# CECM algorithm - README

Violaine Antoine
violaine.antoine@univ-bpclermont.fr

May 26, 2010

---

## 1 Algorithm explanations

The objectif function of CECM is the following :

$$\begin{aligned} J_{CECM}(M,V) = (1-\varphi)J_{ECM} + \varphi J_C = \quad & (1-\varphi)\sum_{i=1}^{n}\sum_{A_k\neq\emptyset}|A_k|^{\alpha}m_{ik}^{\beta}d_{ik}^{2} + \sum_{i=1}^{n}\rho^2 m_{i\emptyset}^{\beta} \\ & +\varphi(\sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathcal{M}} pl_{i\times j}(\overline{\theta}) + \sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathcal{C}} pl_{i\times j}(\theta)), \end{aligned} \tag{1}$$

We consider $n$ objects. The mass $m_{ik}$ represents the degree of belief that the object $\mathbf{x}_i$ belong the subset $A_k$, $d_{ik}$ is the distance between $\mathbf{x}_i$ and $\omega_k$, $\rho$ is the distance of all the objects to the empty set, $pl_{i\times j}(\theta)$ is the plausibility that the objects $\mathbf{x}_i$ and $\mathbf{x}_j$ are in the same class and $pl_{i\times j}(\overline{\theta})$ is the plausibility that the two objects are in a different class. Theoretical explanations can be found in [1, 2].

Note that in the script CECM, the two terms are normalized : $J_{ECM_N} = \frac{1}{|A|n}J_{ECM}$ and $J_{C_N} = \frac{1}{|\mathcal{M}|+|\mathcal{C}|}J_C$. The algorithm is the following :

1. Initialization : centroids are first calculated randomly or by using the FCM algorithm, then masses are computed using an iteration of the ECM algorithm with an euclidean distance.
2. Compute the masses with the respect of the constraints thanks to the solqp algorithm [3].
3. Compute the centroids.
4. If a mahalanobis distance is selected, compute the distances.
5. Return to 2 until the centroids are stabilized

## 2 Using CECM script

The CECM script is a function. It is composed of three files : *CECM.m*, *setCentersECM.m* and *setDistances.m*. An extra file *addNewConstraints.m* is provided and enables users to introduce constraints by selecting randomly pairs of objects. Finally *iris.m* is a script to show how to use the CECM function.

The input arguments of this function are :

- x : an input matrix of $n \times p$, where $p$ is the number of attributes
- K : the number of desired clusters
- matConst : a matrix $n \times n$ containing constraints : a Must-link constraint is represented by a 1 value and a Cannot-Link contraints by a -1 value. 0 values correspond to no constraints. The matrix is transformed in the algorithm in order to be symetric.
- Optional :
  - option.init :
    - 0 : random initialization of the center (it is the default value)
    - 1 : initialization of the center with FCM.

- option.alpha : exponent $\alpha$ allowing to control the degree of penalization for the subsets with high cardinality. cf equation 1. $\alpha = 1$ by default.
- option.rho2 : squared distance $\rho^2$ of all objects to the empty set. $\rho^2 = 100$ by default.
- option.bal : tradeoff between the objectif function $J_{ecm}$ and the constraints : $Jcecm = (1 - bal)J_{ECM} + bal J_C$, where $bal \in [0, 1]$. Default value is 0.5.
- parameters.distance :
  - 0 : Euclidean distance (default value).
  - 1 : Mahalanobis distance. The mahalanobis distance set a covariance matrix for each cluster.

option is a matlab structure and can be declared like this : option = struct('init',0,'alpha',1,'rho2',1000,'gamma',1,'eta',1);

The output arguments of the CECM function are :

- m : the masses function
- g : the centroids
- BetP : the pignistic probability
- J : the objective function

# References


[1] V. Antoine, B. Quost, M.-H Masson, T. Denoeux, *CECM: Adding pairwise constraints to evidential clustering*, fuzz-ieee, Barcelona, Spain, July 2010.

[2] V. Antoine, B. Quost, M.-H. Masson and T. Denoeux. *CECM: Constrained Evidential C-Means algorithm. Computational Statistics and Data Analysis*, Vol. 56, Issue 4, pages 894-914, 2012.

[3] Y. Ye, E. Tse, *An extension of Karmarkar's projective algorithm for convex quadratic programming*, Mathematical Programming, Springer, 1989.