An EM-like algorithm for semi- and non-parametric estimation in multivariate mixtures

Tatiana Benaglia, Didier Chauveau, David R. Hunter

An EM-like algorithm for semi- and non-parametric estimation in multivariate mixtures

Tatiana Benaglia[1] Didier Chauveau[2] David R. Hunter[1,3]

December 3, 2007

1: Pennsylvania State University, USA
2: Université d'Orléans & CNRS UMR 6628, France
3: Le Studium, CNRS Orléans, France.

Abstract: We propose an algorithm that is not, but that strongly resembles, a true EM algorithm for nonparametric estimation for finite mixtures of multivariate random vectors. The vectors are assumed to have independent coordinates conditional upon knowing which mixture component from which they come, but otherwise their density functions are completely unspecified. In certain special cases, the density functions may be partially specified by Euclidean parameters, a case we call semiparametric. Our algorithm is much more flexible and easily applicable than existing algorithms in the literature; it can be extended to any number of mixture components and any number of vector coordinates of the multivariate observations. Thus it may be applied even in situations where the model is not identifiable, so care is called for when using it in situations for which identifiability is difficult to establish conclusively. Our algorithm yields much smaller mean integrated squared errors than an alternative algorithm in a simulation study. In another example using a real dataset, it provides new insights that extend previous analyses. Finally, we discuss two different variations of our algorithm, one stochastic and one deterministic, and find anecdotal evidence that there is not a great deal of difference between the performance of these two variants.

Keywords: EM algorithm, multivariate mixture, nonparametric mixture.

[1]Research partially supported by NSF Award SES-0518772

1 Introduction

Suppose the vectors $\mathbf{X}_1, \ldots, \mathbf{X}_n$ are a simple random sample from a finite mixture of $m > 1$ arbitrary distributions. The density of each $\mathbf{X}_i$ may be written

$$g_{\boldsymbol{\varphi}}(\mathbf{x}_i) = \sum_{j=1}^{m} \lambda_j \phi_j(\mathbf{x}_i), \tag{1}$$

where $\mathbf{x}_i \in \mathbb{R}^r$; $\boldsymbol{\varphi} = (\boldsymbol{\lambda}, \boldsymbol{\phi}) = (\lambda_1, \ldots, \lambda_m, \phi_1, \ldots, \phi_m)$ denotes the parameter; and the λ_j are positive and sum to unity. We assume that the ϕ_j are drawn from some family $\mathcal{F}$ of multivariate density functions (say, absolutely continuous with respect to Lebesgue measure). Model (1) is not identifiable if no restrictions are placed on $\mathcal{F}$, where by "identifiable" we mean that $g_{\boldsymbol{\varphi}}$ has a *unique* representation of the form (1) and we do not consider that "label-switching" — i.e., reordering the m pairs $(\lambda_1, \phi_1), \ldots, (\lambda_m, \phi_m)$ — produces a distinct representation.

A common restriction placed on $\mathcal{F}$, which we adopt throughout this article, is that each joint density $\phi_j(\cdot)$ is equal to the product of its marginal densities. In other words, the coordinates of the $\mathbf{X}_i$ vector are independent, conditional on the subpopulation or component (ϕ_1 through ϕ_m) from which $\mathbf{X}_i$ is drawn. Therefore, model (1) becomes

$$g_{\boldsymbol{\varphi}}(\mathbf{x}_i) = \sum_{j=1}^{m} \lambda_j \prod_{k=1}^{r} f_{jk}(x_{ik}), \tag{2}$$

where the function $f(\cdot)$, with or without subscripts, will always denote a univariate density function. If the $f_{jk}(\cdot)$ are assumed to come from a particular parametric family of densities, then standard univariate mixture model techniques (cf. MacLachlan and Peel, 2000 or Titterington et al., 1985) may easily be extended to the multivariate case. However, we wish to avoid the parametric assumption; in this article, we introduce an algorithm for estimating the parameter vector $\boldsymbol{\varphi}$ in model (2), where we do *not* assume that $f_{jk}(\cdot)$ comes from a family of densities that may be indexed by a finite-dimensional parameter vector.

Some authors (e.g., Hettmansperger and Thomas, 2000) have considered the special case of model (2) in which the density $f_{jk}(\cdot)$ does not depend on k — that is, in which the $\mathbf{X}_i$ are conditionally independent *and identically distributed*. Others (e.g., Hall and Zhou, 2003) treat equation (2) in its full generality. In order to encompass both the special case and the more general case simultaneously in this article, we introduce one further bit of notation:

We will allow that the coordinates of $\mathbf{X}_i$ are conditionally independent and there exist *blocks* of coordinates that are also identically distributed. These blocks may be of size one so that case (2) is still covered. Therefore, if we assume that the kth coordinate belongs to the b_kth block, $1 \leq b_k \leq B$, where $B \leq r$ is the total number of such blocks, then equation (2) is replaced by

$$g_{\boldsymbol{\varphi}}(\mathbf{x}_i) = \sum_{j=1}^{m} \lambda_j \prod_{k=1}^{r} f_{jb_k}(x_{ik}). \tag{3}$$

With so many different subscripts, the notation itself can become an impediment to understanding. Thus, we will remain consistent in our use of notation and terminology throughout the article. In particular, we use the terms *component* and *coordinate* only to refer, respectively, to one of the distributions (subpopulations) making up the mixture and one of the repeated measurements making up an observation. The indices i, j, k, and ℓ will always denote a generic individual, component, coordinate, and block, respectively. Therefore, we will always have $1 \leq i \leq n$, $1 \leq j \leq m$, $1 \leq k \leq r$, and $1 \leq \ell \leq B$. (Also note that m, r, and B stand for mixture components, repeated measurements, and blocks, and of course n has its usual interpretation as the sample size.)

