A flexible EM-like clustering algorithm for noisy data

CosmoStat Day - ML in Astrophysics

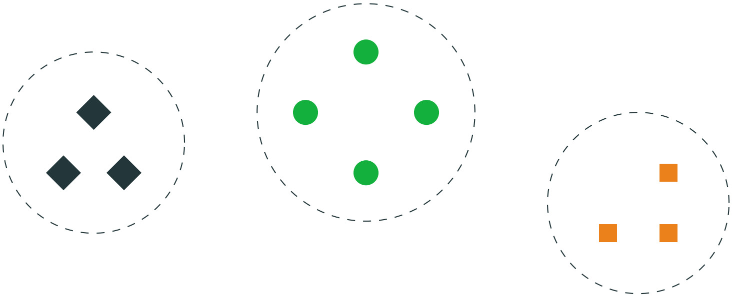
Violeta Roizman @ CentraleSupélec and UBA

Joint work with Matthieu Jonckheere and Frédéric Pascal

Introduction

Clustering

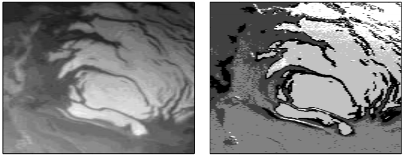
Group data points into clusters to understand the structure of the data.

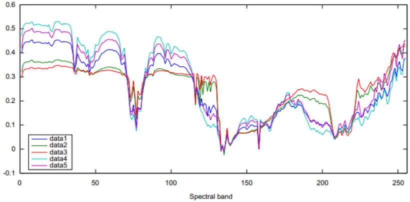


Given a notion of similarity between points, we want:

- similar points to be in the same cluster,
- really different points to be in different clusters, and
- well separated clusters.

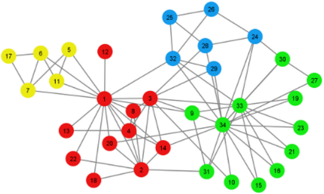
Hyperspectral image segmentation





Bouveyron et al.(2007)

Network community detection



One solution: K-means

Given $\{\mathbf{x}_i\}_{i=1}^n \subset \mathbb{R}^m$, find $\hat{\mathbf{C}} = \{C_1, ..., C_K\}$ with $\boldsymbol{\mu}_k = \frac{1}{|C_k|} \sum_{\mathbf{x} \in C_k} \mathbf{x}$ so that

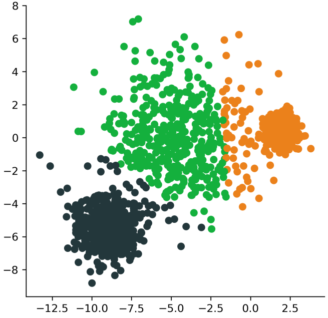
$$\hat{\mathbf{C}} = \underset{C}{argmin} \sum_{k=1}^{K} \sum_{\mathbf{x} \in C_k} \|\mathbf{x} - \boldsymbol{\mu}_k\|_2^2$$

Simple idea. ✓

Very fast. ✓

Works well only when: ✗

- round-shaped clusters,
- with similar variance, and
- well-separated.

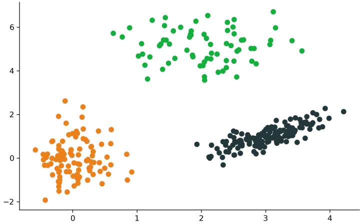


GMM: Improving K-means

We model data as a mixture of Gaussian distributions $\mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$:

$$f(\mathbf{x}) = \sum_{k=1}^{K} \pi_k f_k(\mathbf{x}),$$

with π_k the proportion of cluster k and f_k the normal pdf.



$$f_k(\mathbf{x}) = \frac{1}{(2\pi)^{m/2}|\boldsymbol{\Sigma}_k|^{1/2}} \exp\left[-\frac{(\mathbf{x}-\boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_k^{-1}(\mathbf{x}-\boldsymbol{\mu}_k)}{2}\right]$$

Expectation-Maximization (EM) algorithm

For each $\mathbf{x}_i$, Z_i indicates the cluster it belongs to.

$$E_{Z|\mathbf{X},\boldsymbol{\theta}}[l(\mathbf{x}_i, z_i; \boldsymbol{\theta})] = \sum_{i=1}^{n} \sum_{k=1}^{K} P(Z_i = k|\mathbf{X}_i = \mathbf{x}_i) \log(\pi_k f_k(\mathbf{x}_i))$$

Iterative algorithm to estimate parameters $\boldsymbol{\theta} = (\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)_{1 \leq k \leq K}$.

