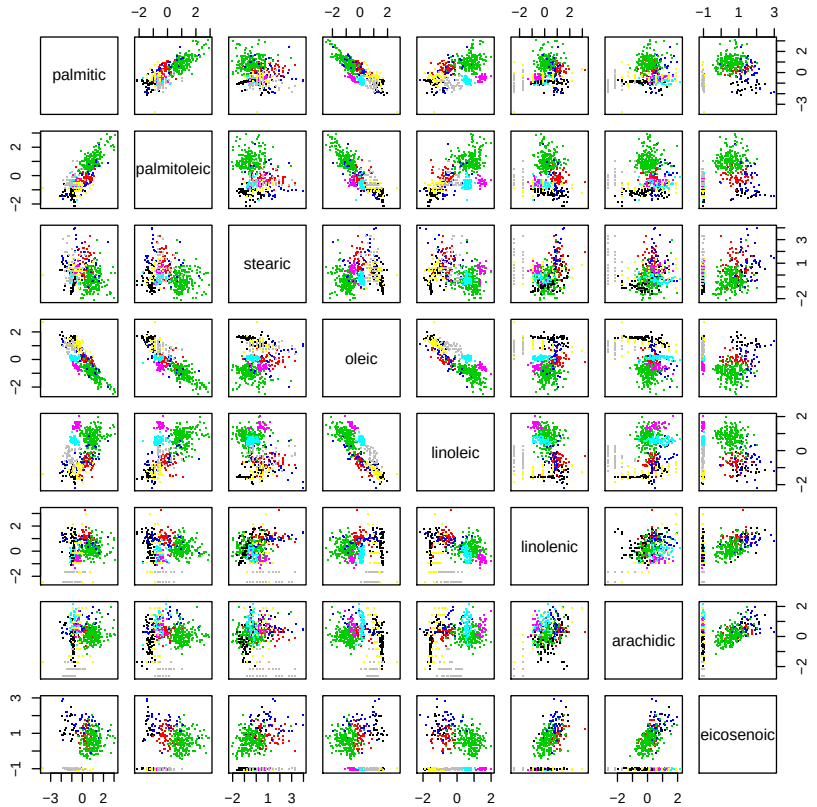
mclust approximates some t-distributions by two Gaussians, a bit messy.

776 folk songs from Luxemburg and Warmia, 18 features



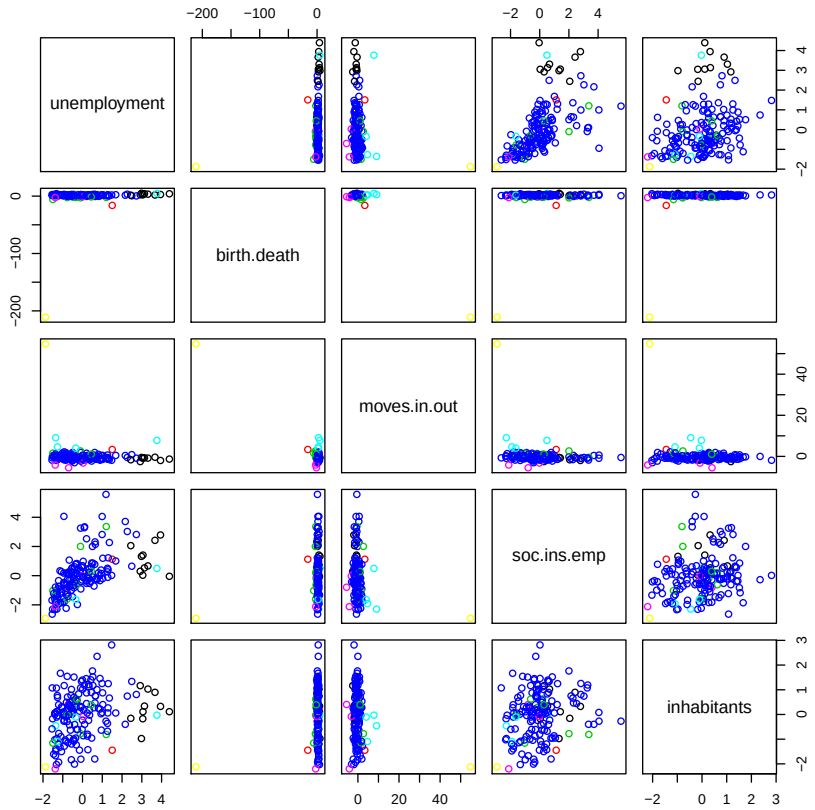
BIC: $G = 2$, but ARI ≈ 0.

572 olive oils from 9 Italian regions, 8 features



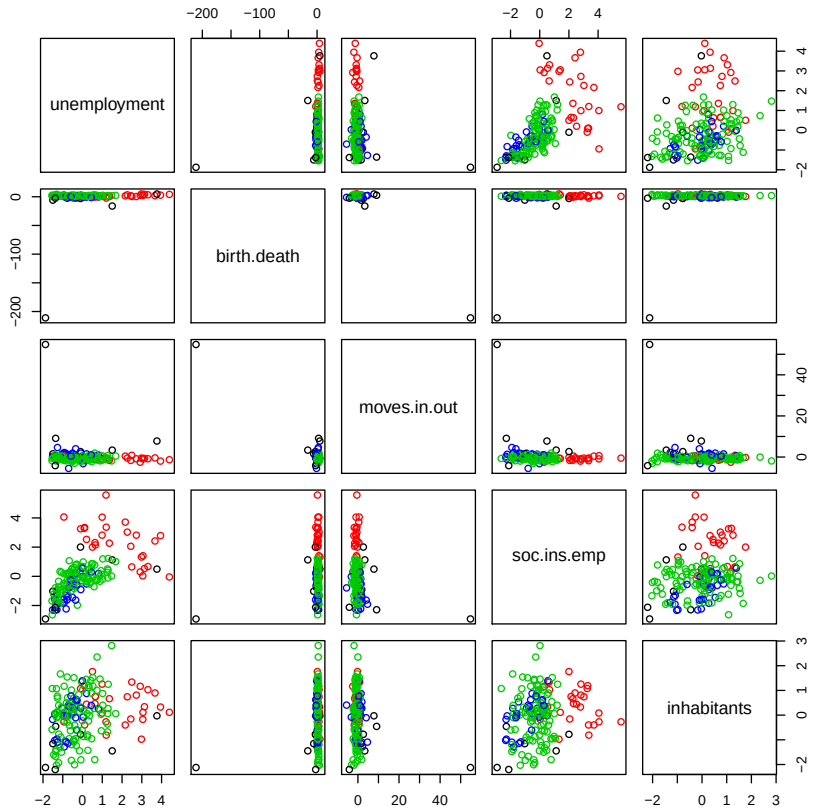
Note discreteness of some variables!

170 districts of city of Dortmund



mclust gives main bulk and various outlier classes.

This is what OTRIMLE gives:



(Black observations are classified as “noise/outliers”.)

1.2 Gaussian mixtures, non-Gaussian data

Why fit Gaussian mixtures?
We don't believe that clusters are really Gaussian,
but we look for clusters
for which the Gaussian shape is a suitable "prototype".

We look for clusters
for which the Gaussian shape is a suitable “prototype”.

What does this mean?
That’s not so clear.
Tolerate skewness, nonlinearity, heavy tails?
Certainly not all of these!

We look for clusters
for which the Gaussian shape is a suitable “prototype”.

What does this mean?
That’s not so clear.
Tolerate skewness, nonlinearity, heavy tails?
Certainly not all of these!

Methods that classify observations as “noise/outlier”
can cut out “near-Gaussian cores”.

Could use other “cluster prototypes”,
mixtures of skew and heavy-tailed distributions,
density mode/level set based clustering.

Could use other “cluster prototypes”,
mixtures of skew and heavy-tailed distributions,
density mode/level set based clustering.

These come with their own problems
(high-d density estimation is hard;
is it really appropriate to integrate
outliers into heavy tailed “cluster”?)

Could use other “cluster prototypes”,
mixtures of skew and heavy-tailed distributions,
density mode/level set based clustering.

These come with their own problems
(high-d density estimation is hard;
is it really appropriate to integrate
outliers into heavy tailed “cluster”?)

Ultimately the user needs to decide
what “clustering” means in an application.

Guiding idea:

Impose Gaussian clusters on non-Gaussian data
for which Gaussian clusters are still “adequate”.

Guiding idea:

Impose Gaussian clusters on non-Gaussian data
for which Gaussian clusters are still "adequate".

The "number of clusters"-problem is central here;
Every continuous distribution can be approximated
by enough Gaussian mixture components,
but these won't always make reasonable "clusters".

Guiding idea:

Impose Gaussian clusters on non-Gaussian data
for which Gaussian clusters are still "adequate".

The "number of clusters"-problem is central here;
Every continuous distribution can be approximated
by enough Gaussian mixture components,
but these won't always make reasonable "clusters".

In cluster analysis, we want a sufficiently small
number of well distinguished clusters.

1.3 The robust Gaussian clustering problem

. . . *oversimplified:* fit

$$f(x) = \sum_{j=1}^{G} \pi_j \varphi_{\mathbf{a}_j, \Sigma_j}(x)$$

and assign points to mixture components in a "robust" way,
not too affected by points not fitted by model,
or by slight deviation of components from φ.

1.4 Existing methods

- Plain Gaussian mixture
- Mixture of t-distributions (McLachlan & Peel 2000)
- Trimmed clustering (Garcia-Escudero et al. 2008, Gallegos & Ritter 2005)
- Gaussian mixture with "noise component" (Banfield & Raftery 1993)
- Trimmed likelihood (Neykov et al. 2007)
- More elaborate mixtures (skew-t etc.)