To further elucidate model (3), consider as an example an experiment involving 405 children aged 11 to 16 years subjected to a water-level task as described by Thomas and Lohaus (1993). Each child is presented with eight rectangular vessels on a sheet of paper, each tilted to one of $r = 8$ clock-hour orientations: in order of presentation to the subjects, these orientations are 11, 4, 2, 7, 10, 5, 1, and 8 o'clock. Each vessel was on a separate sheet of paper and appeared much like the small reproductions in the plots of Figure 1 (Thomas and Lohaus, 1993, p. 40). The children's task was to draw a line representing the surface of still liquid in the closed, tilted vessel in each picture. Each such line describes two points of intersection with the sides of the vessel, and the acute angle, in degrees, formed between the horizontal and the line passing through these two points was measured for each response. The sign of each such measurement was taken to be the sign of the slope of the line.

This water-level dataset and our analysis of it will be described in further detail in Section 5; for now, we state only that we know of no other algorithm currently capable of producing similar results. Some methods have been proposed that could potentially be extended to this case (Hall et al., 2005; Qin and Leung, 2006), but they appear to be extremely complicated computationally for $m > 2$ or $r > 3$. Some other methods have been

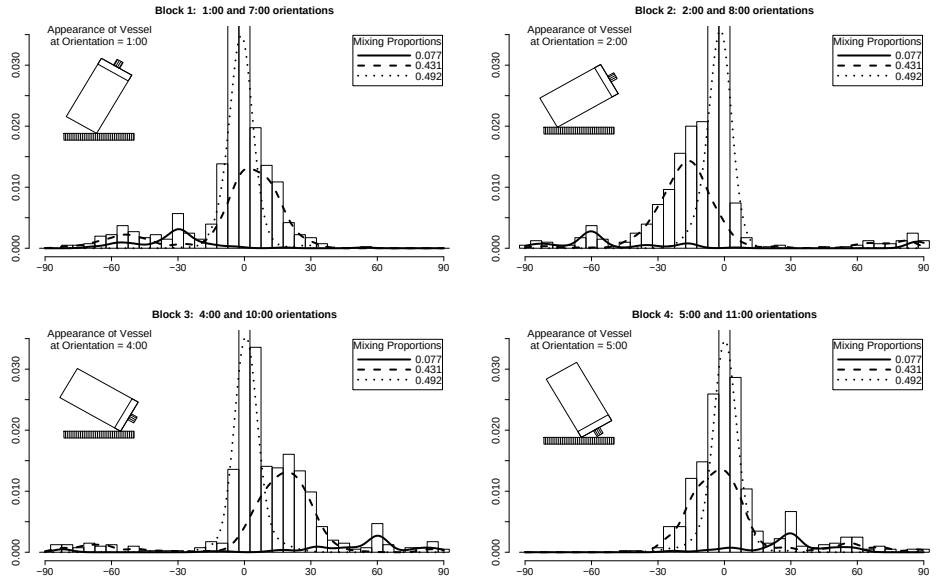

Figure 1: *The water-level data are analyzed using our algorithm, assuming model (3) with three mixture components ($m = 3$) and four coordinate blocks ($B = 4$) in which opposite clock-face orientations are assumed to lead to independent and identically distributed responses, conditional on the component.*

proposed that could handle this number of components and repeated measurements — indeed, the same dataset has been analyzed by several authors (Hettmansperger and Thomas, 2000; Cruz-Medina et al., 2004; Elmore et al., 2004) — yet all these methods rely on the assumption that the $r = 8$ coordinates are all identically distributed. Our method, by contrast, is simple to program and easily generalizable to any values m and r for which model (2) is identifiable.

In Section 2 of this article, we offer a fuller discussion of previous work in this area, including the vexing issue of identifiability. We introduce our algorithm in Section 3 and describe several modifications of the basic algorithm and model in Section 4. Section 5 is devoted to empirical study of the algorithm, both through simulation studies and through analysis of real datasets. Whenever possible in Section 5, we compare our method with results of other known methods.

2 Identifiability and previous work

An interesting question is how restrictive the assumptions on $f_{jb_k}(\cdot)$ must be in order to ensure that the model (3) is identifiable. For instance, in the univariate ($r = 1$) case, Bordes et al (2006) and Hunter et al (2007) found that when $f_j(x) = f(x - \mu_j)$ for some density $f(\cdot)$ that is symmetric about zero, the mixture density $g_{\boldsymbol{\varphi}}(x)$ admits a unique representation whenever $m \leq 3$, except in certain special cases that are easily enumerable.

In the multivariate case, Hall and Zhou (2003) showed that for two components ($m = 2$), model (2) is *always* identifiable as long as $r \geq 3$, even in the most general case in which $b_k = k$ for all k and *no* assumptions are made about the form of the $f_{jb_k}(\cdot)$. In fact, model (2) is a case in which the conditions necessary for identifiability get less restrictive as the dimension r increases; or, as Hall et al. (2005) put it, this is a case in which "from at least one point of view, the 'curse of dimensionality' works in reverse."