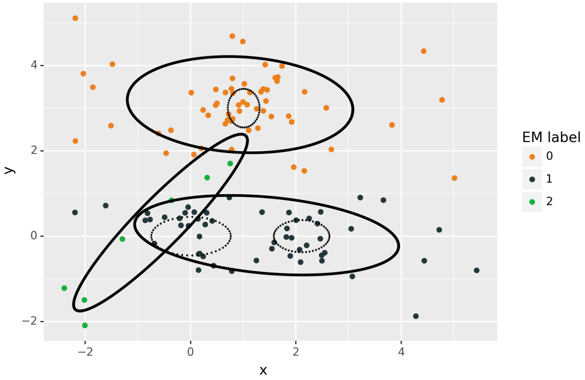
Algorithm 1: General scheme of EM Algorithm for clustering

Set initial random values $\boldsymbol{\theta}_0$;
while *not convergence* **do**
    E: Compute $p_{ik} = P(Z_i = k|\mathbf{X}_i = \mathbf{x}_i)$ based on $\boldsymbol{\theta}_{old}$;
    M: Search $\boldsymbol{\theta}_{new} = (\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)_{1 \leq k \leq K}$ that maximizes the expectation of the likelihood;
end
Assign $\mathbf{x}_i$ to $k^* = \underset{j}{argmax}\, P(Z_i = j|\mathbf{X}_i = \mathbf{x}_i)$;

Motivation

The EM algorithm has problems to cluster data with noise, different distribution shapes and outliers.

Result with data contaminated:



Why? Because estimators are not robust.

Some robust clustering literature

Types of robust clustering algorithms

Mainly two directions to **robustify clustering methods** in the literature:

- model the noise
  - Extra uniform cluster (Banfield and Raftery, 1993)
  - Model low density areas (Coretto and Hennig, 2017)
  - Mixture of Student's *t* (Peel and McLachlan, 2000)
- include classic robust techniques in the estimation
  - Trimming methods (Garcia-Escudero et al, 2008)
  - Plugged-in robust estimators (Gonzalez, 2019)

t-EM

They assume a mixture of Student's t-distributions.

A variable $X \sim t_m(\mu, \Sigma, \nu)$, its pdf is

$$f(x) = \frac{\Gamma(\frac{\nu+m}{2})|\Sigma|^{1/2}}{(\pi\nu)^{m/2}\Gamma(\nu/2)(1 + \Delta(x; \mu, \Sigma)/\nu)^{(\nu+m)/2}}$$

with $\Delta(x; \mu, \Sigma) = (x - \mu)^T \Sigma^{-1}(x - \mu)$.

We can derive an EM algorithm with μ_k and Σ_k robust estimators.

But no closed equations to update the degrees of freedom ν_k. We have to use a non-linear optimizer to estimate it.

F-EM algorithm

F-EM algorithm

Initial idea: Extend GMM to cover more general distributions.

A random vector $\mathbf{X_i}$ in the class of Compound Gaussian distributions can be written like this:

$$\mathbf{X}_i = \boldsymbol{\mu} + \sqrt{\tilde{\tau}_i}\,\mathbf{A_j}\,\mathbf{g}_i,$$

where $\tilde{\tau}_i$ is a positive random variable independent from $\mathbf{g}_i$, $\mathbf{g}_i \sim \mathcal{N}(0, I_m)$ and $\mathbf{A_j}\mathbf{A_j}^T = \boldsymbol{\Sigma}_j$.

