A macro-DAG structure based mixture model

Bernard Chalmond

Preprint, 2013

A macro-DAG structure based mixture model

BERNARD CHALMOND

University of Cergy-Pontoise, France and CMLA, Ecole Normale Supérieure de Cachan, France *

Abstract- In the context of unsupervised classification of multidimensional data, we revisit the classical mixture model in the case where the dependencies among the random variables are described by a DAG structure. The structure is considered at two levels, the original DAG and its macro-representation. This two-level representation is the main base of the proposed mixture model. To perform unsupervised classification, we propose a dedicated algorithm called EM-mDAG, which extends the classical EM algorithm. In the Gaussian case, we show that this algorithm can be efficiently implemented. The experiments reveal that this method favors the selection of a small number of classes.

Keywords: Mixture model, DAG structure, Bayesian network, EM algorithm

.

1. Introduction

Let $\mathbf{X}$ be a random vector with values in $\mathbf{R}^n$ for which we have a N-sample $\mathcal{X} = \{\mathbf{x}_1, ..., \mathbf{x}_N\}$ with $n < N$. Our goal is the clustering of $\mathcal{X}$. This task is approached through a mixture model but with a particular constraint that makes the specificity of our contribution.

The dependency structure among the n components X^j of $\mathbf{X}$ is subject to a structure represented by a DAG, in other words $\mathbf{X}$ is a Bayesian network. This structure induces a partition of $\mathbf{X}$ into $M+1$ random vectors called *macro-variables* : $\mathbf{X} = \uplus_{m=0}^{M} X^{J_m}$, where $X^{J_m} = (X^{j_1}, ..., X^{j_m})$ when $J_m = \{j_1, ..., j_m\}$. Fig.1 depictes an example with $M = 3$ and $J_0 = \{1\}$, $J_1 = \{2, 3\}$, $J_2 = \{4, 5\}$, $J_3 = \{6, 7, 8\}$.

Each macro-variable X^{J_m} is dependent on a hidden class variable C^m with values in $\mathcal{K}_m = \{1, 2, ..., \nu_m\}$. Each occurrence in $\mathcal{K}_m$ is the number of a class called *elementary class*. Therefore $\mathbf{X}$ is dependent on the hidden multi-class variable $\mathbf{C} = (C^1, ..., C^M)$ whose values are in $\mathcal{K} = \otimes_{m=0}^{M} \mathcal{K}_m$. Each $(M+1)$-tuple of $\mathcal{K}$ refers to a set of elementary classes called *composite class*. The $(M+1)$-tuples can be interpreted as pathways connecting the elementary classes through the macro-variables as it is illustrated in Table 1. The objective is to find the most probable pathways. We considerer the mixture model

$$p_{\bar{\theta}}(\mathbf{x}) = \sum_{\mathbf{k} \in \mathcal{K}} \alpha_{\mathbf{k}} \, p_{\bar{\theta}_{\mathbf{k}}}(\mathbf{x} \mid \mathbf{k}) \, ,$$

where the probability distribution $p_{\bar{\theta}_{\mathbf{k}}}(\mathbf{x} \mid \mathbf{k})$ is that of the Bayesian network conditional on the composite class $\mathbf{k}$ and $\bar{\theta}_{\mathbf{k}}$ denotes the set of parameters defining this distribution.

* *E-mail* : bernard.chalmond@cmla.ens-cachan.fr

Table 1. Composite class numbering for $M = 3$ and $\nu_0 = 1, \nu_1 = \nu_2 = 2, \nu_3 = 4$. This table gives the exhaustive list of the 16 composite classes $\mathcal{K}$, where each column is an $(M+1)$-tuple $(1, \mathbf{k})$ with $\mathbf{k} \in \mathcal{K}$.

m=0 :	1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1
m=1 :	1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2
m=2 :	1, 1, 1, 1, 2, 2, 2, 2, 1, 1, 1, 1, 2, 2, 2, 2
m=3 :	1, 2, 3, 4, 1, 2, 3, 4, 1, 2, 3, 4, 1, 2, 3, 4

In this paper we describe this mixture model and we give a version of the EM algorithm, called EM-mDAG, for performing unsupervised classification. One of the main role of the EM-mDAG algorithm is to reveal probabilistic relationships among hidden elementary classes. Its implementation is done in the Gaussian case. Simulations illustrate the method and reveal a specific property. The EM-mDAG algorithm can select a small number of significant composite classes in $\mathcal{K}$.

2. Models and Method

2.1. Basic knowledge

• *Conventional mixture model for non supervised classification.*

Let a random vector $\mathbf{X} = (X^1, ..., X^j, ..., X^n)$ with values in $\mathbf{R}^n$. We assume that its probability distribution $p_\phi(\mathbf{x})$ is a mixture of ν distributions $\{p_{\theta_k}(\mathbf{x})\}$ as follows :

$$p_\phi(\mathbf{x}) = \sum_{k=1}^{\nu} \alpha_k \; p_{\theta_k}(\mathbf{x}) \;\; \text{with} \;\; \sum_{k=1}^{\nu} \alpha_k = 1. \tag{2.1}$$

$p_{\theta_k}(\mathbf{x})$ is defined by a parametric law of parameters θ_k, as for instance the Gaussian law. The parameter set is denoted $\phi = \{\alpha, \theta\}$ where $\alpha = \{\alpha_k\}$ and $\theta = \{\theta_k\}$. This mixture model can be interpreted in the context of unsupervised data classification. Let C be the hidden variable, which is an indicator variable of classes with values in $\{1, ..., \nu\}$. Then, (2.1) is rewritten as

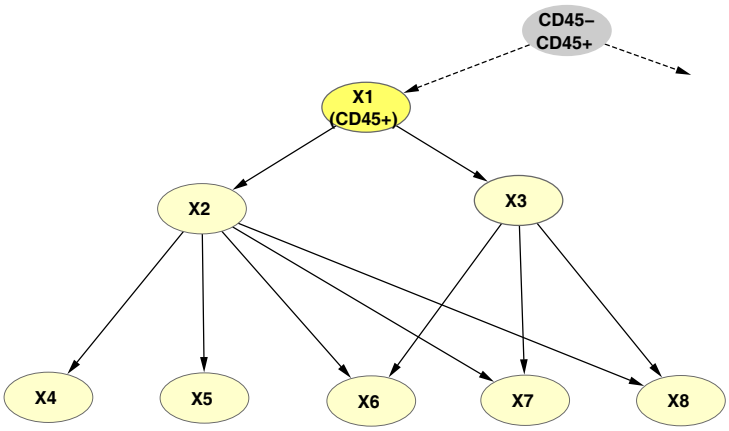
$$p_\phi(\mathbf{x}) = \sum_{k=1}^{\nu} \; P(C = k) \; p_{\theta_k}(\mathbf{x} \mid C = k). \tag{2.2}$$

The classification is to assign a class to every observation $\mathbf{x}$, [1]. When ϕ is given, the MAP decision rule consists to choose the class

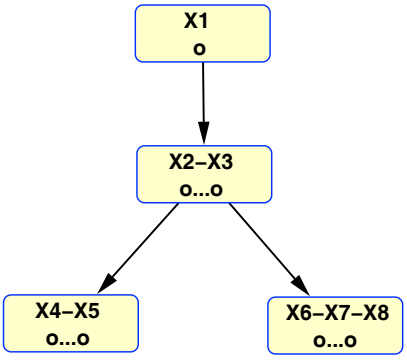
$$\hat{k}(\mathbf{x}) = \arg\max_k P_\phi(C = k \mid \mathbf{x}). \tag{2.3}$$

Otherwise, things are more complicated because $k(\mathbf{x})$ and ϕ have to be simultaneously estimated. On the basis of maximum likelihood, the EM algorithm allows this estimation from a sample $\mathcal{X} = \{\mathbf{x}_1, ..., \mathbf{x}_N\}$ of $\mathbf{X}$.