We use the term "nonparametric" to describe the case in which no assumptions are made about the form of the $f_{jb_k}(\cdot)$ even though the parameter $\boldsymbol{\lambda}$ is of course Euclidean. We reserve the term "semiparametric" for the case in which $f_{jb_k}(\cdot)$ is partly specified by a finite-valued parameter, such as the case discussed above in which $f_j(x) = f(x - \mu_j)$ for a symmetric but otherwise completely unspecified density $f(\cdot)$. Note that Lindsay (1995) speaks of "nonparametric mixture modeling" in a different sense: The family $\mathcal{F}$ from which the component densities come is fully specified up to a parameter $\boldsymbol{\theta}$, but the mixing distribution from which the $\boldsymbol{\theta}$ are drawn, rather than having finite support of known cardinality m as in the present article, is assumed to be completely unspecified *a priori*.

When $b_k = k$ for all k, several authors have recently addressed the problem of estimating the f_{jb_k} in model (3). Yet the estimation methods they propose appear to apply in only very limited cases. Qin and Leung (2006) and Leung and Qin (2006) adapt the exponential tilt model of Anderson (1979) and apply their methods to the cases when $m = 2$ and $r = 2$ or $r = 3$. Hall et al. (2005) give estimators based on inversion of mixture models that apply only to the case when $m = 2$ and $r = 3$. Analytical difficulties appear to preclude the application of either of these methods beyond these cases. Even in the case $r = 1$, where restrictions must be placed on $f_j(\cdot)$ in order to ensure identifiability as described at the beginning of Section 2, the estimation methods of Bordes et al. (2006) and Hunter et al. (2007) are difficult if not impossible to apply beyond the case $m = 2$. We consider a numerical example for which $r = 1$ in Section 4.3.

By contrast, in the case of continuous data when $b_k = 1$ for all k — that

is, the case of conditionally independent *and identically distributed* coordinates — several other authors (Hettmansperger and Thomas, 2000; Cruz-Medina and Hettmansperger, 2003; Elmore et al., 2006) have developed a different estimation method. This method, the *cutpoint approach*, discretizes the continuous measurements by replacing each observation $(x_{i1}, \ldots, x_{ir})$ by a multinomial vector $(n_1, \ldots, n_p)$, where

$$n_a = \sum_{k=1}^{r} I\{c_{a-1} < x_{ik} \leq c_a\}, \quad 1 \leq a \leq p,$$

and the cutpoints $-\infty = c_0 < c_1 < \cdots < c_p = \infty$ are specified by the experimenter. The cutpoint approach is completely general in the sense that it can be applied to any number of components m and any number of repeated measures r, just as long as $r \geq 2m - 1$, a condition that guarantees identifiability (Elmore and Wang, 2003). However, some information is lost in the discretization step and for this reason it becomes difficult to easily obtain density estimates of the component densities. Furthermore, even if the assumption of conditional independence is warranted, the extra assumption of identically distributed coordinates may not be; and the cutpoint method collapses when the coordinates are not identically distributed.

Here, we take a different approach and adapt an algorithm of Bordes et al (2007). Originally, this algorithm is presented as a stochastic algorithm for the particular univariate case of model (1) under the assumption that $\phi_j(x) = f(x - \mu_j)$ for some symmetric density $f(x)$. We demonstrate how to extend the algorithm to model (3) and eliminate the stochasticity. Our algorithm combines the best features of all the algorithms discussed previously: It is simple to program, it is applicable to any m and r as well as any set of blocks b_k, and it gives kernel-density-like estimates for each of the f_{jb_k}.

Yet with such flexibility also comes a bit of danger, since the identifiability question for the general model (3) has not yet been settled. Hall et al. (2005) discuss this question and give a lower bound on r, as a function of m, that is *necessary* in order to guarantee identifiability: They state that r and m should satisfy $2^r - 1 \geq mr + 1$. Yet they do not give an explicit bound that is *sufficient* to guarantee identifiability; however, Elmore et al. (2005) prove that such a (finite) upper bound exists.

Since extending our estimation method to an arbitrary number of coordinates or mixture components is very easy — unlike any previously published algorithms for the general model (3) — we are in a position in which practice is more advanced than theory. Thus, it is prudent to exercise caution when

trying to fit a model for which the identifiability question is not settled. The water-level data of Section 1 gives such an example if we take $m = 3$ or $m = 4$. We discuss this example in more detail, and give reasons that we are fairly confident about interpreting our results, in Section 5.2.

3 Estimating the parameters

We propose both a refinement and a generalization of the algorithm of Bordes et al. (2007). Although we use the term EM in connection with this algorithm, we stress that this algorithm is not an EM algorithm in the usual sense (Dempster et al., 1977) because there is no likelihood that this algorithm may be shown to maximize. However, we retain the name "EM" because the algorithm strongly resembles a true EM algorithm for the parametric mixture case, i.e., the case in which $\mathcal{F}$ is a family indexed by some Euclidean parameter. For instance, as in an EM algorithm for mixtures, we define $Z_{ij} \in \{0, 1\}$ to be a Bernoulli random variable indicating that individual i comes from component j. Since each individual comes from exactly one component, this implies $\sum_{j=1}^{m} Z_{ij} = 1$. Thus, the *complete data* is the set of all $(\mathbf{x}_i, \mathbf{Z}_i)$, $1 \le i \le n$.