We do not fix a distribution for $\tilde{\tau}_i \rightarrow$ consider an approximated model:

deterministic τ_i [PCO$^+$08]

F-EM algorithm

Given $\{\mathbf{x}_i\}_{i=1}^n \in \mathbb{R}^m$ we have to estimate the usual parameters

$$\boldsymbol{\Theta} = \{(\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)\}_{k=1,..,K}$$

but we now have a lot of τ parameters

$$\widetilde{\boldsymbol{\Theta}} = \{\tau_{ik}\}_{\substack{k=1,..,K \\ i=1,..,n}}$$

that give F-EM the flexibility to accommodate to heavier (or lighter) tails or outliers.

We derive the two-step algorithm based on the likelihood with fixed τ and obtain the following:

$$\widehat{\tau}_{ik} = \frac{(\mathbf{x}_i - \widehat{\boldsymbol{\mu}}_k)^T \widehat{\boldsymbol{\Sigma}}_k^{-1} (\mathbf{x}_i - \widehat{\boldsymbol{\mu}}_k)}{m}$$

F-EM algorithm

On the other side we have linked fixed-point equations for the parameters we most care about:

$$\widehat{\boldsymbol{\mu}}_k = \frac{\displaystyle\sum_{i=1}^{n} \frac{p_{ik}\mathbf{x}_i}{(\mathbf{x}_i - \widehat{\boldsymbol{\mu}}_k)^T \widehat{\boldsymbol{\Sigma}}_k^{-1} (\mathbf{x}_i - \widehat{\boldsymbol{\mu}}_k)}}{\displaystyle\sum_{i=1}^{n} \frac{p_{ik}}{(\mathbf{x}_i - \widehat{\boldsymbol{\mu}}_k)^T \widehat{\boldsymbol{\Sigma}}_k^{-1} (\mathbf{x}_i - \widehat{\boldsymbol{\mu}}_k)}}$$

$$\widehat{\boldsymbol{\Sigma}}_k = m \sum_{i=1}^{n} \frac{w_{ik}(\mathbf{x}_i - \widehat{\boldsymbol{\mu}}_k)(\mathbf{x}_i - \widehat{\boldsymbol{\mu}}_k)^T}{(\mathbf{x}_i - \widehat{\boldsymbol{\mu}}_k)^T \widehat{\boldsymbol{\Sigma}}_k^{-1} (\mathbf{x}_i - \widehat{\boldsymbol{\mu}}_k)},$$

with $w_{ik} = p_{ik} / \sum_{l=1}^{n} p_{lk}$. We impose $\mathrm{tr}(\boldsymbol{\Sigma}) = m$

They are like Tyler's M-estimators with extra weights coming from the mixture.

Tyler's estimators intuitively

Like usual sample estimators with small weights for outlying points

$$\frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i \Longrightarrow \frac{1}{n}\sum_{i=1}^{n} w_i\mathbf{x}_i$$

$$\frac{1}{n}\sum_{i=1}^{n}(\mathbf{x}_i-\widehat{\boldsymbol{\mu}})^T(\mathbf{x}_i-\widehat{\boldsymbol{\mu}}) \Longrightarrow \frac{1}{n}\sum_{i=1}^{n} w_i(\mathbf{x}_i-\widehat{\boldsymbol{\mu}})^T(\mathbf{x}_i-\widehat{\boldsymbol{\mu}})$$

with $w_i \approx \frac{1}{(\mathbf{x}_i-\widehat{\boldsymbol{\mu}})^T\widehat{\boldsymbol{\Sigma}}^{-1}(\mathbf{x}_i-\widehat{\boldsymbol{\mu}})}$

High-dimension can actually help

When the dimension grows we can estimate the τ_i's better.

Under some assumptions, if n and m are big enough then

$$\sqrt{m}(\widehat{\tau}_i - \tau_i) \overset{approx}{\sim} \mathcal{N}\left(0, 2\tau_i^2\right)$$

This is in accordance with previous RMT results ($m/n = \gamma \to (0, 1)$).

We can combine this result with parsimonious restrictions on the covariance matrix to avoid identifiability issues in the case of very big m.