[1] A class is defined by its number and its parameters. More often, we confound "class" and "class number".



(a)



(b)

Figure 1. Two-level structure. (a) DAG structure. (b) Macro-DAG structure with its macro-variables $X^{J_1} = (X^2, X^3)$, $X^{J_2} = (X^4, X^5)$ and $X^{J_3} = (X^6, X^7, X^8)$; the small circles depict the elementary classes.

The general formulation of the EM algorithm, which is also valid for our particular case, reads as follows. If $\phi(\ell)$ is an estimation of ϕ, then an updated estimation is :

$$\begin{aligned} \phi(\ell+1) &= \arg\max_{\phi} Q(\phi|\phi(\ell)) \ , \\ Q(\phi|\phi(\ell)) &= I\!E_{\boldsymbol{C}|\mathcal{X}} \left[\log p_{\phi}(\mathcal{X}, \boldsymbol{C}) \mid \phi(\ell)\right] \ , \end{aligned} \tag{2.4}$$

where $\boldsymbol{C} = \{C_1, ..., C_N\}$ is a series of i.i.d. variables related to C. Q is an expected log-likelihood with respect to $p_{\phi(\ell)}(\boldsymbol{C} \mid \mathcal{X})$. The EM algorithm is an iterative procedure. From an initial estimate $\phi(0)$, it computes successively $\phi(0) \to ... \to \phi(\ell) \to ....$ The marginal likelihood series $\{p_{\phi(\ell)}(\mathcal{X}), \ell = 0, 1, ...\}$ is non-decreasing.

• *Bayesian network.*

The previous classical formalism is the primal version for mixture modeling in the context of classification [4]. The EM algorithm also applies to more complex situations such as those where the $\mathbf{X}_i$ are not i.i.d. variables, but are dependent through hidden variables C_i governed by a Markov chain [2] or a Markov random field [3]. In this article, we remain in the case where $\mathcal{X}$ is a sample from i.i.d. variables, but we consider a Markov structure for the dependence of the components X^j. This Markovian structure is based on a DAG denoted $G = (V, E)$. $V = \{1, ..., j, ..., n\}$ denotes the variable numbers. The edges $E \subset V \times V$ are directed : $(j', j) \in E$ is denoted $j' \to j$. The set $\bar{j} = \{j' : \ j' \to j\}$ denotes the parents of the node j. The DAG structure has a fundamental property due to its acyclic nature : there is a numbering of the nodes such that $\bar{j} \subset \{1, 2, ..., j-1\}$. We assume that the nodes have been ordered in this way. With this property and that of Markov, we get the factorization

$$p(\mathbf{x}) = \prod_j p(x^j \mid x^{\bar{j}}). \tag{2.5}$$

The set $B = (\mathbf{X}, G, \{p(x^j \mid x^{\bar{j}}\})$ is called Bayesian network. When the distribution $p(\mathbf{x})$ is non homogeneous, a mixture model as (2.2) can be considered in which $p_{\theta_k}(\mathbf{x} \mid C = k)$ denotes a Bayesian network conditional on the hidden class C. This mixture model has been investigated in [6] with a particular interest for DAG structure estimation.

2.2. Mixture model, composite class and Bayesian network

2.2.1. Composite class model

Let a partition of V composed of $M+1$ macro-nodes : $V = J_0 \uplus J_1 \uplus ... \uplus J_M$, built from the DAG structure : J_m is a macro-node if all its nodes have the same parents (In Fig.1, $M = 3$, and $J_0 = \{1\}$, $J_1 = \{2, 3\}$, $J_2 = \{4, 5\}$, $J_3 = \{6, 7, 8\}$). J_0 is the root of the tree and most often is a single node [2]. Let $J_{\overline{1}}, ..., J_{\overline{M}}$ be the parents of $J_1, ..., J_M$, respectively. Given the definition of macro-nodes, each $J_{\overline{m}}$ is composed of a single macro-node (In Fig.1, $J_{\overline{1}} = J_0$, $J_{\overline{2}} = J_1$, $J_{\overline{3}} = J_1$). The macro-nodes $\mathcal{V} = \{J_m\}$ and their connexions $\mathcal{E}$ induced by $\{J_{\overline{m}}\}$ define a new directed acyclic graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ called macro-DAG.

[2] J_0 has only one class and therefore $\nu_0 = 1$.

Given a set of specifications $\{p(x^{J_m} \mid x^{\bar{J}_m}\}$ for B, a Bayesian network $\mathcal{B} = (\mathbf{X}, \mathcal{G}, \{p(x^{J_m} \mid x^{\bar{J}_m}\})$ can be defined for the macro-variables $\{X_m^J\}_{m=0}^M$. The difference with B is essentially that $\mathcal{B}$ is a vectorial process whose factorization formula is written as

$$p(\mathbf{x}) = \prod_m p(x^{J_m} \mid x^{\bar{J}_m}). \tag{2.6}$$

The factorization (2.6) assumes that the probability distribution is homogeneous, whereas it is not the case in our context. The distribution is depending on a hidden class variable $\mathbf{C}$, which implies that $p(\mathbf{x})$ is a mixture of distributions.

Firstly we assume that each macro-variable X_m^J is characterized by ν_m classes, called *elementary classes*, whose parameters are denoted $\theta^m = \{\theta_1^m, ..., \theta_{\nu_m}^m\}$. If we forget for a while the DAG structure, then each variable taken independently of the others, is defined by a mixture model for which (2.1) is rewritten as

$$p(x^{J_m}) = \sum_{k=1}^{\nu_m} P(C^m = k)\, p_{\theta_k^m}(x^{J_m} \mid C^m = k). \tag{2.7}$$

Secondly, we consider the indicator variable of *composite classes* $\mathbf{C} = (C^1, ..., C^M)$ with values in the set of M-tuples $\mathcal{K} = \{\mathbf{k} = (k_1, ..., k_M)\}$ where $k_m \in \{1, ..., \nu_m\}$, as represented in Table 1. The classification is to assign a composite class to each observation $\mathbf{x}$. This involves selecting an elementary class k_m for each macro-variable. An immediate solution would be to perform M independent classifications, based on (2.7) but this approach would have the disadvantage of not considering the DAG structure. Therefore we must address the classification as a whole.

Considering the DAG structure, a composite class $\mathbf{k}$ is not only defined by the parameters $\theta_{\mathbf{k}} = \{\theta_{k_m}^m\}_{m=1}^M$ of its elementary classes, but also by the dependency parameters $\bar{\theta}_{\mathbf{k}} = \{\bar{\theta}_{k_m}^m\}_{m=1}^M$ that define the specifications of the Bayesian network $\mathbf{X}$ conditionally to $\mathbf{C} = \mathbf{k}$ [3]. These parameters are related to the parameters $\theta_{\mathbf{k}}$. For each composite class, the factorization formula (2.6) based on the macro-DAG is written as

$$p_{\bar{\theta}_{\mathbf{k}}}(\mathbf{x} \mid \mathbf{k}) = p(x^{J_0}) \prod_{m=1}^{M} p_{\bar{\theta}_{\mathbf{k}}^m}(x^{J_m} \mid x^{J_{\overline{m}}}, k_m, k_{\overline{m}}) \,, \tag{2.8}$$

where $k_{\overline{m}}$ denotes the class number associated to $x^{J_{\overline{m}}}$ and appearing in $\mathbf{k}$. In the notation $p_{\bar{\theta}_{\mathbf{k}}^m}$, only the classes k_m and $k_{\overline{m}}$ of $\mathbf{k}$ are active. Finally the mixture model is written as

$$p_{\bar{\theta}}(\mathbf{x}) = \sum_{\mathbf{k} \in \mathcal{K}} \alpha_{\mathbf{k}}\, p_{\bar{\theta}_{\mathbf{k}}}(\mathbf{x} \mid \mathbf{k}) \,. \tag{2.9}$$

Initially in (2.7) the definition of elementary classes has been made independently for each macro-variables. Now, the Markov dependence (2.8) introduces dependencies among these classes. The parameter setting of the mixture model (2.9) differs from the classical mixture model (2.1).