3.1 The nonparametric EM algorithm

The algorithm described here is implemented in an R package (R Development Core Team, 2007) called `mixtools` (Young et al., 2005), available online from the Comprehensive R Archive Network (CRAN). Suppose we are given initial values $\boldsymbol{\varphi}^0 = (\boldsymbol{\lambda}^0, \mathbf{f}^0)$. Then for $t = 1, 2, \ldots$, we follow these three steps:

1. **E-step:** Calculate the "posterior" probabilities (conditional on the data and $\boldsymbol{\varphi}^t$) of component inclusion,

$$p_{ij}^t \stackrel{\text{def}}{=} P_{\boldsymbol{\varphi}^t}(Z_{ij} = 1 | \mathbf{x}_i) \tag{4}$$

$$= \frac{\lambda_j^t \prod_{k=1}^{r} f_{jb_k}^t(x_{ik})}{\sum_{j'=1}^{m} \lambda_{j'}^t \prod_{k=1}^{r} f_{j'b_k}^t(x_{ik})} \tag{5}$$

   for all $i = 1, \ldots, n$ and $j = 1, \ldots, m$.

2. **M-step:** Set

$$\lambda_j^{t+1} = \frac{1}{n} \sum_{i=1}^{n} p_{ij}^t \tag{6}$$

   for $j = 1, \ldots, m$.

3. **Nonparametric density estimation step:** For any real u, define for each component $j \in \{1, \ldots, m\}$ and each block $\ell \in \{1, \ldots, B\}$

$$\begin{aligned} f_{j\ell}^{t+1}(u) &= \frac{\frac{1}{h}\sum_{k=1}^{r}\sum_{i=1}^{n} p_{ij}^{t} I\{b_k = \ell\} K\left(\frac{u - x_{ik}}{h}\right)}{\sum_{k=1}^{r}\sum_{i=1}^{n} p_{ij}^{t} I\{b_k = \ell\}} \\ &= \frac{1}{nhC_\ell \lambda_j^{t+1}} \sum_{k=1}^{r}\sum_{i=1}^{n} p_{ij}^{t} I\{b_k = \ell\} K\left(\frac{u - x_{ik}}{h}\right), \quad (7) \end{aligned}$$

where $K(\cdot)$ is a kernel density function, h is a bandwidth chosen by the user, and

$$C_\ell = \sum_{k=1}^{r} I\{b_k = \ell\}$$

is the number of coordinates in the ℓth block. Note that in the case in which $b_k = k$ for all k, equation (7) becomes

$$f_{jk}^{t+1}(u) = \frac{1}{nh\lambda_j^{t+1}} \sum_{i=1}^{n} p_{ij}^{t} K\left(\frac{u - x_{ik}}{h}\right). \quad (8)$$

In the original Bordes et al (2007) algorithm, the nonparametric density estimation step differs in that p_{ij} is replaced by z_{ij}^* in equation (7), where $(z_{i1}^*, \ldots, z_{im}^*)$ is a simulated multinomial random variable with a single trial and with probability vector given by $(p_{i1}, \ldots, p_{im})$. Thus, the original algorithm has a stochastic element. In various tests, we find consistent empirical evidence that the deterministic version presented here is slightly, though not overwhelmingly, more efficient than the stochastic version. An example of such a comparison is given in section 5.3. Because the deterministic algorithm does not require any additional overhead relative to the stochastic algorithm, we use it here exclusively.

To initialize the algorithm, it is often easier to start with an initial $n \times m$ matrix $\mathbf{P}^0 = (p_{ij}^0)$ than with an initial parameter vector $\boldsymbol{\varphi}^0$. Thus, during the first iteration, we skip directly to the M-step. To obtain this $\mathbf{P}^0$ matrix, it is possible to use (say) a k-means clustering algorithm to assign each observation to one of the components. This procedure forces $\mathbf{P}^0$ to consist of just zeros and ones, but we find that it works well in practice.

3.2 Bandwidth and kernel selection

The density estimation step in the algorithm above relies on a kernel density $K(\cdot)$ and a bandwidth h. Kernel density estimation is a well-studied topic

in statistics, and for our implementation in the `mixtools` package, we tried to adopt standard techniques. In particular, because much literature on this topic suggests that the choice of a kernel function does not have a dramatic impact on the resulting density estimate, we simply take $K(t)$ to be the standard normal density function.

Choosing a bandwidth is a more complicated issue, particularly since this choice affects the density estimates dramatically. Although we do not attempt a thorough exploration of this topic in the current article, we describe here some of our experience in choosing a bandwidth.

As a default value for the bandwidth h, we simply take the entire $n \times r$ dataset, treat it as a vector of length nr, and use the default bandwidth selection of the `density` function in R — namely, a rule of thumb due to Silverman (1986, page 48) in which

$$h = 0.9(nr)^{-1/5} \min\left\{\text{SD}, \frac{\text{IQR}}{1.34}\right\}, \tag{9}$$

where SD and IQR are the standard deviation and interquartile range of all nr data values. This is a very crude method in the nonparametric mixture setting, and there are several reasons why it might produce an under- or overestimate. First, pooling all of the data implicitly treats all of the different components as though they are from the same distribution. This can lead to an inflation of the bandwidth, particularly if the mixture components' centers are well-separated, because in that case, the variability of the pooled dataset will be larger than that of the individual components. Similarly, if the vector coordinates are not identically distributed within each component, the bandwidth could be biased upward for the same reason.