2 Optimally tuned robust improper ML (OTRIMLE)

(Coretto and Hennig, JASA 2016)

2.1 Robust improper ML (RIMLE)

Use improper fixed density for "noise"
(inspired by Banfield & Raftery's "noise component").
Fit "pseudo-density" by "pseudo-ML/EM"

$$\psi_c(x,\theta) = f(x) = \pi_0 c + \sum_{j=1}^{G} \pi_j \varphi_{\mathbf{a}_j,\Sigma_j}(x),$$

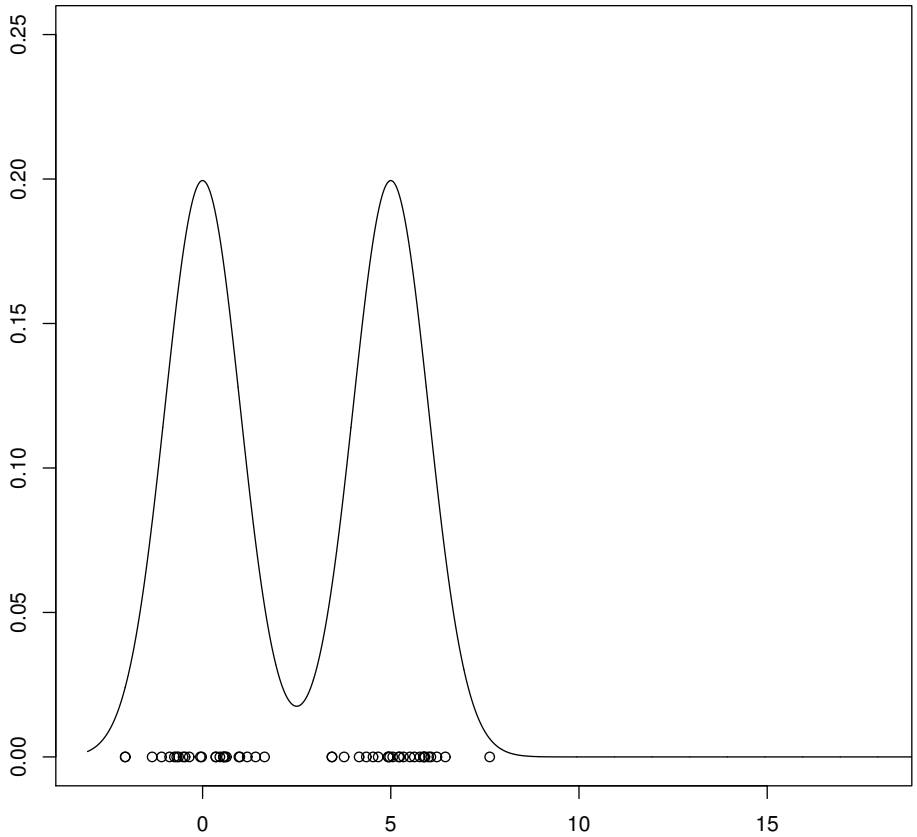
with tuning constant c.
(c is not a model parameter;
$c \to \infty \Rightarrow$ likelihood $\to \infty$.)

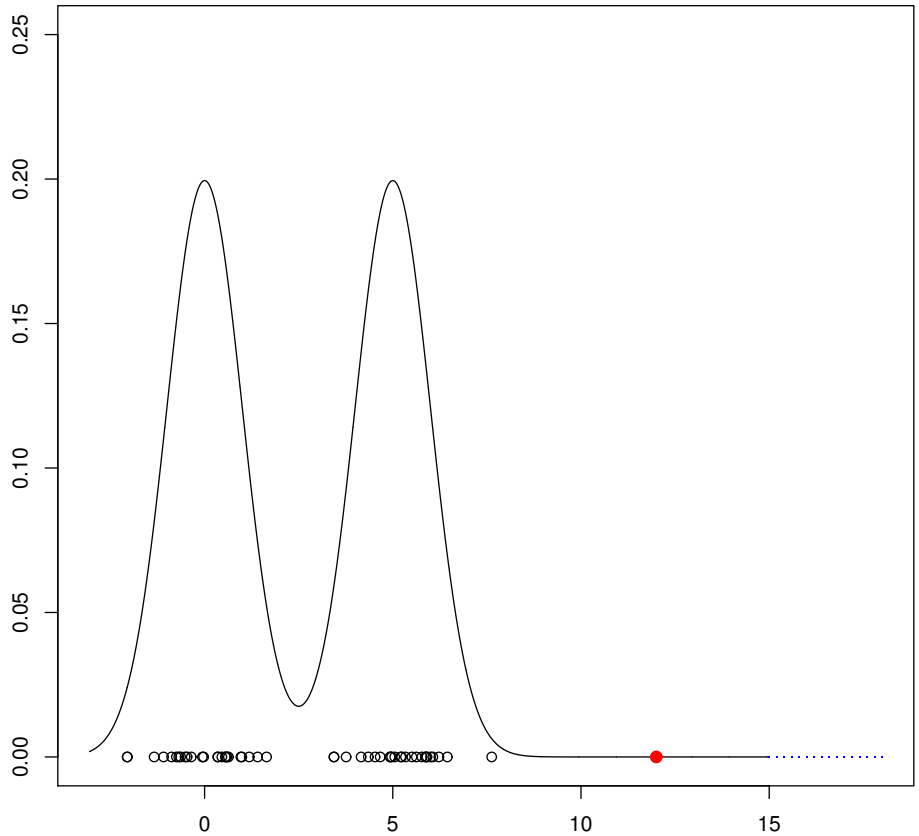
Pseudo posterior probabilities:

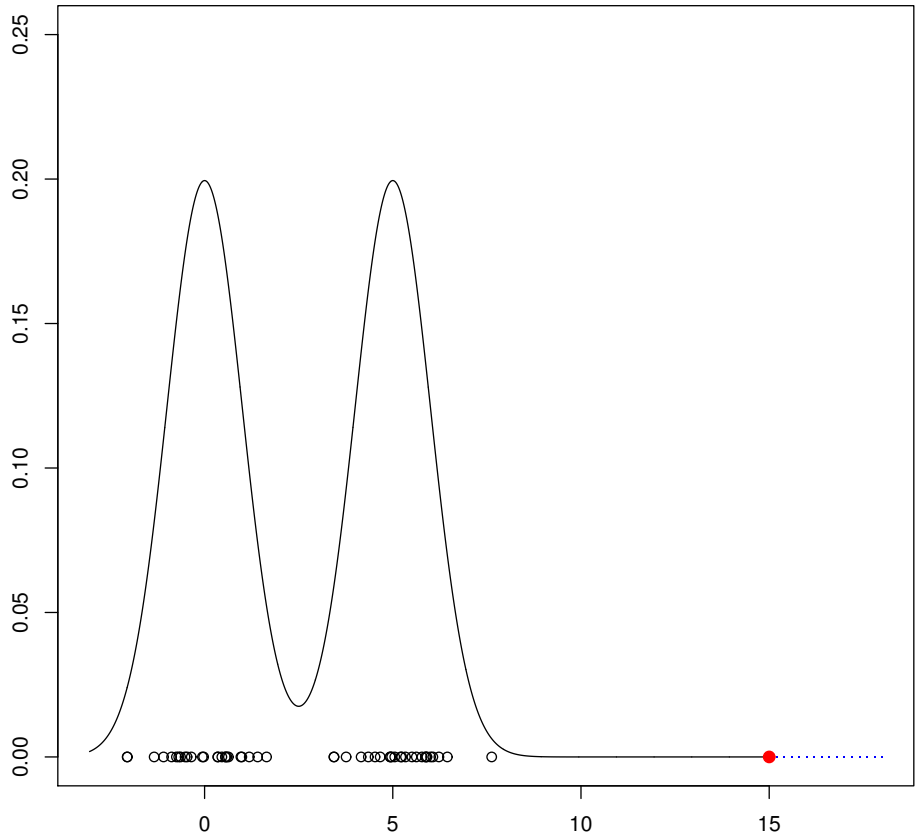
$$\tau_j(x_i,\theta)=\begin{cases}\dfrac{\hat{\pi}_0 c}{\psi_c(x_i,\hat{\theta})} & \text{if } j=0\\ \dfrac{\hat{\pi}_j\phi(x_i,\hat{\mathbf{a}}_j,\hat{\Sigma}_j)}{\psi_c(x_i,\hat{\theta})} & \text{if } j=1,2,\ldots,G\end{cases}$$

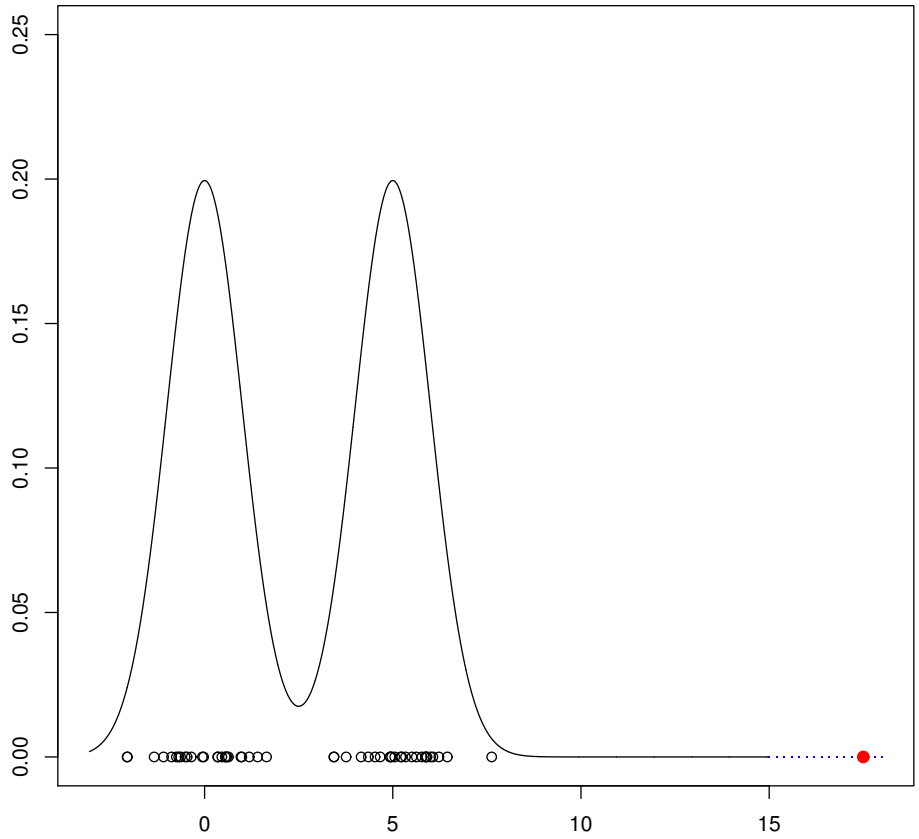
Cluster assignment by maximum posterior.