[3] In this paper, the notation $\bar{.}$ is reserved to parameters associated to the DAG dependencies.

Two M-tuples may have common components. For example, all components of $(1, k_2, ..., k_M)$ and $(2, k_2, ..., k_M)$ are identical, except the first. Thus, since two M-tuples may have common components, two components of the mixture may have common parameters[4]. In fact, there is one parameter setting per class, totaling $\sum_m \nu_m$ settings, while there are $|\mathcal{K}| = \prod_m \nu_m$ composite classes.

2.2.2. EM-mDAG algorithm

The ultimate objective is to assign a composite class to every observation $\mathbf{x}$:

$$\mathbf{x} \to \widehat{\mathbf{k}}(\mathbf{x}) = \arg\max_{\mathbf{k}\in\mathcal{K}} P_\phi(\mathbf{C} = \mathbf{k} \mid \mathbf{x}) \, .$$

Therefore, it is necessary to estimate $\phi = (\alpha, \bar\theta)$. In an equivalent manner to (2.4), the estimation of ϕ is based on the log-likelihood by maximizing the Lagrangian function

$$\begin{aligned} \mathcal{L}(\alpha, \bar\theta) &= \sum_{i=1}^N \log \left[\sum_{\mathbf{k}\in\mathcal{K}} \alpha_{\mathbf{k}} \, p_{\bar\theta_{\mathbf{k}}}(\mathbf{x}_i \mid \mathbf{k}) \right] + \lambda \left[\sum_{\mathbf{k}\in\mathcal{K}} \alpha_{\mathbf{k}} - 1 \right] \, , \\ &= \sum_{i=1}^N \log \left[\sum_{\mathbf{k}\in\mathcal{K}} \alpha_{\mathbf{k}} \prod_{m=1}^M p_{\bar\theta^m_{\mathbf{k}}}(x_i^{J_m} \mid x_i^{J_{\overline{m}}}, k_m, k_{\overline{m}}) \right] + \lambda \left[\sum_{\mathbf{k}\in\mathcal{K}} \alpha_{\mathbf{k}} - 1 \right] \, , \end{aligned} \tag{2.10}$$

where λ denotes the Lagrangian parameter associated to the constraint $\sum_{\mathbf{k}\in\mathcal{K}} \alpha_{\mathbf{k}} = 1$. At the iteration ℓ of the EM algorithm, the re-estimation formula of α is written as in the classical case :

$$\alpha_{\mathbf{k}}(\ell + 1) = \frac{1}{N} \sum_i p_{\phi(\ell)}(\mathbf{k} \mid \mathbf{x}_i) \, , \tag{2.11}$$

where the a posteriori probability of the composite class $\mathbf{k}$ is defined by

$$p_{\phi(\ell)}(\mathbf{k} \mid \mathbf{x}_i) = \frac{\alpha_{\mathbf{k}}(\ell) \, p_{\bar\theta_{\mathbf{k}}(\ell)}(\mathbf{x}_i \mid \mathbf{k})}{p_{\phi(\ell)}(\mathbf{x}_i)} = \frac{\alpha_{\mathbf{k}}(\ell) \, p_{\bar\theta_{\mathbf{k}}(\ell)}(\mathbf{x}_i \mid \mathbf{k})}{\sum_{\mathbf{k}\in\mathcal{K}} \alpha_{\mathbf{k}}(\ell) p_{\bar\theta_{\mathbf{k}}(\ell)}(\mathbf{x}_i \mid \mathbf{k})} \, . \tag{2.12}$$

As we said above, the peculiarity of this variant of the EM algorithm is the fact that a same parameter $\bar\theta^m_{k_m}$ can be present in several composite classes. In the classical case (2.1), the gradient of the Lagrangian function with respect to θ_k concerns only p_{θ_k} while in (2.10), the gradient with respect to $\bar\theta^m_{k_m}$ relates to several components $p_{\bar\theta_{\mathbf{k}}}$.

Proposition 1. *The re-estimation formula of $\bar\theta$ is given by the solution $\bar\theta(\ell + 1)$ of the linear system*

$$\sum_{i=1}^N \sum_{\substack{\mathbf{k}=(k_1,...,k_M): \\ k_m=\tau_m}} p_{\phi(\ell)}(\mathbf{k} \mid \mathbf{x}_i) \, \frac{\partial}{\partial \bar\theta^m_{\tau_m}} \log p_{\bar\theta^m_{\tau_m}}(x_i^{J_m} \mid x_i^{J_{\overline{m}}}, \tau_m, k_{\overline{m}}) \Big|_{\bar\theta=\bar\theta(\ell+1)} = 0 \, , \tag{2.13}$$

$$\tau_m = 1, ..., \nu_m \, , \; m = 1, ..., M.$$

[4] In the classical case, several components can also be concerned by a same parameter, for instance the same variance in the Gaussian case, but it is not a constraint contrary to our mixture model where many parameters are necessarily shared.

Proof. Let's focus on $\bar{\theta}^m_{\tau_m}$ where τ_m is a particular class number in $\{1, ..., \nu_m\}$.

$$\begin{aligned}
\frac{\partial \mathcal{L}(\alpha, \bar{\theta})}{\partial \bar{\theta}^m_{\tau_m}} &= \sum_{i=1}^{N} \frac{1}{p_\phi(\mathbf{x}_i)} \left[\sum_{\substack{\mathbf{k}=(k_1,\ldots,k_M):\\ k_m=\tau_m}} \alpha_{\mathbf{k}} \, \frac{\partial}{\partial \bar{\theta}^m_{\tau_m}} p_{\bar{\theta}_{\mathbf{k}}}(\mathbf{x}_i \mid \mathbf{k}) \right] , \\
&= \sum_{i=1}^{N} \frac{1}{p_\phi(\mathbf{x}_i)} \left[\sum_{\substack{\mathbf{k}=(k_1,\ldots,k_M):\\ k_m=\tau_m}} \alpha_{\mathbf{k}} \, \frac{p_{\bar{\theta}_{\mathbf{k}}}(\mathbf{x}_i \mid \mathbf{k})}{p_{\bar{\theta}_{\mathbf{k}}}(\mathbf{x}_i \mid \mathbf{k})} \, \frac{\partial}{\partial \bar{\theta}^m_{\tau_m}} p_{\bar{\theta}_{\mathbf{k}}}(\mathbf{x}_i \mid \mathbf{k}) \right] , \\
&= \sum_{i=1}^{N} \sum_{\mathbf{k}: k_m=\tau_m} p_\phi(\mathbf{k} \mid \mathbf{x}_i) \, \frac{\partial}{\partial \bar{\theta}^m_{\tau_m}} \log p_{\bar{\theta}_{\mathbf{k}}}(\mathbf{x}_i \mid \mathbf{k}) \, .
\end{aligned} \tag{2.14}$$