Yet operating in the opposite direction is the fact that the expression nr in equation (9) is an overestimate of the "true" sample size. This is especially true when each b_k equals k — where each of the r coordinates gets a separate set of density estimates — in which case it may be sensible to eliminate the r from the equation (9) entirely. But regardless of the values of b_k, there is also the fact that the "true" sample size from each component is actually some fraction of n, namely, about $\lambda_j n$ for the jth component.

The arguments above show first of all that it would be useful to know something about the mixture structure in order to select a bandwidth. This suggests an iterative procedure in which the value of h is modified, and the algorithm reapplied, after the output from the algorithm is obtained. Secondly, there is no reason that the bandwidth should be the same for each component or even for each block: It is easy to modify equation (7) by replacing h by h_j or $h_{j\ell}$.

It is clear that a thorough exploration of the bandwidth question is therefore a research topic unto itself, so in the interest of simplicity we opt for the default value (9) in the simulation studies of Section 5.1. For the water-level data discussed in Sections 1 and 5, where the visual appearance of the density estimates is important for a qualitative appreciation of the results, we found that the default value of $h = 1.47$ produces a "bumpy-looking" set of density estimates, so we chose a larger value that gave smoother results, namely $h = 4$. We provide the very simple code for this example in Section 5.2, and we encourage interested readers to test this example using the default bandwidth. We also found in a couple other datasets that the default (9) gives somewhat "bumpy-looking" results, suggesting that the default value tends to be smaller than a more optimal choice would be in general; yet our evidence for this is only anecdotal at this point.

4 Modifications to the model and algorithm

The general model of Equation (3) and the algorithm of Section 3.1 may be modified in various ways. For example, the density-estimation bandwidth may be allowed to change for each component, each coordinate, or both, as mentioned in Section 3.2. We discuss some of these modifications here.

4.1 Location-scale model

There are some plausible models that are more restrictive than (3) but not as restrictive as the case in which all coordinates are identically distributed. For instance, if in Equation (3) we suppose that

$$f_{jb_k}(x) = \frac{1}{\sigma_{jb_k}} f_j\left(\frac{x - \mu_{jb_k}}{\sigma_{jb_k}}\right) \tag{10}$$

for unknown parameters $(\boldsymbol{\mu}_j, \boldsymbol{\sigma}_j, f_j)$, $j = 1, \ldots, m$, then the totally nonparametric specification of f_{jb_k} becomes a semiparametric specification (note that $\boldsymbol{\mu}_j$ and $\boldsymbol{\sigma}_j$ are both B-vectors). To implement the semiparametric EM algorithm in this case, equations (4) and (6) remain unchanged but it is necessary to modify equation (7) to account for the fact that *all* of the coordinates provide information about the form of each f_j. Thus, in the case (10), equation (7) is replaced by

$$f_j^{t+1}(u) = \frac{1}{nrh\lambda_j^{t+1}} \sum_{i=1}^{n} \sum_{k=1}^{r} p_{ij}^t K\left(\frac{u - x_{ik} + \mu_{jb_k}}{h\sigma_{jb_k}}\right). \tag{11}$$

Furthermore, the M-step also includes updates of the $\mu_{j\ell}$ and $\sigma_{j\ell}$ parameters for each $1 \leq j \leq m$ and $1 \leq \ell \leq B$:

$$\mu_{j\ell}^{t+1} = \frac{\sum_{i=1}^{n}\sum_{k=1}^{r} p_{ij}^{t} I\{b_k = \ell\} x_{ik}}{\sum_{i=1}^{n}\sum_{k=1}^{r} p_{ij}^{t} I\{b_k = \ell\}} = \frac{\sum_{i=1}^{n}\sum_{k=1}^{r} p_{ij}^{t} I\{b_k = \ell\} x_{ik}}{n\lambda_j^{t+1} C_\ell} \tag{12}$$

$$\sigma_{j\ell}^{t+1} = \left[\frac{1}{nC_\ell \lambda_j^{t+1}} \sum_{i=1}^{n}\sum_{k=1}^{r} p_{ij}^{t} I\{b_k = \ell\} (x_{ik} - \mu_{j\ell}^{t+1})^2 \right]^{1/2} \tag{13}$$

Naturally, it is possible to place constraints on the $\boldsymbol{\mu}_j$ or $\boldsymbol{\sigma}_j$ vectors when this is sensible. For instance, if the mixture is purely a location mixture, then we might stipulate that $\boldsymbol{\sigma}_j = \boldsymbol{\sigma}$ for each j and for some B-vector $\boldsymbol{\sigma}$. Similarly, we might stipulate that $\boldsymbol{\mu}_j = \boldsymbol{\mu}$ if the mixture is purely a scale mixture. In these latter two cases, note that we still allow the different blocks to have different scale and location parameters, though of course this may be restricted as well. Also note that because f_j is completely unconstrained (except in special cases like Section 4.3), each element of the $\boldsymbol{\mu}_j$ may only be identified up to a constant shift and each element of $\boldsymbol{\sigma}_j$ may only be identified up to a constant multiple. Stated differently, there is no loss of generality in assuming that $\sum_k \mu_{jb_k} = 0$ and $\sum_k \sigma_{jb_k} = 1$ for each j; however, when implementing the algorithm, it is generally not necessary to enforce these constraints.