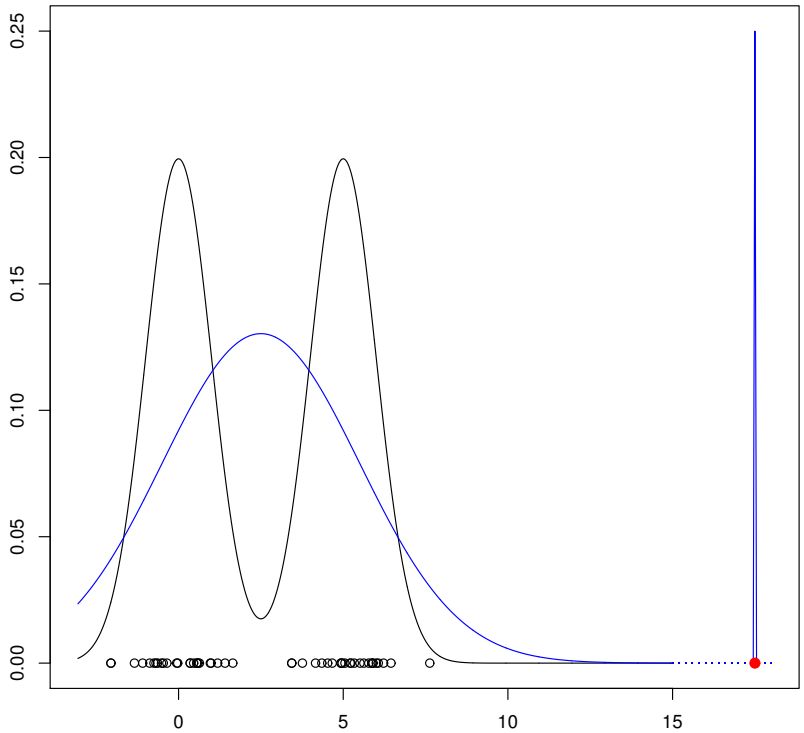
Robustness in 1-d (Hennig 2004)



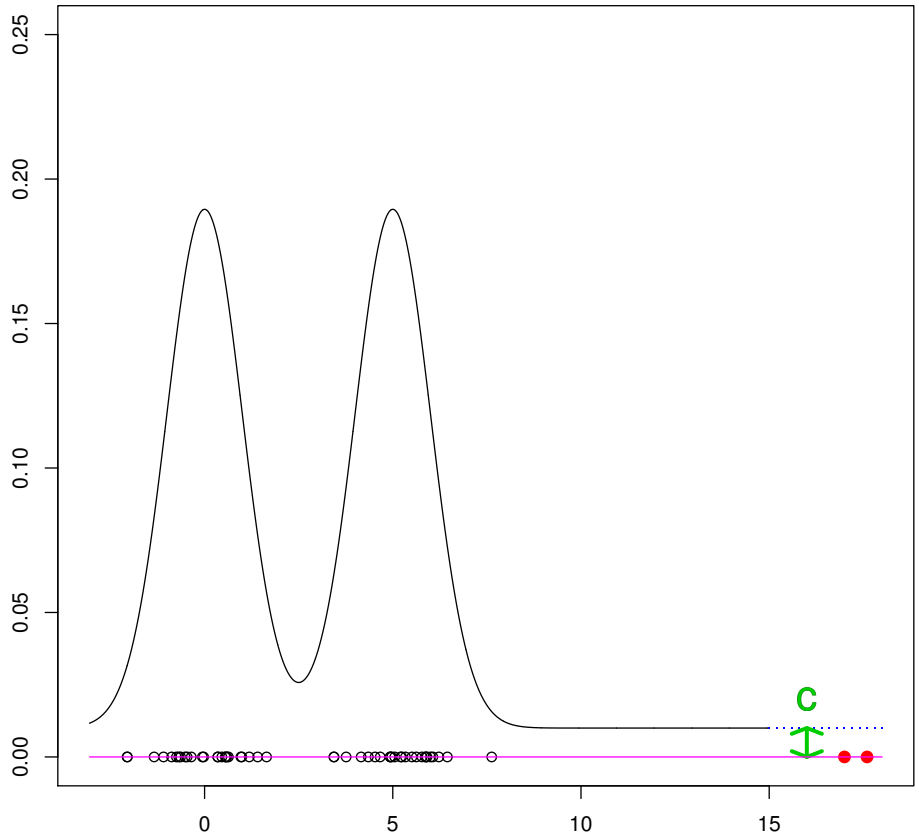

0.25
0.20
0.15
0.10
0.05
0.00
0
5
10
15


0.25
0.20
0.15
0.10
0.05
0.00
0
5
10
15


0.25
0.20
0.15
0.10
0.05
0.00
0
5
10
15



This happens when density $\searrow 0$
(Gaussian, t-mixture, Banfield & Raftery's "noise component")


0.25
0.20
0.15
0.10
0.05
0.00
0
5
10
15
C

2.2 The constrained parameter space

Likelihood degenerates if $\lambda_{min} \to 0$
(problem for all methods).
Hathaway (1985), Garcia-Escudero et al. (2008):

$$\lambda_{\max}(\theta)/\lambda_{\min}(\theta) \leq \gamma < +\infty$$

Too much noise causes trouble for parameters:

$$\frac{1}{n}\sum_{i=1}^{n} \tau_0(x_i, \theta) \leq \pi_{\max},$$

Pseudo-ML in constrained parameter space Θ_G.

2.3 Tuning of c:

Optimal tuning for **RIMLE** (**OTRIMLE**):
Minimising, for $c \in [0, C]$,

$$K_{n,G}(c) := \max_{i=1,2,\ldots,n} \sum_{j=1}^{G} \hat{\pi}_j |\mathbb{M}_j(x_i; \hat{\eta}_n(c)) - \chi^2_p(x_i)|,$$

where, with $\hat{\delta}_{ij}(c)$ Mahalanobis-distance of x_i to comp. j,

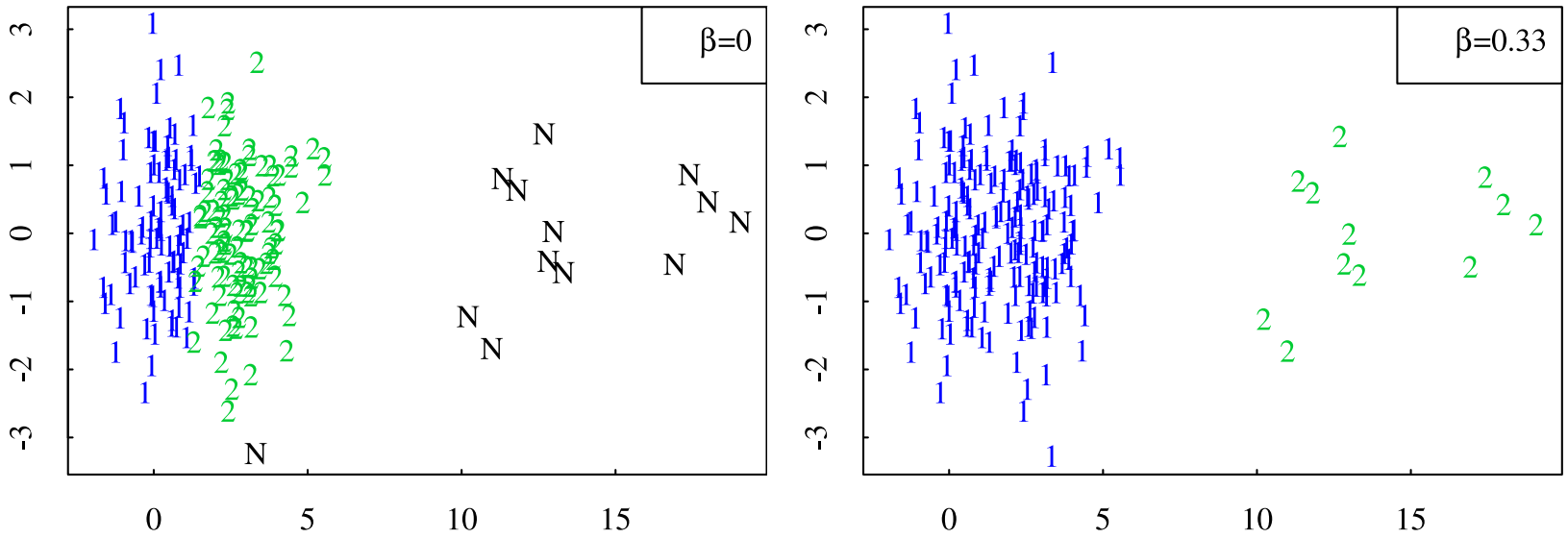
$$\mathbb{M}_j(t; c) = \frac{1}{W_j} \sum_{i=1}^{n} \hat{\tau}_{ij}(c) \mathbf{1}(\hat{\delta}_{ij}(c) \leq t).$$

Idea: try to find c so that the non-outliers look like Gaussian mixture.

Version (**OTRIMLE.P**):
Minimise, for $c \in [0, C]$, with fixed $\beta \geq 0$,

$$K_n(c) + \beta\hat{\pi}_0,$$

to allow some non-normality if this helps to integrate more points into clusters.


β=0
β=0.33
3
2
1
0
-1
-2
-3
0
5
10
15

2.4 Further issues

(Coretto & Hennig 2015, 2016, arxiv)

- Existence and consistency theory
- Breakdown theory
- Pseudo-EM algorithm, convergence, initialisation

2.5 OTRIMLE vs. tclust

tclust (Garcia-Escudero et al., 2008) fits:

$$f(x_1, \ldots, x_n) = \prod_{i \in R} \varphi_{\mathbf{a}_{\gamma(i)}, \Sigma_{\gamma(i)}}(x_i) \prod_{i \notin R} g_i(x_i),$$

$|R| \approx (1 - \alpha)n$.
Tuning: trimming rate α.

tclust is similar to OTRIMLE;
similar benefits and issues;
trimming rate vs. noise pseudo-density level.

tclust advantages:

- Trimming rate easier to interpret (and probably to initialise) than noise level.
- Discrete algorithm is faster.
- Impressive new tclust additions (could be done for OTRIMLE approach).

tclust advantages:

- Trimming rate easier to interpret (and probably to initialise) than noise level.
- Discrete algorithm is faster.
- Impressive new tclust additions (could be done for OTRIMLE approach).

OTRIMLE advantages:

- Smooth mixture classification should often be better and more stable.
- Automatic tuning of noise level (could be applied to trimming rate).

3 The number of clusters G

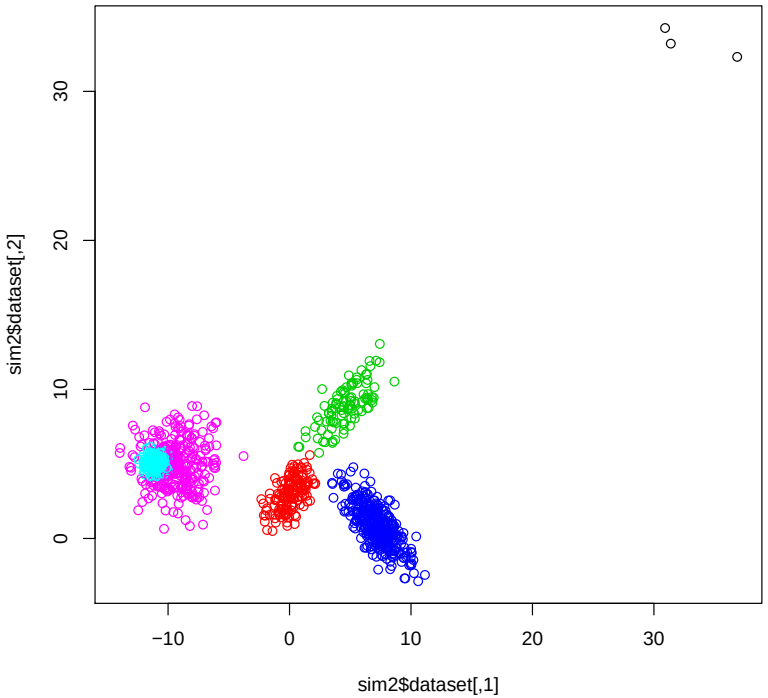
. . . is hardly known in any real application.