Recalling the factorization formula (2.5), the gradient can be written as

$$\begin{aligned}
\frac{\partial \mathcal{L}(\alpha, \bar{\theta})}{\partial \bar{\theta}^m_{\tau_m}} &= \sum_{i=1}^{N} \sum_{\mathbf{k}: k_m=\tau_m} p_\phi(\mathbf{k} \mid \mathbf{x}_i) \, \frac{\partial}{\partial \bar{\theta}^m_{\tau_m}} \left[\sum_{a=1}^{M} \log p_{\bar{\theta}^a_{\mathbf{k}}}(x_i^{J_a} \mid x_i^{J_{\overline{a}}}, k_a, k_{\overline{a}}) \right] , \\
&= \sum_{i=1}^{N} \sum_{\mathbf{k}: k_m=\tau_m} p_\phi(\mathbf{k} \mid \mathbf{x}_i) \, \frac{\partial}{\partial \bar{\theta}^m_{\tau_m}} \left[\log p_{\bar{\theta}^m_{\tau_m}}(x_i^{J_m} \mid x_i^{J_{\overline{m}}}, \tau_m, k_{\overline{m}}) \right] ,
\end{aligned} \tag{2.15}$$

which leads after a shortcut, to the system (2.13). □

2.2.3. Gaussian case, linear dependency model and DAG

• *Linear dependency model and DAG.*

Under the Gaussian assumption, conditionally on the elementary classes, the law of the macro-variables are

$$X^{J_m|k_m} \doteq [X^{J_m} \mid k_m] \sim \mathcal{N}(\mu^m_{k_m}, \Gamma^m_{k_m}) \, , \tag{2.16}$$

and with respect to the DAG, the transition laws among these variables are

$$[X^{J_m|k_m} \mid x^{J_{\overline{m}}}, k_{\overline{m}}] \; = \; [X^{J_m} \mid x^{J_{\overline{m}}}, k_m, k_{\overline{m}}] \sim \mathcal{N}(\mu^{m,\mathbf{x}}_{k_m|k_{\overline{m}}}, \Gamma^{m,\mathbf{x}}_{k_m|k_{\overline{m}}}) \, . \tag{2.17}$$

We assume the linear regression model

$$\begin{aligned}
\mu^{m,\mathbf{x}}_{k_m|k_{\overline{m}}} &= A^m_{k_m,k_{\overline{m}}} \, x^{J_{\overline{m}}} + b^m_{k_m,k_{\overline{m}}} \, , \\
\Gamma^{m,\mathbf{x}}_{k_m|k_{\overline{m}}} &= \Gamma^m_{k_m|k_{\overline{m}}} \, .
\end{aligned} \tag{2.18}$$

Therefore, the respective parameter settings of (2.16) and (2.17) are respectively

$$\begin{aligned}
\theta^m_{k_m} &= \{\mu^m_{k_m}, \Gamma^m_{k_m}\} \, , \\
\bar{\theta}^m_{k_m} &= \{A^m_{k_m,k_{\overline{m}}}, \; b^m_{k_m,k_{\overline{m}}}, \; \Gamma^m_{k_m|k_{\overline{m}}}\} \, .
\end{aligned}$$

Note that the linear regression model (2.18) depends on the direction of the DAG. $A^m_{k_m,k_{\overline{m}}}$ characterizes the regression of X^{J_m} on $x^{J_{\overline{m}}}$ and not the reverse. Note also that the regression model is multidimensional in output since $X^{J_m|k_m}$ is a random vector in $\mathbf{R}^{|J_m|}$.

• *Re-estimation formulas.*

The first approach to obtain these formulas would be to use (2.13) taking into account the Gaussian log-density that for any observation $\mathbf{x}_i$ can be written as :

$$\log p_{\bar\theta^m_{k_m}}(x_i^{J_m} \mid x_i^{J_{\overline{m}}}, k_m, k_{\overline{m}}) = cst - \frac{1}{2}|\log \Gamma^m_{k_m|k_{\overline{m}}}| \\ - \frac{1}{2}(x_i^{J_m} - \mu^{m,\mathbf{x}_i}_{k_m|k_{\overline{m}}})'(\Gamma^m_{k_m|k_{\overline{m}}})^{-1}(x_i^{J_m} - \mu^{m,\mathbf{x}_i}_{k_m|k_{\overline{m}}}) .$$

Establish equation (2.13) requires to derivate with respect to all the components of $\bar\theta^m_{k_m}$. For instance, the partial derivative with respect to $A^m_{\tau_m,k_{\overline{m}}}$ is written as

$$\frac{\partial}{A^m_{\tau_m,k_{\overline{m}}}} \log p_{\bar\theta^m_{\tau_m}}(x_i^{J_m} \mid x_i^{J_{\overline{m}}}, \tau_m, k_{\overline{m}}) = (\Gamma^m_{\tau_m,k_{\overline{m}}})^{-1} x_i^{J_{\overline{m}}}(x_i^{J_m} - \mu^{m,\mathbf{x}_i}_{\tau_m,k_{\overline{m}}})' .$$

We could go on, but there is a more direct way to operate.

Proposition 2. *Since (2.18) is a linear regression model, it is easier to use the classical formulas of the maximum likelihood estimation of this model. In our context, these formulas [5] are written as*

$$b^m_{\tau_m,k_{\overline{m}}}(\ell+1) = \widehat{I\!E}(X^{J_m|\tau_m}) - A^m_{\tau_m,k_{\overline{m}}}(\ell+1)\widehat{I\!E}(\ X^{J_{\overline{m}}|k_{\overline{m}}}) ,$$

$$A^m_{\tau_m,k_{\overline{m}}}(\ell+1) = \widehat{Cov}(X^{J_m|\tau_m},\ X^{J_{\overline{m}}|k_{\overline{m}}}) \left[\widehat{Var}(X^{J_{\overline{m}}|k_{\overline{m}}})\right]^{-1} ,$$

where $\widehat{\cdot}$ denotes an empirical estimate.

However, the empirical estimate of the moments (expectations, covariance matrix and variance-covariance matrix) must be weighted by weights w derived from the DAG :

$$b^m_{\tau_m,k_{\overline{m}}}(\ell+1) = \sum_{i=1}^N \sum_{\mathbf{k}:k_m=\tau_m} w^{\tau_m}_{i,\mathbf{k}}(\ell)\ x_i^{J_m} \\ - A^m_{\tau_m,k_{\overline{m}}}(\ell+1) \sum_{i=1}^N \sum_{\mathbf{k}:k_m=\tau_m} w^{\tau_m}_{i,\mathbf{k}}(\ell)\ x_i^{J_{\overline{m}}} , \tag{2.19}$$

where the weights at iteration ℓ are

$$w^{\tau_m}_{i,\mathbf{k}}(\ell) = \frac{p_{\phi(\ell)}(\mathbf{k} \mid \mathbf{x}_i)}{\sum_{i=1}^N \sum_{\mathbf{k}:k_m=\tau_m} p_{\phi(\ell)}(\mathbf{k} \mid \mathbf{x}_i)} .$$

Similarly, by denoting $\hat{\mu}^{J_m|\tau_m} = \widehat{I\!E}(X^{J_m|\tau_m})$, we have

$$A^m_{\tau_m,k_{\overline{m}}}(\ell+1) = \sum_{i=1}^{N} \sum_{\mathbf{k}:k_m=\tau_m} w^{\tau_m}_{i,\mathbf{k}}(\ell)\ (x_i^{J_{\overline{m}}} - \hat{\mu}^{J_{\overline{m}}|\tau_{\overline{m}}})(x_i^{J_m} - \hat{\mu}^{J_m|\tau_m})' \times \left[\sum_{i=1}^{N} \sum_{\mathbf{k}:k_m=\tau_m} w^{\tau_m}_{i,\mathbf{k}}(\ell)\ (x_i^{J_{\overline{m}}} - \hat{\mu}^{J_{\overline{m}}|\tau_{\overline{m}}})(x_i^{J_{\overline{m}}} - \hat{\mu}^{J_{\overline{m}}|\tau_{\overline{m}}})'\right]^{-1} . \tag{2.20}$$

Finally, we have also

$$\Gamma^m_{\tau_m,k_{\overline{m}}}(\ell+1) = \sum_{i=1}^{N} \sum_{\mathbf{k}:k_m=\tau_m} w^{\tau_m}_{i,\mathbf{k}}(\ell) \left(x_i^{J_m} - \mu^m_{\tau_m,k_{\overline{m}}}(\ell+1)\right) \left(x_i^{J_m} - \mu^m_{\tau_m,k_{\overline{m}}}(\ell+1)\right)' . \tag{2.21}$$

Note that the programming of the re-estimation formulas (2.19, 2.20, 2.21) is relatively difficult because two data structures interfere : the dependency structure derived from the DAG, and the list structure of the composite classes (cf. (1) for instance).