4.2 Location-scale model, revisited

Note that we may obtain yet a different model by writing

$$f_{jk}(x) = \frac{1}{\sigma_{jk}} f_k \left(\frac{x - \mu_{jk}}{\sigma_{jk}} \right) \tag{14}$$

instead of equation (10). These two equations, which differ only in the replacement of a single j by k, in fact involve assumptions that are quite distinct. In one case, we are assuming that the measurements within an individual have the same shape (depending on the individual's mixture component) but may differ by a location and scale factor; in the other, we are assuming that individual differences, i.e., the mixture components, only account for differences up to a location and scale parameter, but otherwise the repeated measurements' distributions do not relate to one another in any

way. Note also that the corresponding form of Equation (7) looks somewhat different than its earlier counterpart in Equation (11):

$$f_k^{t+1}(u) = \frac{1}{nh} \sum_{i=1}^{n} \sum_{j=1}^{m} p_{ij}^t K\left(\frac{u - x_{ik} + \mu_{jb_k}}{h\sigma_{jb_k}}\right). \tag{15}$$

As a special case of both (10) and (14), we may assume that all measurements in all components have the same shape, summarized by the density $f(\cdot)$, and

$$f_{jk}(x) = \frac{1}{\sigma_{jk}} f\left(\frac{x - \mu_{jk}}{\sigma_{jk}}\right). \tag{16}$$

In case (16), equation (7) becomes

$$f^{t+1}(u) = \frac{1}{nrh} \sum_{i=1}^{n} \sum_{j=1}^{m} \sum_{k=1}^{r} p_{ij}^t K\left(\frac{u - x_{ik} + \mu_{jk}}{h\sigma_{jk}}\right). \tag{17}$$

4.3 Symmetric components

If we consider case (16) without repeated measures ($r = 1$) and for purely a location mixture ($\sigma_j = 1$, $j = 1, ..., m$), then the model becomes

$$g_{\boldsymbol{\varphi}}(x_i) = \sum_{j=1}^{m} \lambda_j f(x_i - \mu_j). \tag{18}$$

When $m = 2$, Equation (18) is exactly the location-shifted semiparametric mixture model that is proved identifiable by Bordes et al (2006) and Hunter et al (2007) under the additional assumptions that $\lambda_1 \neq 1/2$ and that the density f is symmetric about zero. This special case is also the model for which the original (semiparametric) stochastic EM algorithm is proposed in Bordes et al (2007). In the non-stochastic version, Equation (17) may be combined with a symmetrization step to give

$$f^{t+1}(u) = \sum_{i=1}^{n} \sum_{j=1}^{m} \frac{p_{ij}^t}{2nh} \left[K\left(\frac{u - x_i + \mu_j}{h}\right) + K\left(\frac{-u - x_i + \mu_j}{h}\right)\right]. \tag{19}$$

A comparison of the stochastic and non-stochastic versions of this algorithm is given in section 5.3.

4.4 Changing block structure

In Figure 1 summarizing the water-level results for three components (a dataset that is discussed further in Section 5.2), we see that the largest component, into which roughly half of the subjects fall, appears to have roughly the same density for all four blocks. We might therefore guess that for individuals in this component, observations $\mathbf{x}_i$ consist of 8 independent and identically distributed (i.i.d.) coordinates. Yet the remaining two components' observations do not appear to be identically distributed; the block structure exhibited in the plots, in which the eight coordinates fall into 4 blocks of two i.i.d. oservations each, seems appropriate. It is therefore reasonable to allow the model to encompass the possibility that the block structure could be different in each component. In other words, Equation 3 would be modified slightly to produce

$$g_{\boldsymbol{\varphi}}(\mathbf{x}_i) = \sum_{j=1}^{m} \lambda_j \prod_{k=1}^{r} f_{jb_{jk}}(x_{ik}),$$

in which the only difference is that b_k has been replaced by b_{jk} — thus, the block in which the kth observation falls depends on j as well.

Though this generalization of the model is not currently implemented in the `mixtools` package, it would be conceptually easy to do so. However, there is a theoretical issue that must be addressed in this case: Label-switching becomes a problem. By "label-switching", we mean the result of permuting the labels of the m components. When each component is assumed to follow the same model, it is not important which is labeled component 1, which is labeled component 2, etc. But if we now assume that component 1 (say) has i.i.d. coordinates whereas components 2 and 3 have a different block structure, then it is necessary to ensure that "component 1" always refers to a *particular* one of the three components. This might be easiest to achieve in practice using a two-step approach: First, obtain results for a model in which the block structure is assumed the same for all three components (as depicted in Figure 1). Then, use the final posterior probabilities of component inclusion as starting values for a second algorithm for fitting the more general model.

5 Examples for real and simulated datasets

5.1 A simulation study

We applied the nonparametric EM algorithm (npEM) to the same synthetic examples for which Hall et al. (2005) tested their estimation technique, a method based on inverting the mixture model. The three simulated models are trivariate mixtures with independent repeated measures and a shift of location between the two components in each coordinate, i.e., $m = 2$, $r = 3$, and $b_k = k$, $k = 1, ..., r$. We ran $S = 300$ replications of $n = 500$ observations each and computed the errors in terms of the square root of the Mean Integrated Squared Error (MISE) for the densities as in Hall et al. (2005), where

$$\text{MISE} = \frac{1}{S}\sum_{s=1}^{S}\int \left(\hat{f}_{jk}^{(s)}(u) - f_{jk}(u)\right)^2 du, \quad j = 1, 2 \text{ and } k = 1, 2, 3;$$

and the integral is computed numerically. Each density $\hat{f}_{jk}^{(s)}$ is computed using equation (8) together with the posterior probabilities after convergence of the algorithm, i.e., the final values of the p_{ij}^t's.