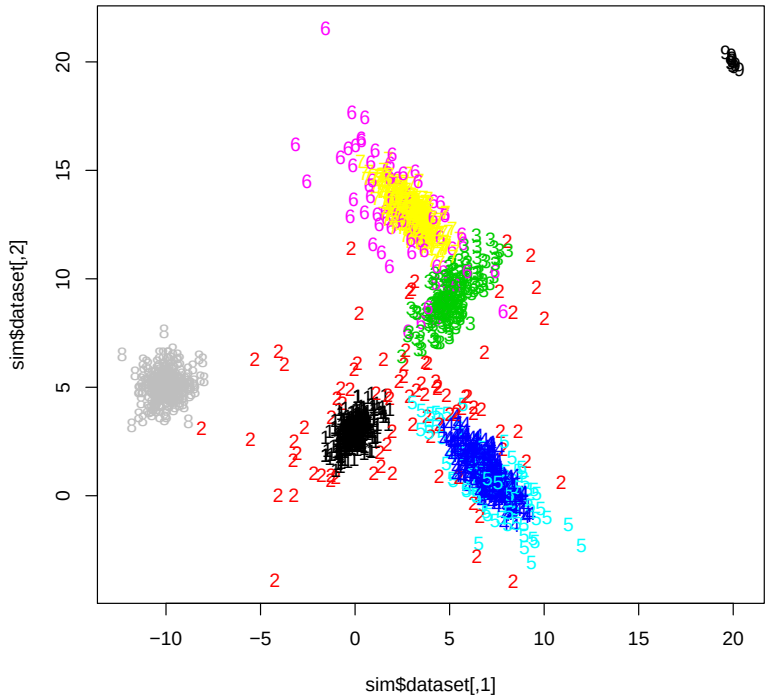
Estimating G (robustly) comes with problems.

1. Estimating number of Gaussian mixture components is ill-posed problem.
   In reality nothing is exactly Gaussian,
   and with large n everything can be fitted
   better with more components.
2. Are "clusters of outliers" clusters or outliers?

Ultimately, estimation of the number of clusters needs user tuning:

- When should something fitted by k mixture components be considered one cluster?
- How big and clear a group of outliers is to be considered a cluster?





Tuning-free methods such as BIC and $K_{n,G}$
rely strongly on model assumptions
and will with larger n
deliver ever larger G to fit non-Gaussian data better.

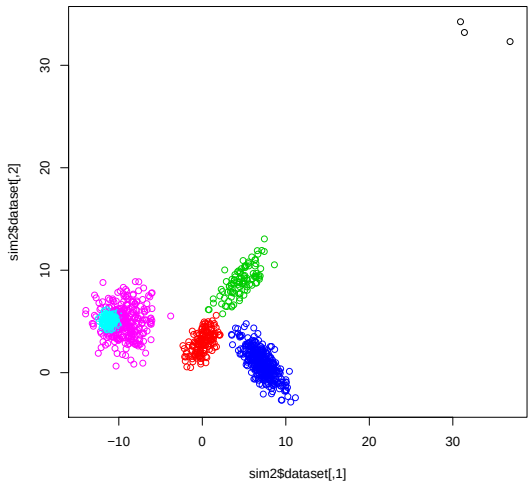
Tuning-free methods such as BIC and $K_{n,G}$
rely strongly on model assumptions
and will with larger n
deliver ever larger G to fit non-Gaussian data better.

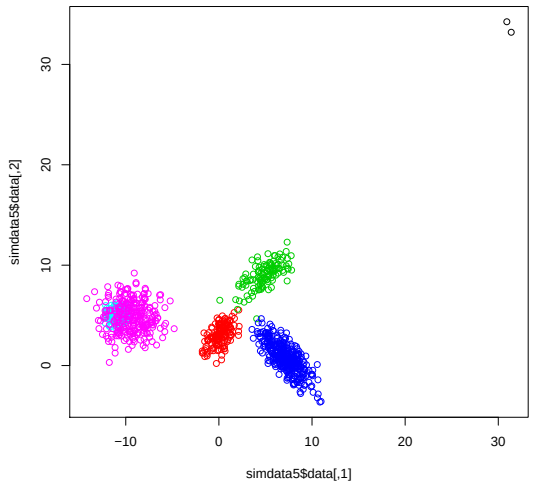
For OTRIMLE, BIC, likelihood and $K_{n,G}$ depend on c,
c depends on G,
criteria with different c hard to compare.

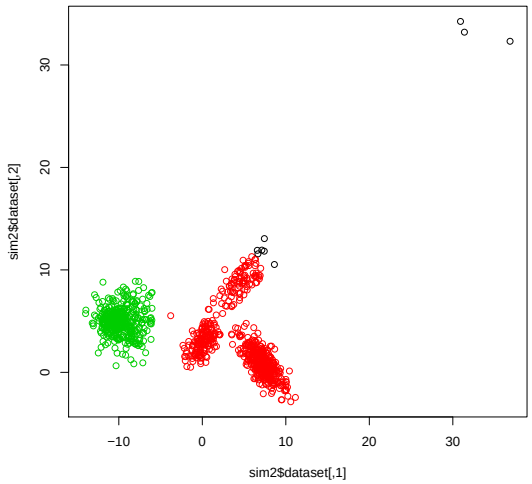
3.2 Simplicity and adequacy

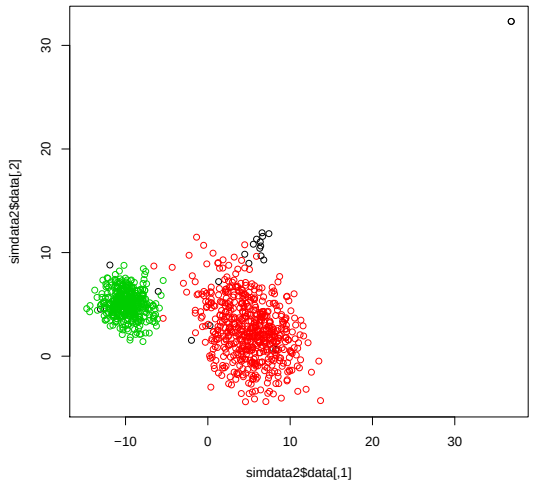
Idea: (Davies 1995, Davies and Kovac 2004)
Find simplest model (smallest G?)
that fits data *adequately*.

Adequacy: A model fit is "adequate" if its
quality on given data
cannot be told apart from fit quality
to typical data generated from the model.









Approach:

1. Generate B datasets $D_{G,b}$ from fitted models with range of G (*parametric bootstrap*).
2. Compute statistic S measuring quality of clustering.
3. G adequate if $S(\mathbf{x})$ not clearly worse than expected $S(D_{G,b})$.
4. Choose smallest/simplest/best adequate G.

Requirements:

1. OTRIMLE fits "pseudo-model", so what's the model?
2. How to choose S?

3.3 OTRIMLE's fitted model

OTRIMLE fits pseudo-density

$$\psi_c(x,\theta) = f(x) = \pi_0 c + \sum_{j=1}^{G} \pi_j \varphi_{\mathbf{a}_j,\Sigma_j}(x).$$

Can generate data from cdf

$$\Psi_c(x,\hat{\theta}) = \hat{\pi}_0 \hat{F}(x) + \sum_{j=1}^{G} \hat{\pi}_j \Phi_{\hat{\mathbf{a}}_j,\hat{\Sigma}_j}(x).$$

$\hat{F}$ weighted empirical noise distribution with weights $\frac{\hat{\pi}_0 c}{\psi_c(x_i,\hat{\theta})}$.

No assumption on distribution of noise/outliers;
this is reproduced from data.

3.4 Measuring clustering quality

Don't want to rely on Gaussian assumption,
but want clusters for which Gaussian fit makes sense.

3.4 Measuring clustering quality

Don't want to rely on Gaussian assumption,
but want clusters for which Gaussian fit makes sense.

What characterises a non-Gaussian cluster
that is well modelled by a Gaussian component?

Look for *approximately elliptical* clusters
with density decreasing from mean in all directions.

Measure deviation from such a shape.

1-d measure of symmetric density decrease from mean

. . . to be applied to single cluster.

(a) Compute kernel density estimator at q points symmetric around mean $\hat{f}(y_1), \ldots, \hat{f}(y_q)$.

(b) Sort these: $\hat{f}^{(1)} \geq \ldots \geq \hat{f}^{(q)}$.

(c) Compute $\hat{f}^{*1} \geq \ldots \geq \hat{f}^{*(q/2)}$ by averaging pairs of $\hat{f}(y_1), \ldots, \hat{f}(y_q)$.

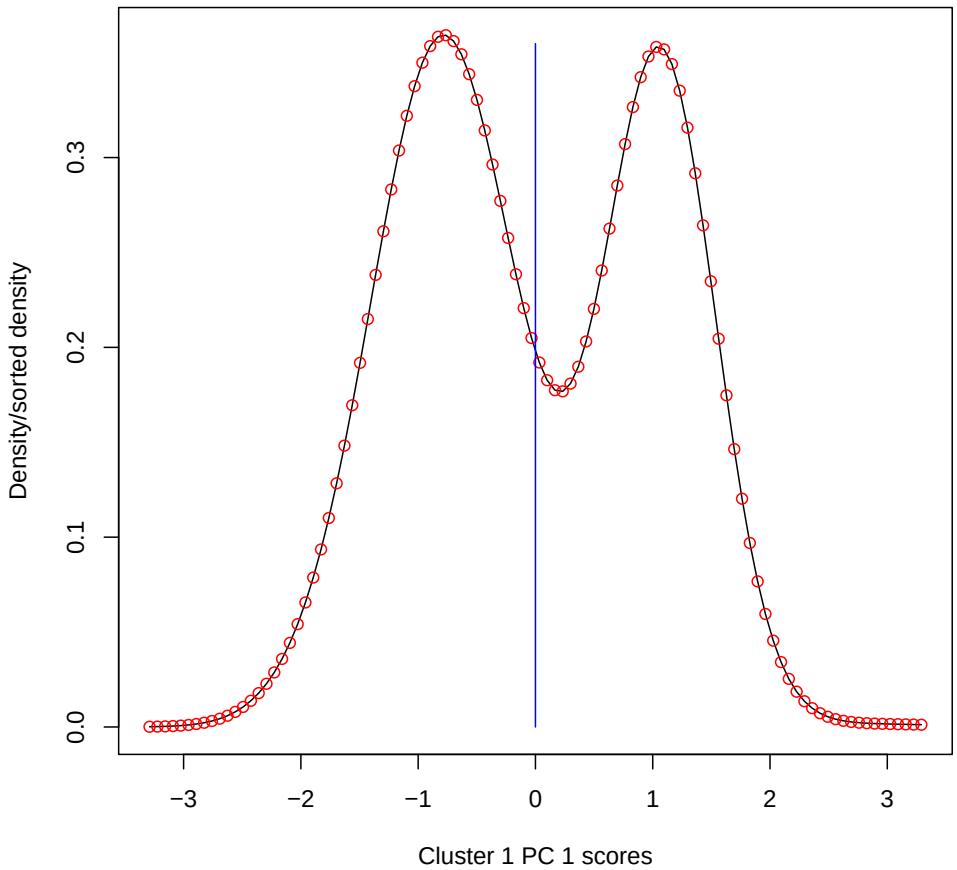
(d) Compare with kernel density from mean:

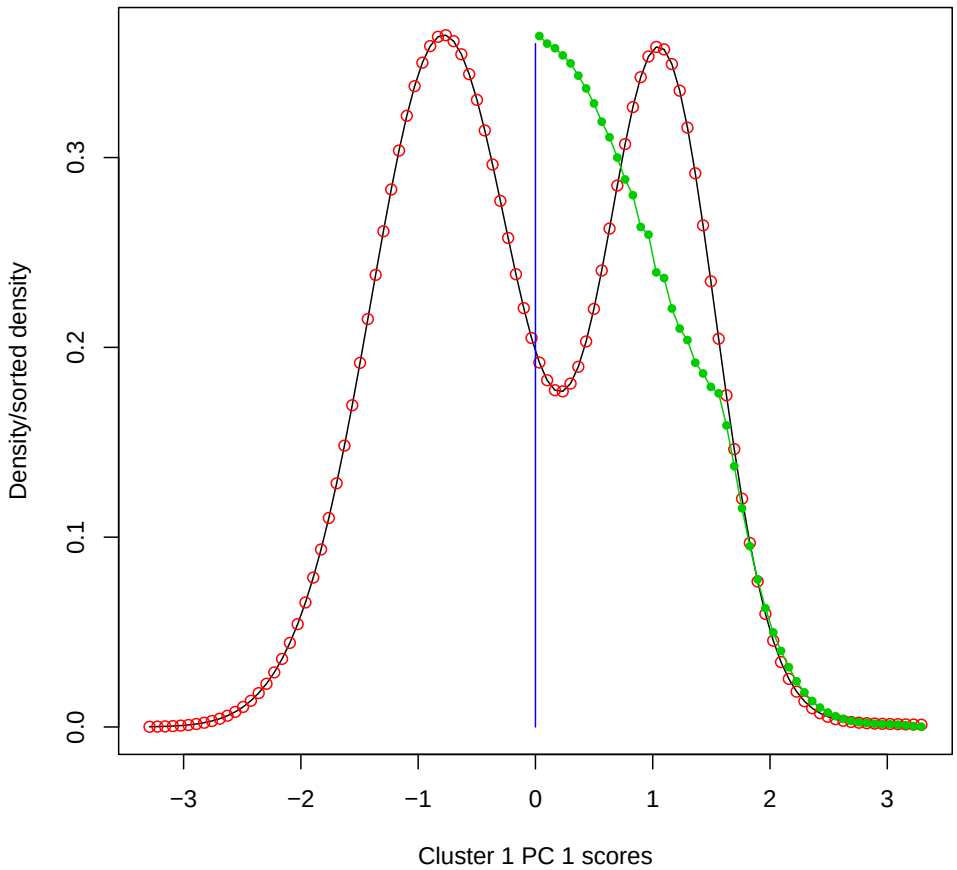
$$s_l = \sum_{i=1}^{q/2} (\hat{f}(y_{q/2+1-i}) - \hat{f}^{*i})^2,$$

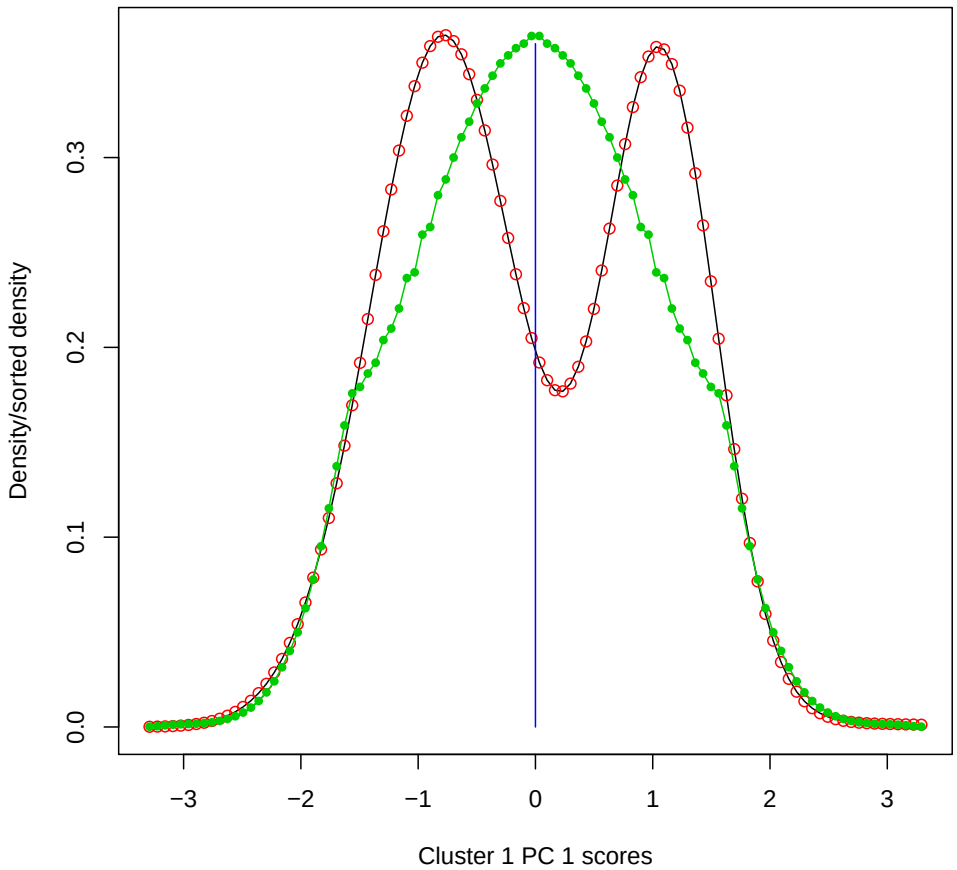
$$s_r = \sum_{i=1}^{q/2} (\hat{f}(y_{q/2+i}) - \hat{f}^{*i})^2,$$

$$T(y_1, \ldots, y_q) = \sqrt{\frac{1}{q}(s_l + s_r)}.$$

Perfect symmetric decrease from mean: $T = 0$.


Density/sorted density
0.0
0.1
0.2
0.3
−3
−2
−1
0
1
2
3
Cluster 1 PC 1 scores


Density/sorted density
0.0
0.1
0.2
0.3
−3
−2
−1
0
1
2
3
Cluster 1 PC 1 scores


Density/sorted density
0.0
0.1
0.2
0.3
−3
−2
−1
0
1
2
3
Cluster 1 PC 1 scores

Application to p-dimensional clusters

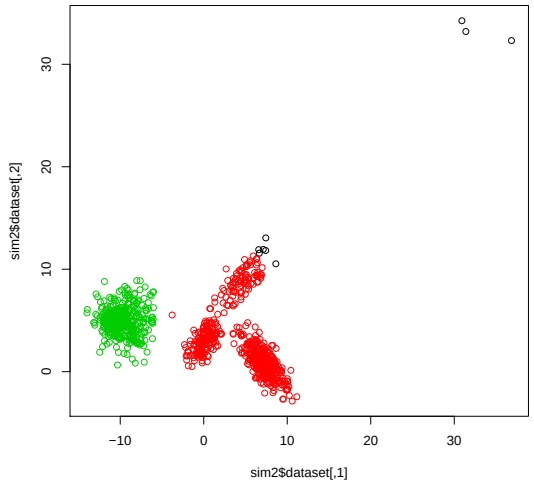
(a) Compute weighted PCA of cluster $\mathcal{C}_j$ characterised by $\big((x_1,\ldots,x_n).(\hat{\tau}_j(x_1),\ldots,\hat{\tau}_j(x_n))\big)$.

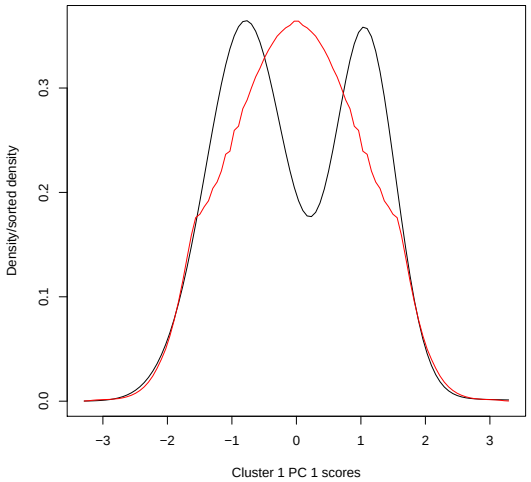
(b) $T_i,\ i=1,\ldots,p$: D-measure for weighted ith PC scores.

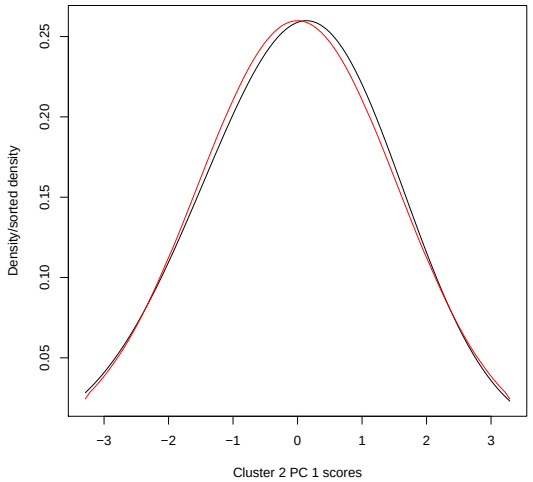
(c) $T_i^* = \frac{T_i - ET_i}{\sqrt{\mathrm{Var}(T_i)}}$, ET_i and $\mathrm{Var}(T_i)$ assuming Gaussianity.

(d) $T_j^* = \frac{1}{p}\sum_{i=1}^{p}(T_i^*)^2 1(T_i^* > 0)$.