• *Back to the elementary class parameters.*

For all $\mathbf{x}_i$, the estimated composite class $\widehat{\mathbf{k}}(\mathbf{x}_i) = (\widehat{k}_1(\mathbf{x}_i), ..., \widehat{k}_M(\mathbf{x}_i))$ has been computed. The user is also interested by the parameters $\theta_{\widehat{\mathbf{k}}}$ of the elementary classes $\{\widehat{k}_m(\mathbf{x}_i)\}_{m=1}^M$ in order to interpret the leaves of the tree. The law (2.16) ignores the DAG, contrary to the law (2.17). However, in the Gaussian case, the parameters $\theta_{\mathbf{k}}$ are related to the parameters $\bar{\theta}_{\mathbf{k}}$ as follows [5] :

$$\begin{aligned} \mu^{m,\mathbf{x}}_{k_m|k_{\overline{m}}} &= \mu^m_{k_m} + \Gamma^m_{k_m,k_{\overline{m}}} (\Gamma^{\overline{m}}_{k_{\overline{m}}})^{-1} (x^{J_{\overline{m}}} - \mu^{\overline{m}}_{k_{\overline{m}}}) , \\ \Gamma^m_{k_m|k_{\overline{m}}} &= \Gamma^m_{k_m} + \Gamma^m_{k_m,k_{\overline{m}}} (\Gamma^{\overline{m}}_{k_{\overline{m}}})^{-1} \Gamma^{\overline{m}}_{k_{\overline{m}},k_m} . \end{aligned} \tag{2.22}$$

To avoid the difficulty of solving the system with respect to $\mu^m_{k_m}$ and $\Gamma^m_{k_m}$, we consider

$$\begin{aligned} \hat{\mu}^m_{k_m} &= \frac{1}{N} \sum_i x_i^{J_m} \mathbf{1}_{\widehat{k}_m(\mathbf{x}_i)=k_m} , \\ \hat{\Gamma}^m_{k_m} &= \frac{1}{N} \sum_i (x_i^{J_m} - \hat{\mu}^m_{k_m})(x_i^{J_m} - \hat{\mu}^m_{k_m})' \mathbf{1}_{\widehat{k}_m(\mathbf{x}_i)=k_m} . \end{aligned}$$

• *Initial solution.*

The solution at the first step of the EM-mDAG algorithm is obtained by performing M independent classifications using the conventional EM algorithm. Therefore, for each macro-variable, we have ν_m clusters in $\mathbf{R}^{|J_m|}$ whose labels are $\{\widehat{k}_m(x_i^{J_m}),\ i = 1, ..., n\}$. From there, the initial solution $\bar{\theta}^m_{k_m,k_{\overline{m}}}(0)$ at iteration $\ell = 0$ is computed using ordinary linear regression for every pair of clusters $(k_m, k_{\overline{m}})$ for which there are observations : $\{i : \widehat{k}_m(x_i^{J_m}) = k_m,\ \widehat{k}_{\overline{m}}(x_i^{J_{\overline{m}}}) = k_{\overline{m}}\} \neq \emptyset$. Starting from this initial solution, the role of the EM-mDAG algorithm is to re-organize the clusters in order to extract from $\mathcal{K}$ a set of composite classes of high likelihood.

Table 2. Expectations $\mu_{\mathbf{k}}$ of the 5 composite classes $\mathcal{K}_0$ for data simulation with $M = 3$, $\nu_0 = 1$, $\nu_1 = 2$, $\nu_2 = 3$, $\nu_3 = 4$. Here $c = 1.5$.

	$\mu_{\mathbf{k}}$:
X_1 :	0, 0, 0, 0, 0
X_2 :	-c, -c, +c, +c, +c
X_3 :	+c, +c, -c, -c, -c
X_4 :	-c, -c, +c, +c, +c
X_5 :	-c, -c, -c, +c, +c
X_6 :	-c, +c, -c, +c, +c
X_7 :	-c, +c, +c, -c, -c
X_8 :	-c, +c, -c, +c, +c

3. Experiments on simulated data

The random vector $\mathbf{X}$ of dimension $n = 8$ consists of $M = 3$ macro-variables with $\nu_1 = 2$, $\nu_2 = 3$, $\nu_3 = 4$. Among the $|\mathcal{K}| = 24$ potential composite classes, only 5 significant composite classes $\mathcal{K}_0$ were considered. It implies that for $\mathbf{k} \notin \mathcal{K}_0$ no data has been generated, or in another words

$$\alpha_{\mathbf{k}} = 0, \ \forall \, \mathbf{k} \notin \mathcal{K}_0. \tag{3.1}$$

Table 2 gives the expectation of these 5 composite classes and therefore the expectation of the elementary classes within the 3 macro-variables.

Fig.4 shows the observations of the macro-variables with their labels. This simulation was inspired by the cytometry data analysis domain (see Appendix) but with much more overlapping of the elementary classes. The first macro-variable $X^{J_1} = (X^2, X^3)$ shows two groups that it is possible to manually split, giving rise to two elementary classes denoted X^2+ and X^3+. Each group is a mixture that the other two macro-variables help to identify. The macro-variable $X^{J_2} = (X^4, X^5)$ highlights the components of the group X^2+, while the macro-variable $X^{J_3} = (X^6, X^7, X^8)$ highlights the components of the group X^3+. However the overlapping of the mixture components in the groups X^2+ and X^3+ does not allow a partitionning of these groups as easy as for X^{J_1}. Therefore we must address the classification as a whole.

• *Data processing.*

Fig.5 shows the initial solution of the EM-mDAG algorithm at step $\ell = 0$. This solution results from $M = 3$ independent classifications by applying the classical EM algorithm on each macro-variable. The class labels $\hat{k}(x_i^{J_m})$ defined in (2.3) are gathered for making composite class labels by using a table similar to Table 1. This initial solution is unsatisfactory. The macro-variables X^{J_2} and X^{J_3} are strongly blurred by several small composite classes that are artefacts. The final solution of the EM-mDAG algorithm is shown in Fig.6. The representation in terms of mixture components is close to the original in Fig.4. The macro-variables X^{J_2} and X^{J_2} respectively highlight the components of the group X^2+ and X^3+, despite the fifth class that is divided into two neighbour classes.

Fig.7 and Fig.8 show respectively the classifications obtained with the usual EM algorithm successively performed on the basis of 24 classes and 5 classes. With 24 classes, the number of non-empty classes is large and therefore the classification is greatly erroneous. With 5 classes, the classification provided by the EM algorithm does not meet the specificity of the data. The macro-variable $X^{J_2} = (X^4, X^5)$ does not highlight the mixture components in the group X^2+. This is a major problem that hinders significantly the interpretation of composite classes in terms of macro-variables.

• *Property of the EM-mDAG algorithm.*

The experiments show that the EM-mDAG algorithm has the property to keep a small number of α_k different from zero when there is a limited number of significant composite classes $\mathcal{K}_0 \subset \mathcal{K}$. This selection ability is not so surprising. Firstly, $\mathbf{X}$ is not observable along $\mathbf{k}$ when $\mathbf{k} \notin \mathcal{K}_0$, which means that its conditional distribution is not defined for these $\mathbf{k}$. There exists at least one couple $(k_m, k_{\overline{m}})$ in $\mathbf{k}$ whose observability of $[X^{J_m}|X^{J_{\overline{m}}}, k_m, k_{\overline{m}}]$ is undefined. At every step ℓ of the algorithm, there are several couples $(k_m, k_{\overline{m}})$ such that no observation $\mathbf{x}_i$ is simultaneously present in the clusters k_m and $k_{\overline{m}}$: $\{i : \widehat{k}_m(x_i^{J_m}) = k_m,\ \widehat{k}_{\overline{m}}(x_i^{J_{\overline{m}}}) = k_{\overline{m}}\} = \emptyset$. Secondly, the Markovian dependency introduced by the specifications $p(x_i^{J_m}|x_i^{J_{\overline{m}}}, k_m, k_{\overline{m}})$ has for effect to reorganize the initial clustering while maintaining a well-contrasted partitioning. This is a well-known property of the Markovian approach. These two remarks should be useful for establishing an analytical proof of the selection property.