As suggested in section 3.1, we started each algorithm with an initial $n \times m$ matrix $\mathbf{P}^0 = (p_{ij}^0)$, and this matrix was determined by a k-means algorithm applied to each trivariate dataset, with initial cluster centers $(0, 0, 0)$ and $(4, 4, 4)$. This testing protocol is adapted to this particular location-shifted model in order to prevent label-switching among replications. In comparison, Hall et al. (2005) dealt with label-switching by enforcing the constraint $\hat{\lambda}_1 < \hat{\lambda}_2$. After finishing our simulation, we verified that our results would not have changed if we had used the Hall et al. approach because in every trial, we observed that $\hat{\lambda}_1 < \hat{\lambda}_2$.

To set up tuning parameters (including bandwidh, though their inversion method has several other tuning parameters), Hall et al. (2005) used near-optimal values derived by fitting a Gaussian model. With our method, we used the default bandwidth described in Section 3.2.

The first example is a normal model, for which the individual densities f_{jk} are the pdf's of $\mathcal{N}(\mu_{jk}, 1)$, with component means $\boldsymbol{\mu}_1 = (0, 0, 0)$ and $\boldsymbol{\mu}_2 = (3, 4, 5)$. The second example is the same as the first except that the individual densities are double exponential instead of normal. The third example is similar, but the individual densities are central or noncentral t densities on ten degrees of freedom: The first component has a central $t(10)$ distribution and thus $\boldsymbol{\mu}_1 = (0, 0, 0)$, whereas the second component's

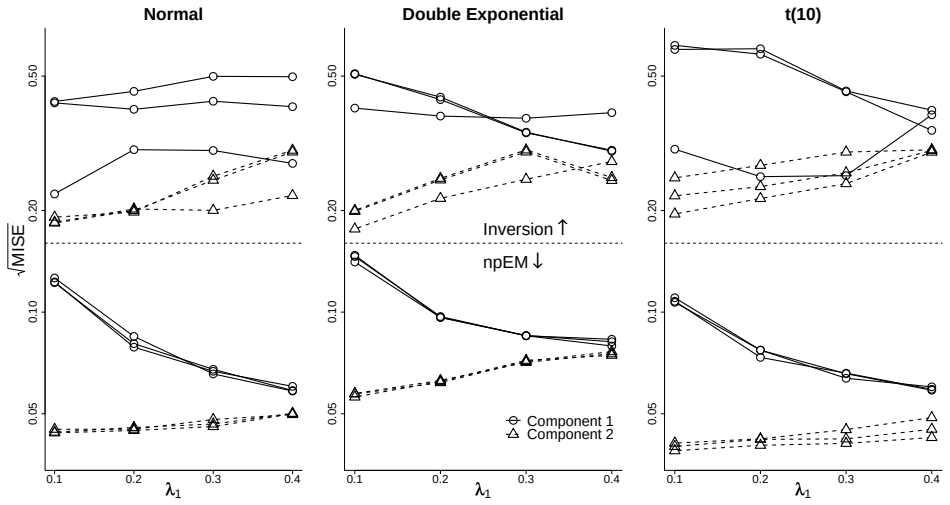

Figure 2: *Square roots of Mean Integrated Squared Errors (MISE) are shown on a logarithmic scale as a function of λ_1, the proportion of component 1, for three different simulated distributions for all f_{jk}, $j = 1, 2$ and $k = 1, 2, 3$. The results for the inversion algorithm of Hall et al. (2005) are approximated from their plots on page 675, Fig. 2, with a small bit of noise added to separate coincident points.*

coordinates are noncentral $t(10)$ distributions with noncentrality parameters 3, 4, and 5. Thus, the mean of the third component is $\boldsymbol{\mu}_2 = (3, 4, 5) \times 1.0837$. Recall that in all three examples — and indeed throughout this article — the coordinates are independently distributed conditional on their component membership.

Results given in Figure 2 show that our algorithm dramatically outperforms the inversion method for the three models. Note that the smallest value of MISE for the inversion method for any example is greater than the greatest value of MISE for our npEM algorithm; thus, the horizontal dotted line at $\sqrt{\text{MISE}} = 0.16$ in Figure 2 separates the two sets of results entirely. Because the three coordinates within each component and value of λ_1 are so similar relative to the scale of the plots, we do not distinguish among them in Figure 2. Predictably, we can see that the MISE is much smaller for the second component than the first when λ_1 is small (which means a larger proportion of the sample gives information about the second component), but the values appear to converge as λ_1 nears 1/2.

5.2 The Water-level data

The water-level dataset described in Section 1 is available in the **mixtools** package (Young et al., 2005) in R (R core development team, 2007) by typing `data(Waterdata)`. These data, with $n = 405$ and $r = 8$, have been analyzed by several authors using nonparametric mixture models based on converting the continuous angle measurements into binomial or multinomial data (Hettmansperger and Thomas, 2000; Cruz-Medina et al., 2004; Elmore et al., 2004). The last of these references, Elmore et al. (2004), gives the most sophisticated analysis of this dataset; we use it as a basis of comparison for our method.