$(T_i^*)^2 1(T_i^* > 0)$ is dominated by worst dimension,
$T_i^* < 0$ for some i will not mask bad value elsewhere.
Should be sensitive to issue in any dimension.


sim2$dataset[,2]
sim2$dataset[,1]
−10
0
10
20
30
0
10
20
30


Density/sorted density
Cluster 1 PC 1 scores
0.0
0.1
0.2
0.3
−3
−2
−1
0
1
2
3


Density/sorted density
Cluster 2 PC 1 scores
0.05
0.10
0.15
0.20
0.25
−3
−2
−1
0
1
2
3

Aggregate over clusters:

$$S(\mathbf{x}, \mathcal{C}) = \sqrt{\sum_{j=1}^{G} (T_j^*)^2}.$$

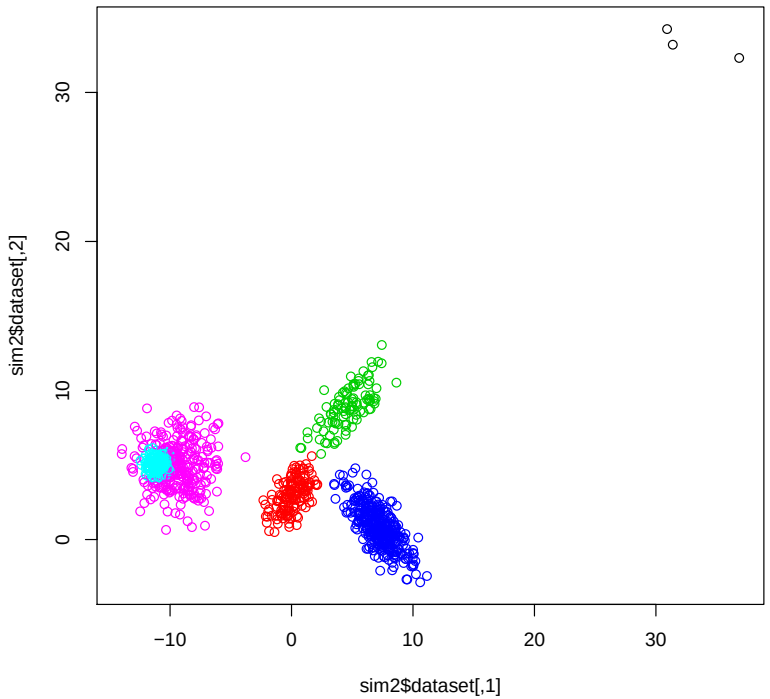
Square again to make it sensitive
to biggest issue in any cluster.

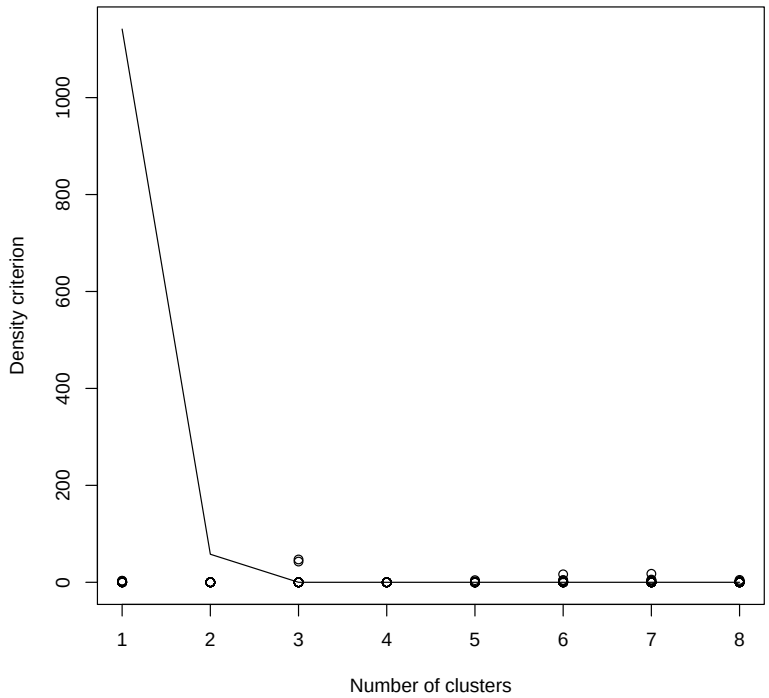
Compare with bootstrap under fitted model:

$$S^*(\mathbf{x}, \mathcal{C}) = \frac{S(\mathbf{x}, \mathcal{C}) - \bar{S}(D_{G,b})}{sd(S(D_{G,b}))}.$$

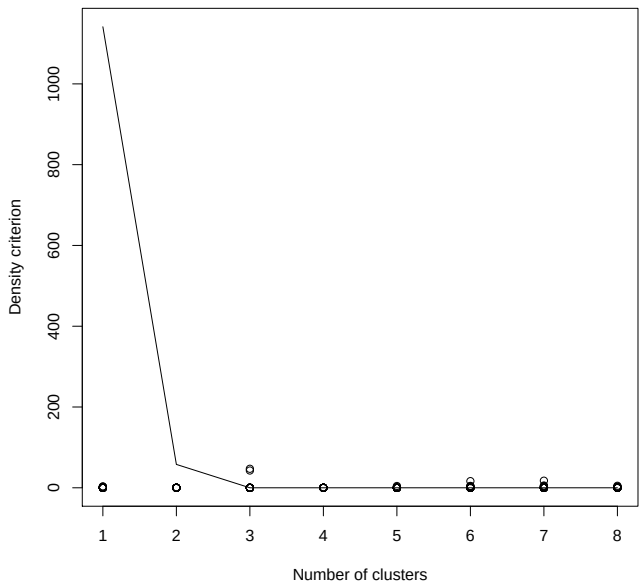
G is adequate if $S^*(\mathbf{x}, \mathcal{C}) < 2$, say.

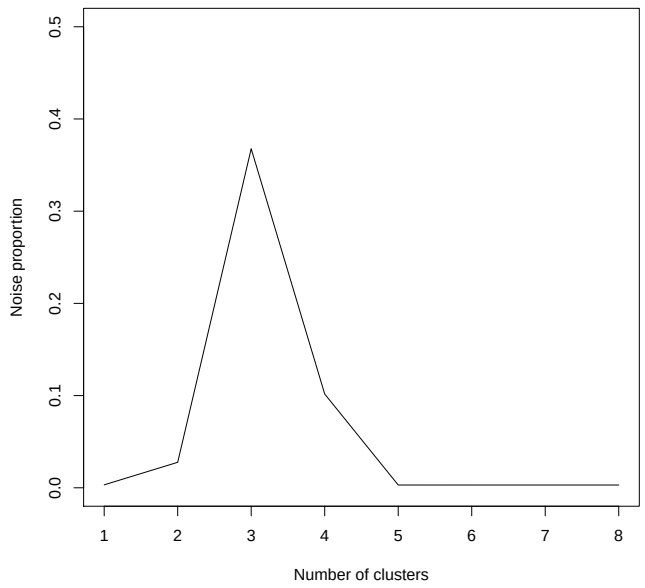
All from $G = 3$ upwards are OK.

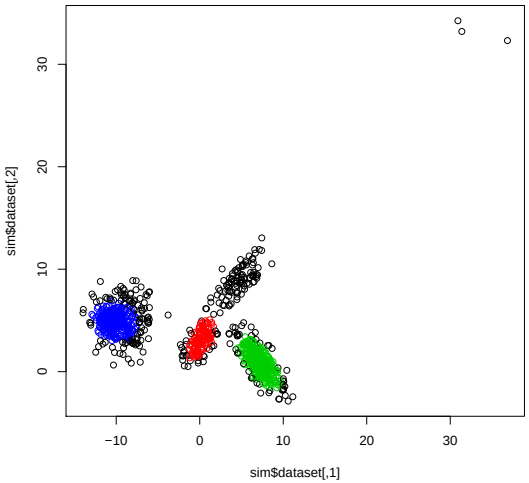


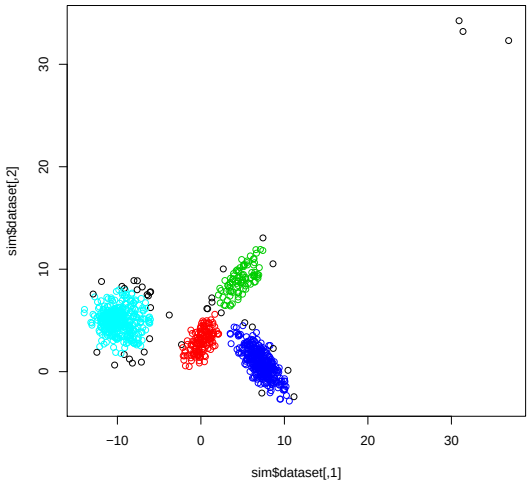


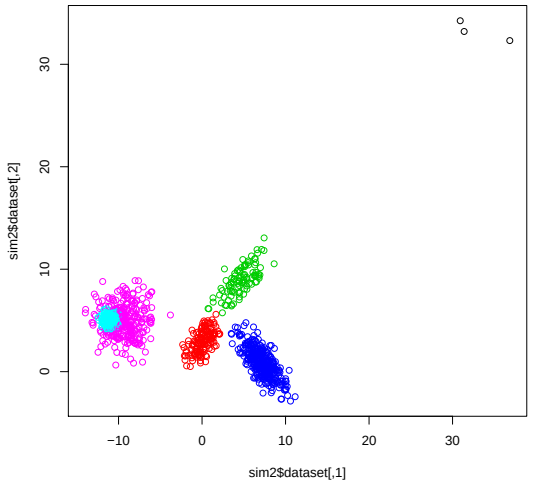
$G = 3$ and $G = 4$ have substantial noise proportion;
achieve good clustering
by "outnoising" good observations.






sim$dataset[,2]
sim$dataset[,1]
−10
0
10
20
30
0
10
20
30


sim$dataset[,2]
sim$dataset[,1]
−10
0
10
20
30
0
10
20
30


sim2$dataset[,2]
sim2$dataset[,1]
−10
0
10
20
30
0
10
20
30

Issue raised earlier:
How big and clear a group of outliers
is to be considered a cluster?

More generally: most desirable model
is not only parsimonious (low G)
but also expains many points (low π_0).

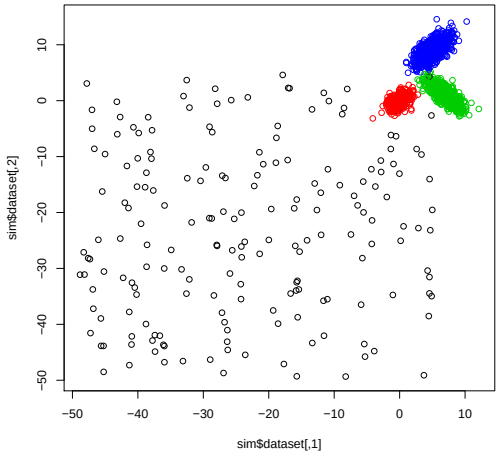
Choose critical noise proportion p_0
to trade against one cluster more.