4. Discussion

In this paper, we have presented a mixture model dedicated to the case where the dependencies among the components of the multidimensional random vector are governed by a DAG structure. The mixture model takes advantage of a two-level structure, which is composed by the DAG itself and its macro-representation. A dedicated EM algorithm has been efficiently implemented for the Gaussian case. The experiments show that this algorithm is able to select a small number of composite classes. This selection ability is important because it allows to circumvent the difficulty of choosing the exact number of elementary classes for each macro-variable. In fact, one of the main role of the EM-mDAG algorithm is to reveal significant relationships among the hidden elementary classes, some of them becoming empty during the procedure.

Appendix : a case study

This section presents a domain in which our method should be useful, as it is currently being undertaken by Xiaoyi Chen at Institut Pasteur (Systems Biology team). A N-sample $\mathcal{X}$ of tens and even hundreds of thousands of cells is observed by flow cytometry. For each cell $i = 1, ..., N$, the instrument provides a measurement vector $\mathbf{x}_i$ of dimension n. This sample is a mixture of several cell populations. The goal is to group these measurements so that each class corresponds to a well-identified cell type [1].

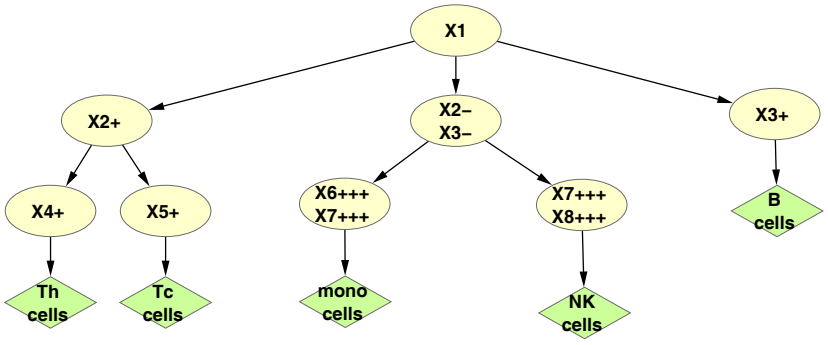


Figure 2. A partial decision tree with biological classes on leaves, (thanks to Xiaoyi Chen, Institut Pasteur).

The analysis, which is based on a dependency tree as illustrated in Fig.1-a, is usually accomplished by sequential manual partitioning (called "Gating") of the sample $\mathcal{X}$ from the top to the bottom of the tree. Rather than watching simultaneously the n dimensions, that is to say the cloud $\mathcal{X}$ in the space $\mathbf{R}^n$, the biologist works in subspaces of smaller dimensions, 1, 2 or 3, according to associations of variables X^j, here called macro-variables, as shown in Fig.1-b.

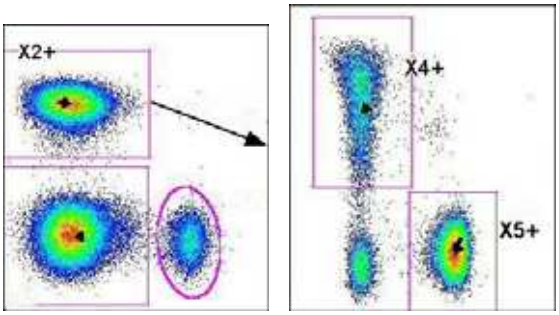
At the top of the tree, only one coordinate of the cloud $\mathcal{X}$ is analyzed. This is the variable X^1 corresponding to high values $CD45+$ of the biological variable $CD45$. In this example, to simplify, the tree height was reduced by starting the tree with $X^1 = CD45+$ instead of $(CD45-, CD45+)$. To determine the two groups $CD45-$ et $CD45+$, a threshold τ_{CD45} is manually selected for separating the small and large values of $CD45$. Conditionally on the elementary class $X^1 = CD45+$, the procedure continues along the tree structure, as follows.

Three elementary classes are extracted from the 2-D distribution of the sample $\{(x_i^2, x_i^3)\}_{i=1}^N$ and denoted (X^2+), (X^2-, X^3-), (X^3+) as illustrated in Fig.2 and Fig.3-a, On each group, this operation is repeated on the following macro-variables in dimension 2 for (X^4, X^5) conditionally on (X^2+) as illustrated in Fig.3-b, and in dimension 3 for (X^6, X^7, X^8) conditionally on (X^2-, X^3-).

This conditional and sequential procedure can be represented by a DAG and then modeled by a Bayesian network. The main advantage of using the EM-mDAG is its ability to global classification while keeping the biological dependency structure, which is necessary for identifying the cell types.

References

[1] Nima Aghaeepour, Greg Finak, The FlowCAP Consortium, The DREAM Consortium, Holger Hoos, Tim R Mosmann, Ryan Brinkman, Raphael Gottardo, and Richard H Scheuer-



(a) (b)

Figure 3. Two steps of the sequential procedure leading to the biological classes "Th cells" and "Tc cells" in Fig.2, (thanks to Xiaoyi Chen, Institut Pasteur). From the distribution of the sample $\{(x_i^2, x_i^3)\}_{i=1}^N$ shown in (a), 3 elementary classes (X^2+), (X^2-, X^3-), (X^3+) are manually extracted. (b) shows the distribution of the sample $\{(x_i^4, x_i^5)\}$ limited to the records i coming from the class (X^2+). Conditionally to (X^2+), 2 new elementary classes (X^4+) and (X^5+) are manually extracted.

mann. Critical assessment of automated flow cytometry data analysis techniques. *Nature Methods*, 10(3):228–238, 2013.

[2] Leonard E. Baum, Ted Petrie, George Soules, and Norman Weiss. A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. *The Annals of Mathematical Statistic*, 41(1):164–171, 1970.

[3] Bernard Chalmond. An iterative Gibbsian technique for the reconstruction of m-ary images. *Pattern Recognition*, 22:747–761, 1989.

[4] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, 39:1–38, 1977.

[5] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Springer, 2009.

[6] Bo Thiesson, Christopher Meek, David Maxwell Chickering, and David Heckerman. Learning mixtures of DAG models. Technical report, Microsoft Research, 1997.