Measuring the performance

We compare our algorithm to

- k-means
- EM (GMM)
- Mixture of Student's *t* (*t*-EM or EMMIX)
- HDBSCAN
- Spectral Clustering

Based on the ground truth, we use metrics to compare:

- Adjusted Mutual Information (AMI)
- Adjusted Rand Index (AR)

For simulations also:

- Estimation error of the parameters

Some simulation results

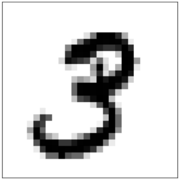
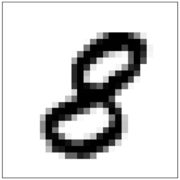
Simulations: Mixtures of t-distributions with different degrees of freedom and covariance matrix classes

Setup	distributions	$\boldsymbol{\mu}_1$	$\boldsymbol{\mu}_2$	$\boldsymbol{\mu}_3$	$\boldsymbol{\Sigma}_1$	$\boldsymbol{\Sigma}_2$	$\boldsymbol{\Sigma}_3$
1	$3\ t, dof = 3$	$\mathcal{U}_{(0,1)}$	$2 * \mathbf{1}_m$	$1.5 * \mathbf{1}_m + 3e_1$	diag	diag	I_m
2	$3\ t, dof = 10$	$\mathcal{U}_{(0,1)}$	$5 * \mathbf{1}_m$	$1.5 * \mathbf{1}_m + \varepsilon$	diag	diag	I_m

Dataset	error	EM	**EM** (sd)	t-EM	**t-EM** (sd)	F-EM	**F-EM** (sd)
Setup 1	μ_1	0.2179	0.3373	**0.0220**	0.0079	0.0237	0.0075
Setup 1	μ_2	0.2725	0.6624	**0.0209**	0.0068	0.0235	0.0080
Setup 1	μ_3	0.3281	0.8190	**0.0232**	0.0067	0.0235	0.0077
Setup 1	Σ_1	0.2534	0.4563	0.0097	0.0028	**0.0089**	0.0020
Setup 1	Σ_2	0.2566	0.5023	0.0089	0.0021	**0.0087**	0.0018
Setup 1	Σ_3	0.2633	0.5442	0.0097	0.0020	**0.0089**	0.0019
Setup 2	μ_1	0.0398	0.0559	0.0306	0.0390	**0.0224**	0.0072
Setup 2	μ_2	0.0408	0.0541	**0.0190**	0.0063	0.0218	0.0072
Setup 2	μ_3	0.0338	0.0305	0.0340	0.0503	**0.0234**	0.0077
Setup 2	Σ_1	0.0196	0.0111	0.0104	0.0086	**0.0081**	0.0017
Setup 2	Σ_2	0.0203	0.0125	**0.0077**	0.0018	0.0078	0.0016
Setup 2	Σ_3	0.0187	0.0110	0.0097	0.0062	**0.0083**	0.0017

Table 1: Average and standard deviation of the errors.

Real data clustering results

MNIST[LeCun'98]

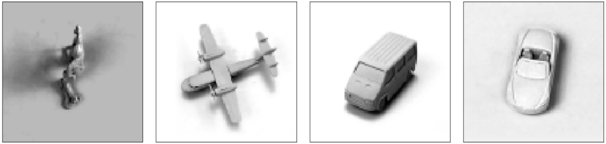
NORB[LeCun'04]

Dataset	m	n	kmeans	EM	t-EM	F-EM	spectral
MNIST 38	30	1600	0.2203	0.4878	0.5520	**0.5949**	0.5839
MNIST 71	30	1600	0.7839	0.8414	**0.8947**	0.8811	0.8852
MNIST 386	30	1800	0.6149	0.7159	0.7847	0.7918	**0.8272**
MNIST 386+noise	30	2080	0.3622	0.4418	0.4596	**0.4664**	0.3511
small NORB	30	1400	0.0012	0.0476	0.4894	**0.4997**	~ 0
20newsgroup	100	1400	0.2637	0.3526	0.4496	**0.5087**	0.1665

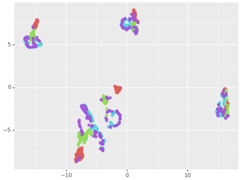
Table 2: AMI index median measuring the performance of the different algorithms.