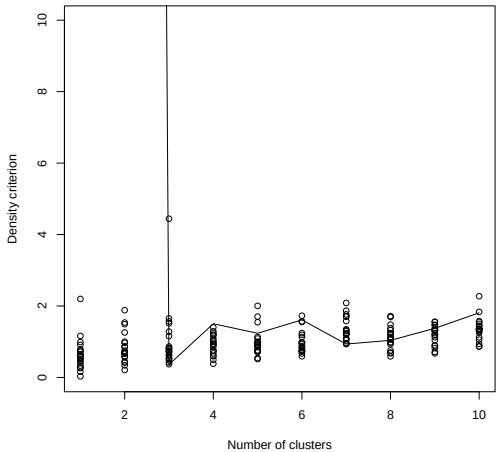
Choose G so that model is adequate and
$G + \frac{\pi_0}{p_0}$ is minimum.

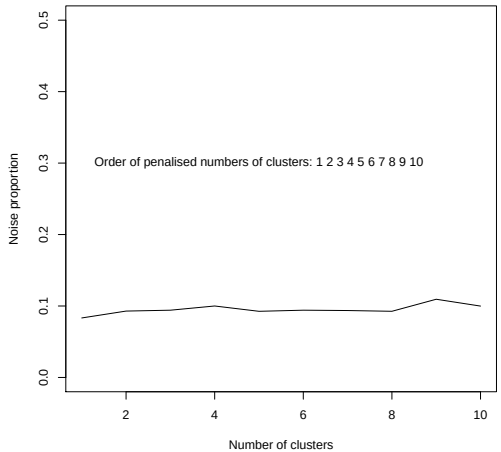
For example dataset and $p_0 = 0.05$ this yields $\hat{G} = 5$:

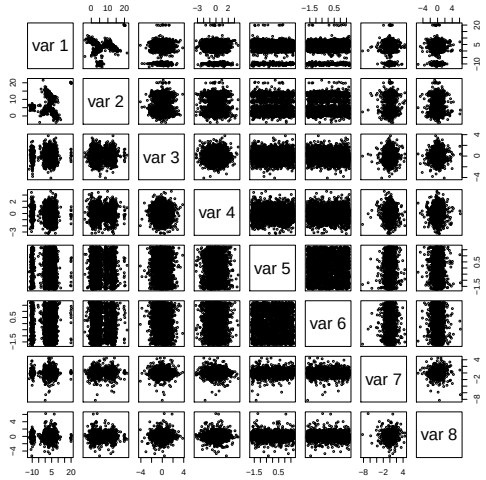
G	1	2	3	4	5	6
$G + \frac{\pi_0}{p_0}$	1.1	2.6	10.4	6.0	5.1	6.1
S^*	905.0	652.9	-0.6	-0.2	-0.4	-0.4

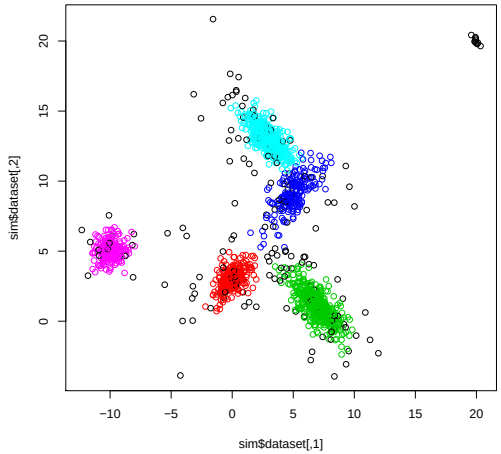
3.5 Examples

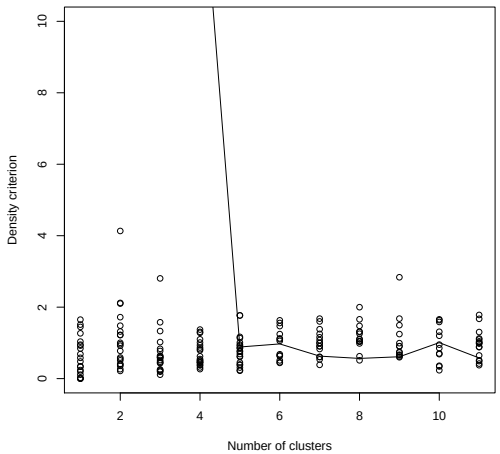

sim$dataset[,2]
sim$dataset[,1]
10
0
−10
−20
−30
−40
−50
−50
−40
−30
−20
−10
0
10

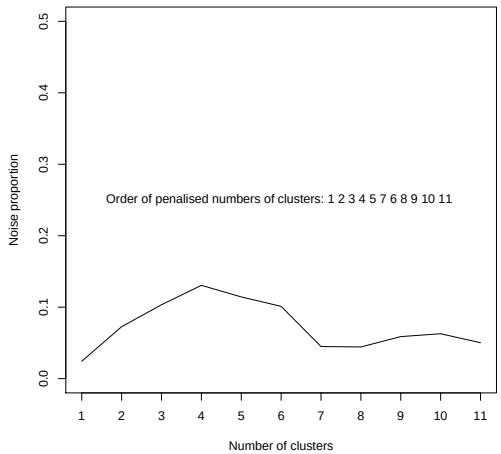

Density criterion
Number of clusters
0
2
4
6
8
10

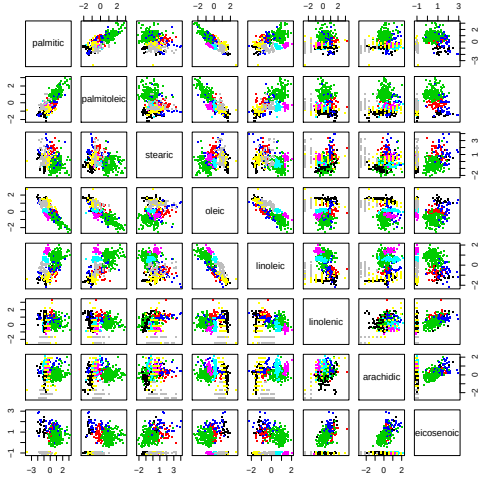

Order of penalised numbers of clusters: 1 2 3 4 5 6 7 8 9 10
Noise proportion
Number of clusters
0.0
0.1
0.2
0.3
0.4
0.5
2
4
6
8
10

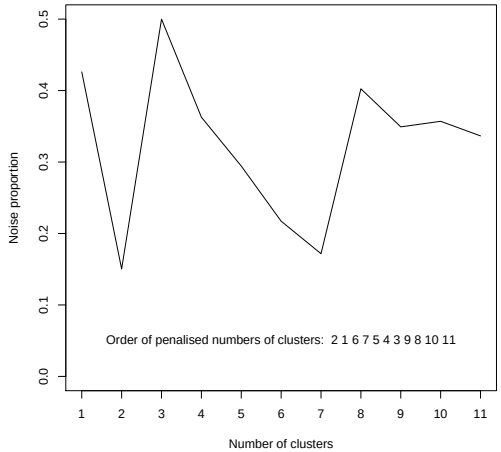

var 1
var 2
var 3
var 4
var 5
var 6
var 7
var 8

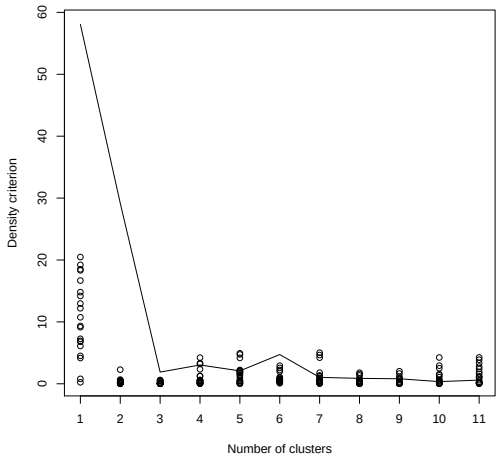

sim$dataset[,2]
sim$dataset[,1]

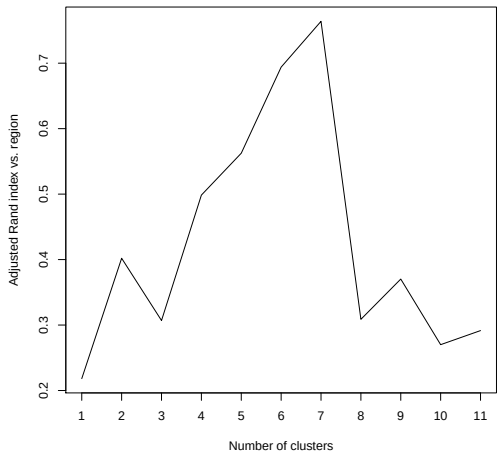

Density criterion
Number of clusters


Order of penalised numbers of clusters: 1 2 3 4 5 7 6 8 9 10 11
Noise proportion
Number of clusters


palmitic
palmitoleic
stearic
oleic
linoleic
linolenic
arachidic
eicosenoic


Noise proportion
Order of penalised numbers of clusters: 2 1 6 7 5 4 3 9 8 10 11
Number of clusters

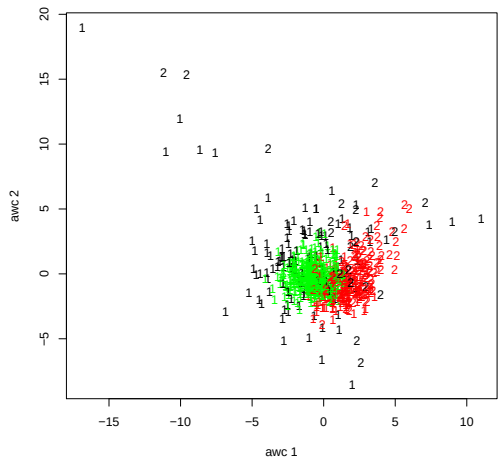

Density criterion
Number of clusters

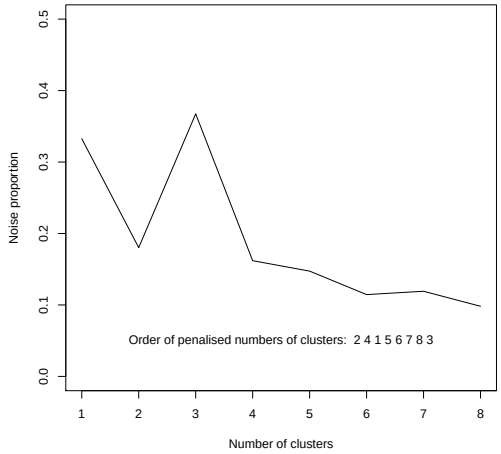

Adjusted Rand index vs. region
Number of clusters

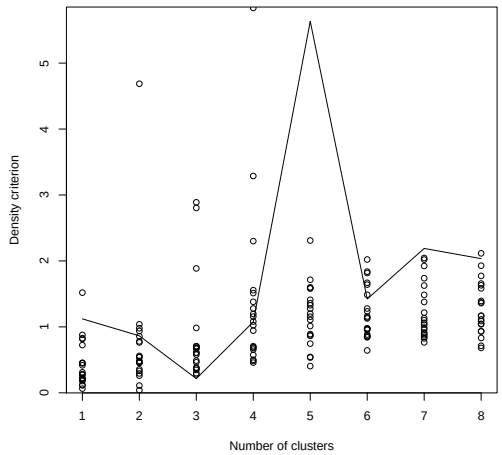
Comparing with "true" region:

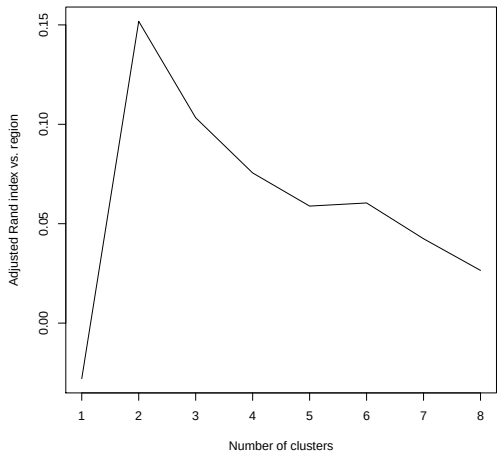
method	$\hat{G}$	ARI	remark
OTRIMLE/*S*	7	0.76	sensitive to γ
mclust	9	0.65	
mclust/noise	9	0.61	
tclust/ctlcurves	7	0.68	
t-mixture (teigen)	5	0.77	
skew-t-mixture (smsnmix)	9	0.77	

Caution: there can be reasonable clusters other than the regions.


awc 2
awc 1


Noise proportion
Order of penalised numbers of clusters: 2 4 1 5 6 7 8 3
Number of clusters

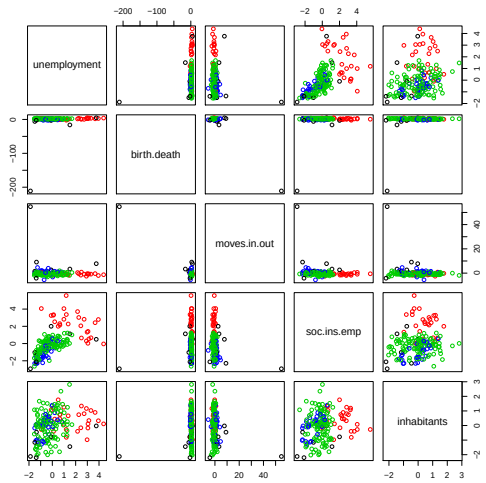

Density criterion
Number of clusters

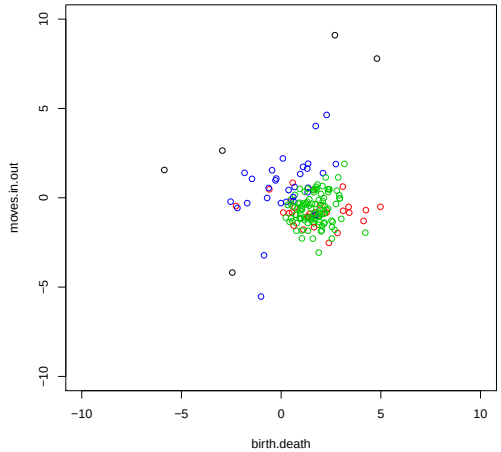

Adjusted Rand index vs. region
Number of clusters

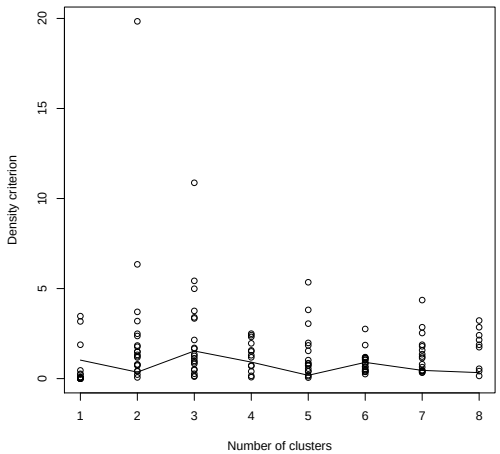
Comparing with "true" region:

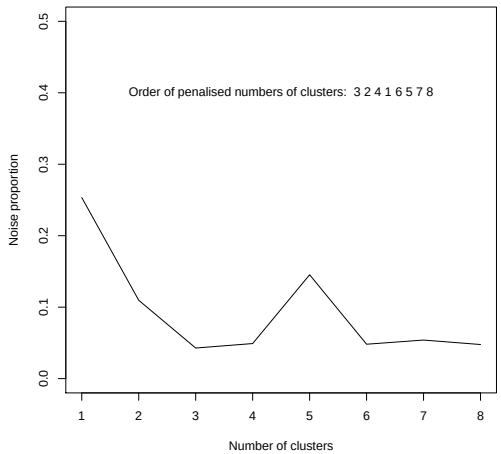
method	$\hat{G}$	ARI	remark
OTRIMLE/S	2	0.15	
mclust	2	-0.05	
mclust/noise	3	-0.01	
tclust/ctlcurves	2	0.16	$\hat{G}$ ambiguous; sens. to γ
t-mixture (teigen)	5	0.03	
skew-t-mixture (smsnmix)	2	-0.06	

Caution: there can be reasonable clusters other than the regions.


unemployment
birth.death
moves.in.out
soc.ins.emp
inhabitants


moves.in.out
birth.death


Density criterion
Number of clusters


Order of penalised numbers of clusters: 3 2 4 1 6 5 7 8
Noise proportion
Number of clusters

4 Conclusion

- Find adequate fit with smallest $G + \frac{\pi_0}{p_0}$.
- Measure adequacy by comparing within-cluster density to "ideal shape" along all PCs.
- Number of clusters problem is ill-posed and needs user decisions.
- Idea can be applied to other statistics and other clustering methods.

Solutions may be sensitive to constraint γ,
algorithm initialisation etc.
(there are stability issues and artifacts
with *all* clustering methods) $\Rightarrow$

Don't trust any automatic method
and do cluster validation!

The Gaussian clusters assumption
. . . is used:

- in the pseudo-likelihood,
- in $K_{n,G}$ for tuning the noise level
  (noise penalty can be added),
- for dimension/clusterwise standardisation of S
  (need to make clusters and losses comparable),
- for parametric bootstrap overall standardisation of S
  (only being significantly *worse* than Gaussian is rejected)

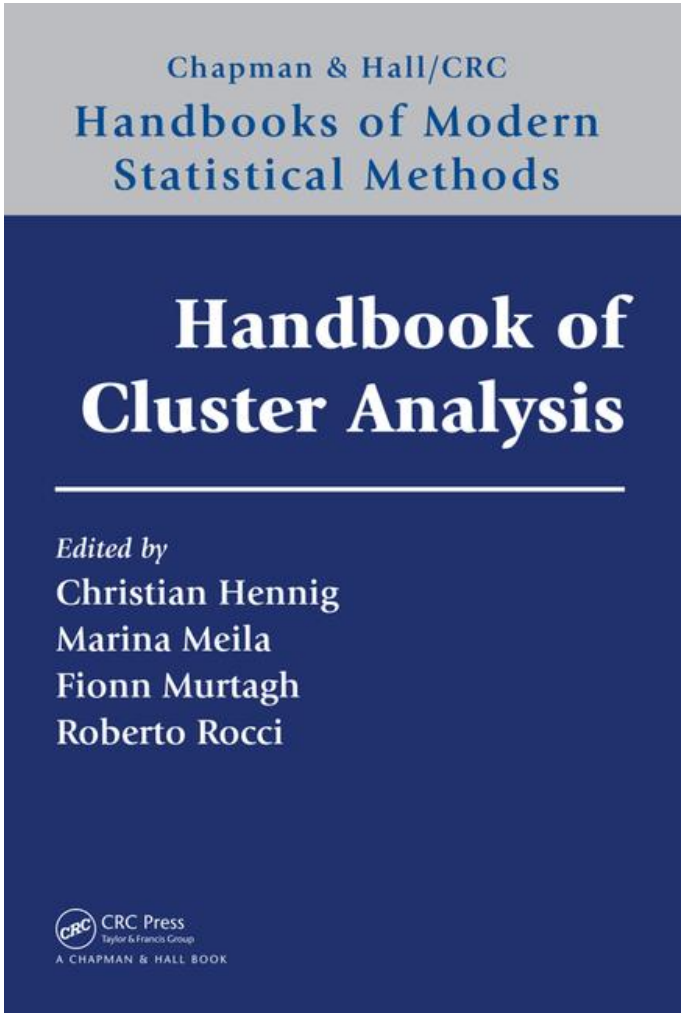
- May want to tolerate clusters slightly worse than Gaussian (or slightly skew).
  May simulate $D_{G,b}$ from model with weaker density decay; need not enforce symmetry of comparison density.
- PCA may miss critical direction;
  may use Tyler et al. ICS instead of PCA.
  (Experience: this hasn't been a problem yet.)
- Could use S instead of $K_{n,G}$ for tuning.

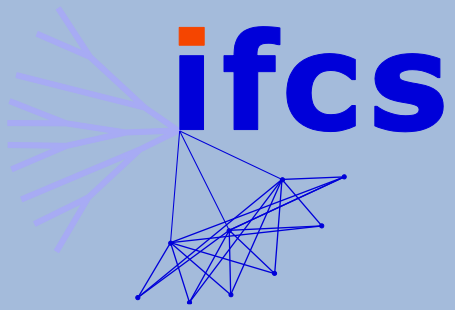
P. Coretto and C. Hennig: *Robust improper maximum likelihood: tuning, computation, and a comparison with other methods for robust Gaussian clustering*
arXiv:1406.0808
To appear in JASA (2016).

P. Coretto and C. Hennig: *A consistent and breakdown robust model-based clustering method*
arXiv:1309.6895
Submitted.

This work is supported by EPSRC Grant EP/K033972/1.

. . . and . . .


Chapman & Hall/CRC
Handbooks of Modern Statistical Methods
Handbook of Cluster Analysis
Edited by
Christian Hennig
Marina Meila
Fionn Murtagh
Roberto Rocci
CRC Press
Taylor & Francis Group
A CHAPMAN & HALL BOOK



CLUSTER BENCHMARK
DATA REPOSITORY

HTTP://IFCS.BOKU.AC.AT/REPOSITORY

ifcs

JCS
Japanese Classification Society

TOKAI UNIVERSITY

CONFERENCE VENUE
Takanawa Campus of Tokai University

IMPORTANT DATES

April 15, 2017:	Deadline abstract submission
April 15, 2017:	Deadline early bird registration
May 15, 2017:	Notification of acceptance for abstract submission
May 31, 2017:	Deadline standard registration
June 30, 2017:	Deadline late registration
August 7, 2017:	Pre-conference workshops
August 8-10, 2017:	IFCS-2017 conference