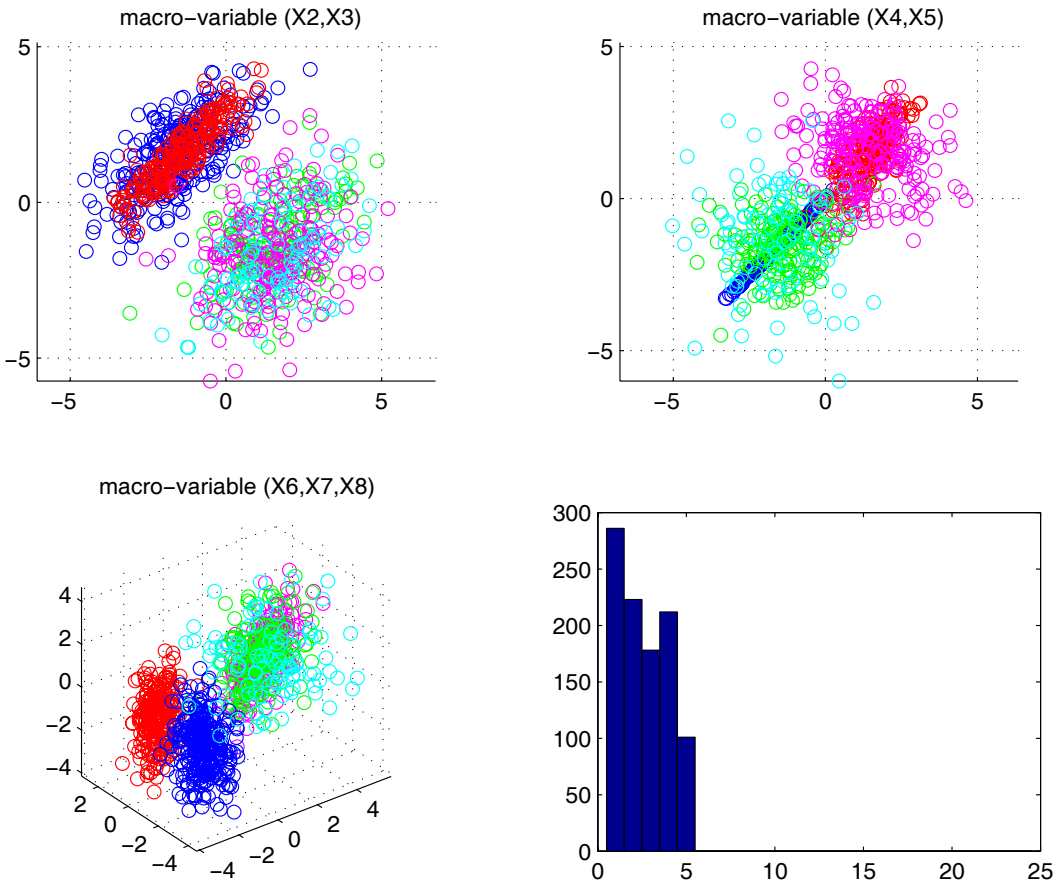


Figure 4. True labeling. There are only 5 composite classes. Simulation was performed with $\nu_1 = 2$, $\nu_2 = 3$, $\nu_3 = 4$ for a sample of size $N = 1000$. The graphic "macro-variable (X2,X3)" depicts the coordinates $\{(x_i^2, x_i^3)\}_{i=1}^N$, the "macro-variable (X4,X5)" depicts $\{(x_i^4, x_i^5)\}_{i=1}^N$, and the "macro-variable (X6,X7,X8)" depicts $\{(x_i^6, x_i^7, x_i^8)\}_{i=1}^N$. The histogram gives the empirical distribution of the composite classes.

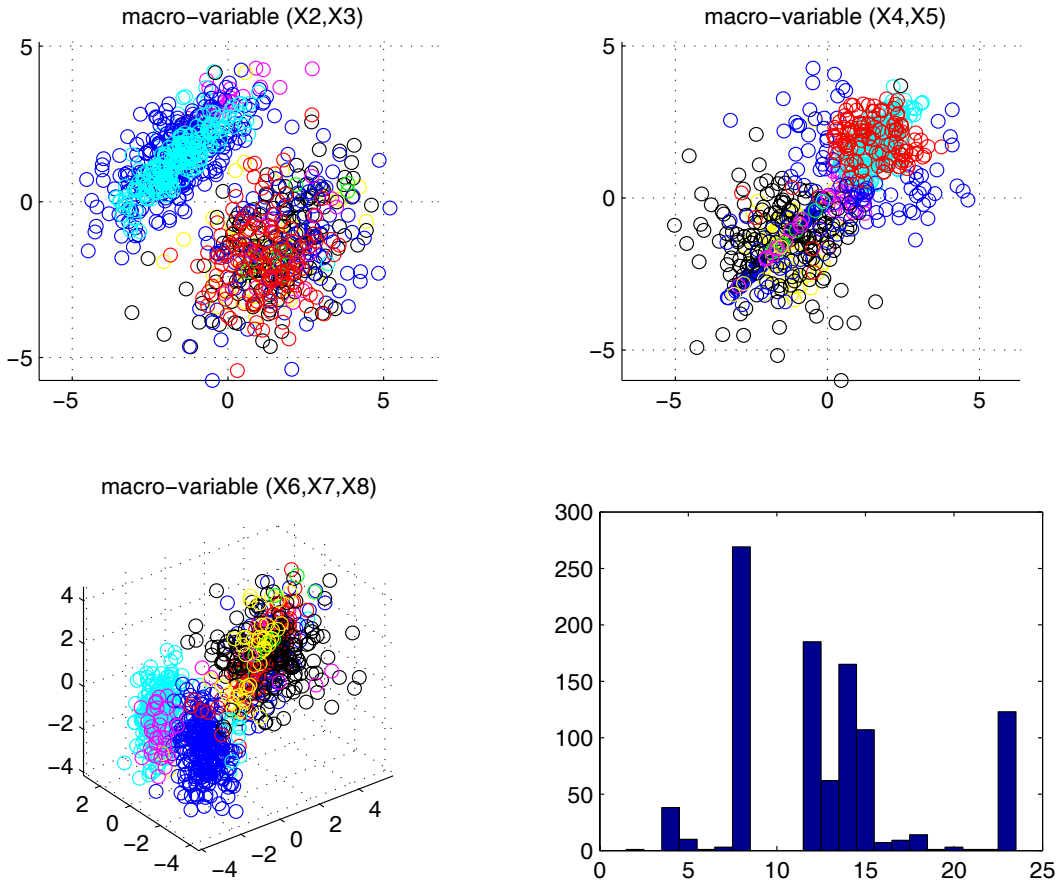


Figure 5. Initial solution of the EM-mDAG at step $\ell = 0$. $M = 3$ independent classifications was achieved by applying the classical EM algorithm on each macro-variable. Compared with the ground true in Fig.4, this representation is strongly blurred.