Real data clustering results - The NORB case

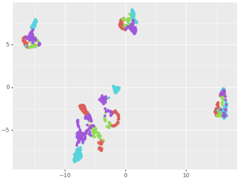
Dataset	kmeans	EM	t-EM	F-EM	spectral
small NORB	0.0012	0.0476	0.4894	**0.4997**	~ 0



t-SNE embedding of the dataset colored with labels:



real labels



F-EM labels

EM labels

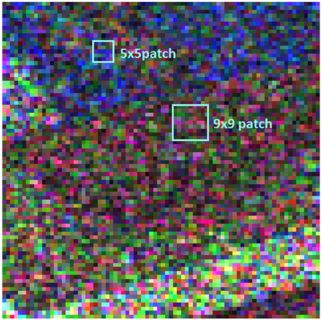
Extension of F-EM for PolSAR Images Segmentation

Extension for PolSAR images segmentation

Segment PolSAR images with a clustering algorithm to detect land use.

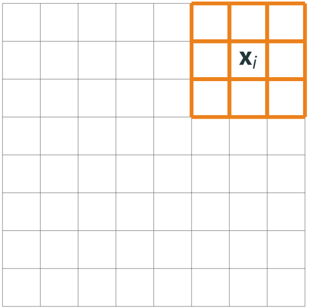
Keep **flexibility** but also take advantage of **spatial structure**.

Compute each τ by **patches** $\rightarrow$ R-EM.



R-EM: Modifying F-EM

We propose this modification to include spacial information of the neighbors in the scale τ computation:



For each pixel $\mathbf{x}_i$:

For each pixel $\mathbf{x}_t$ in the patch of $\mathbf{x}_i$:

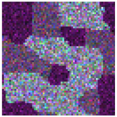
$$\tau_{tk}^{(l)} = \frac{(\mathbf{x}_t - \boldsymbol{\mu}_k^{(l)})^T (\boldsymbol{\Sigma}_k^{(l)})^{-1} (\mathbf{x}_t - \boldsymbol{\mu}_k^{(l)})}{m}$$

Set $\tau_{ik}^{(l)} = g(\{\tau_{tk}^{(l)}\}_t)$

For different patch sizes and different $g(x)$ summary functions as mean, median and trimmed mean.

Simulation example - clustering results

Image example



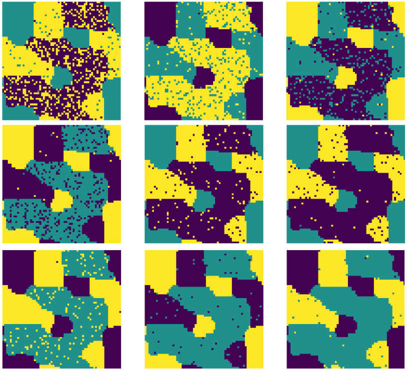
Classes

From left to right: k-means, EM and R-EM

6-looked

9-looked

12-looked



Clustering accuracy

n-looked	k-means	EM	R-EM
6	0.85	**0.92**	**0.92**
9	0.82	0.88	**0.91**
12	0.96	0.98	**0.99**

Conclusions

Conclusions and Future work

- We developed F-EM: a flexible clustering algorithm,
- and an extension for image segmentation applied to PolSAR images , IEEE-CAMSAP 2019.
- The source code of the F-EM algorithm is available here:

`github.com/violetr/fem`

- Consider more general distributions.
- Extend to the complex case.
- Design a method to reject points.

Thank you for your attention.

Any questions?

References

F. Pascal, Y. Chitour, J-P. Ovarlez, P. Forster, and P. Larzabal.
Covariance structure maximum-likelihood estimates in compound gaussian noise: Existence and algorithm analysis.
Trans. Sig. Proc., 56(1):34–48, January 2008.

V. Roizman, G. Drašković, and F. Pascal.
A new clustering algorithm for PolSAR images segmentation.
In *IEEE-CAMSAP-2019*, Guadeloupe, France, December 2019.

V. Roizman, M. Jonckheere, and F. Pascal.
A flexible EM-like clustering algorithm for noisy data.
arXiv e-prints, page arXiv:1907.01660, Jul 2019.