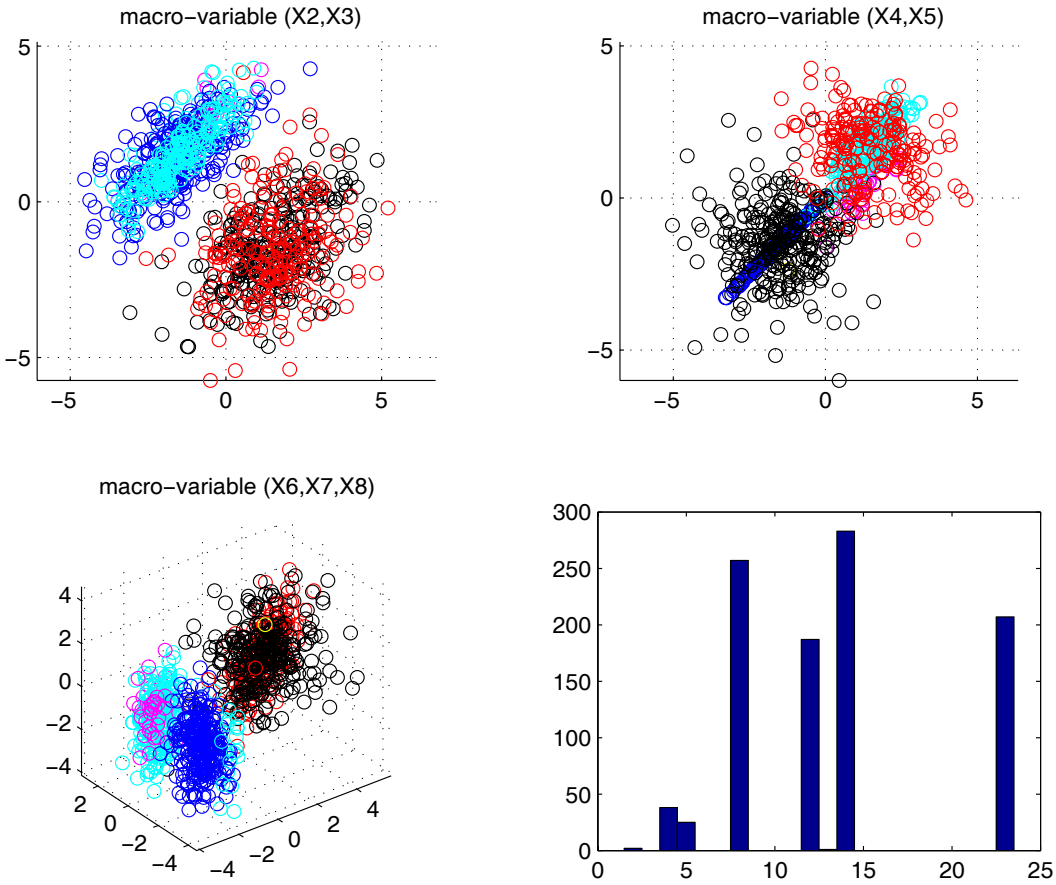


Figure 6. EM-mDAG based clustering at step $\ell = 20$. As in Fig.4, the macro-variables (X^4, X^5) and (X^6, X^7, X^8) respectively highlight the components of the group X^2+ and X^3+ of the macro-variable (X^2, X^3), despite the fifth class that is divided into two neighbour classes, numbered $\mathbf{k} = 4$ and $\mathbf{k} = 5$ in the histogram.

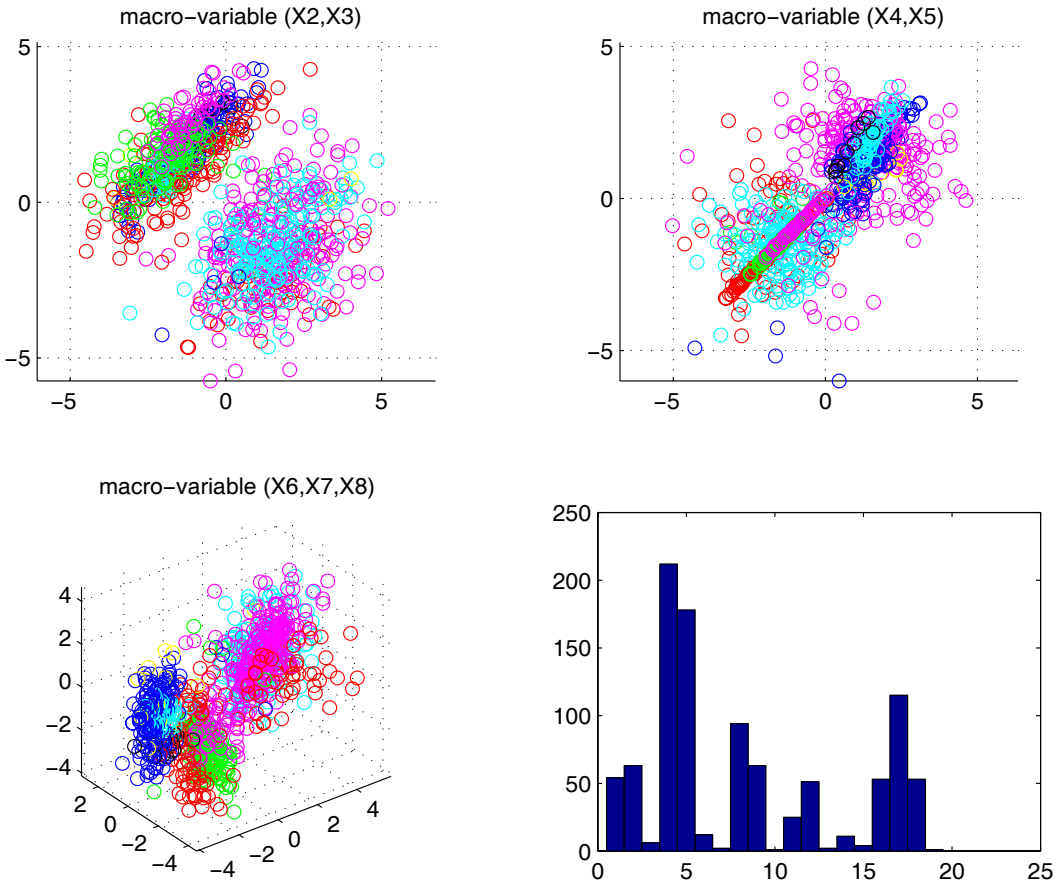


Figure 7. Standard EM algorithm for 24 classes. The number of non-empty classes is large and therefore the classification is greatly erroneous.

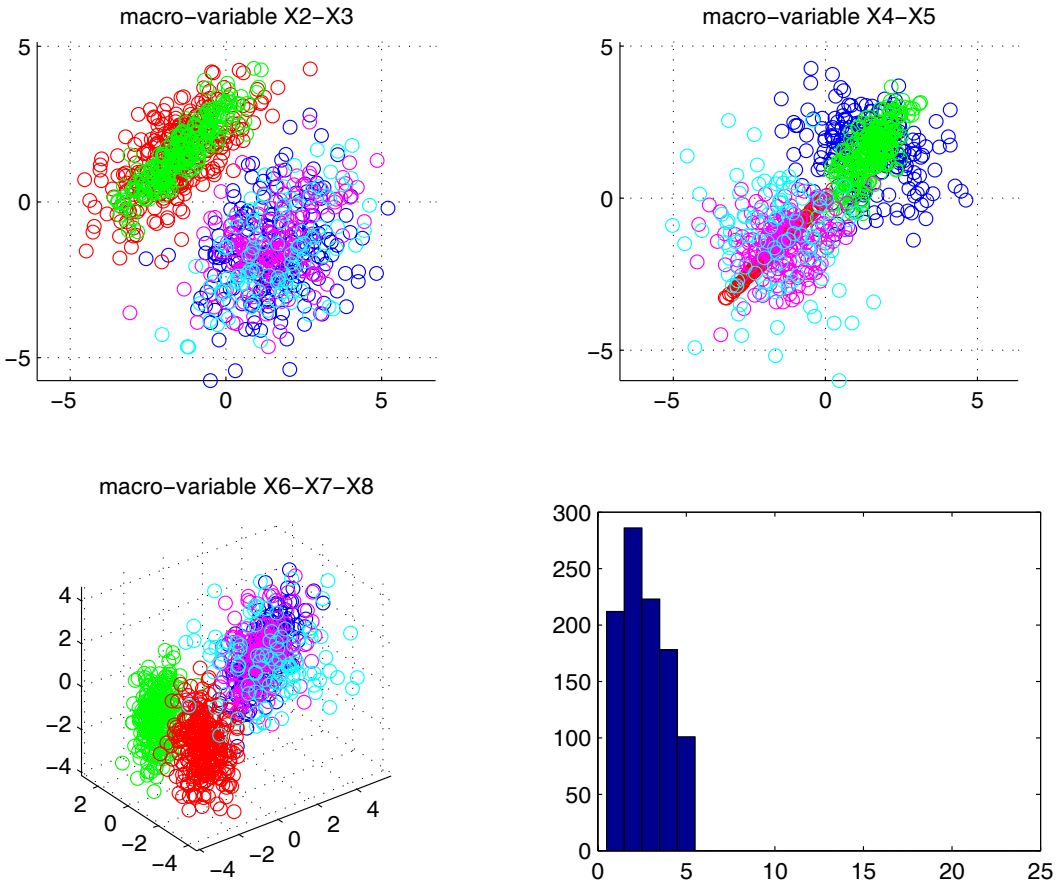


Figure 8. Standard EM algorithm for 5 classes. The macro-variable (X^4, X^5) does not highlight the mixture components of the group X^2+ of (X^2, X^3).