By converting the water-level data into multinomial vectors, Elmore et al. (2004) are assuming that the eight coordinates of an observation vector are i.i.d. conditional on the mixture component from which the vector is drawn. Yet a more careful analysis, not possible using any published method we know of until now, reveals that there are subtle differences among the coordinate distributions. Grouping the coordinates into four blocks of two i.i.d. coordinates each uses knowledge of the task (described in Section 1) and appears more appropriate here.

Figure 1 of Section 1 summarizes our three-component solution, which may be obtained using `mixtools` via

```
blockid <- c(4,3,2,1,3,4,1,2) # blocks 1-4 refer to fig. 1
a <- npEMindrep(Waterdata, 3, blockid=blockid, h=4)
plot(a, hist=T, breaks=5*(0:37)-92.5)
```

(type `?npEMindrep` and `?plot.npEM` for more details on these functions). Note that "`h=4`" specifies the bandwidth, overriding the default value of Equation (9). Also note that because the default starting values are random, the commands above may not result in *exactly* the same solution.

For the three-component solution, our figure 1 clearly shows that one component, comprising almost 50% of the subjects, consists of individuals who know how to complete this task; these individuals' responses are highly peaked around the correct answer of zero degrees. The cutpoint method also finds a similarly shaped component and estimates its proportion at 0.440. However, the second and third components are qualitatively different than those found by the cutpoint method, particularly the smallest component. Using our method, we find that almost 8% of the subjects seem to draw the line parallel to the bottom of the vessel — yet the cutpoint approach misses this group because "parallel to the bottom" means one of -60, -30, 30, or 60 degrees depending on the orientation the vessel. In fact, the assumption that

all eight coordinates are identically distributed leads the cutpoint approach to conclude that the smallest component (with an estimated 17.7% of all subjects) is roughly uniformly distributed over the interval from −90 to 90. The more realistic model that is possible to estimate with our algorithm reveals details that the cutpoint approach simply cannot find easily.

In fact, the cutpoint approach *can* find the group of subjects who draw the water level parallel to the bottom of the vessel, but it needs a four-component model to do so. Elmore et al. (2004, Fig. 2) give a cutpoint solution for the four-component case, and we include the analagous four-component solution using our method here as Figure 3. (To obtain this result using `mixtools`, simply change the 3 to a 4 in the second line of the code given earlier in this section.)

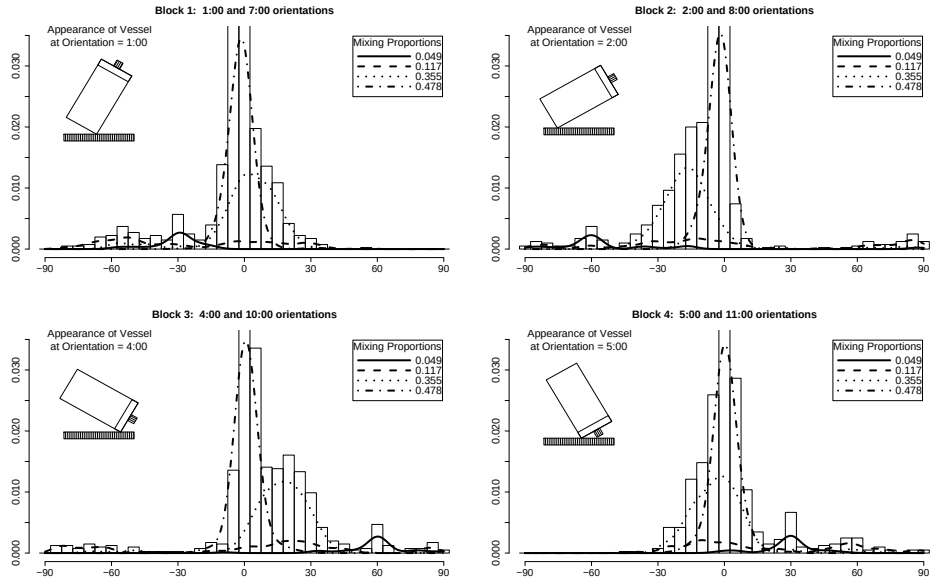

Figure 3: *Results of a four-component analysis of the water-level data using our algorithm.*

The cutpoint result finds one component with 3.3% of the subjects in which the density has four sharp peaks at −60, −30, 30, and 60 degrees. But this result masks the fact that those peaks occur in completely different coordinates, so the implicit assumption of conditionally i.i.d. coordinates using the cutpoint approach is probably not quite appropriate here.

As stated in Section 2, a word of caution is necessary here: it has never

been proven that the general model (3) is identifiable when $r = 8$ and $m = 3$ or 4. By contrast, under the more restrictive assumptions of the cutpoint method, we know that identifiability holds in these cases because $r \geq 2m-1$ (Elmore and Wang, 2003). The necessary (but not sufficient) lower bounds on r given by Hall et al. (2005) are $r \geq 4$, when $m = 3$, and $r \geq 5$, when $m = 4$; so with $r = 8$ there is at least the hope of identifiability. We are encouraged in the present example by several facts: First, the solutions we obtain, for both $m = 3$ and $m = 4$, are stable in the sense that we obtain the same solutions repeatedly for different randomly selected starting values for the algorithm. Furthermore, our results may be explained qualitatively via an understanding of how the data arose, and these results confirm and sharpen those found using a different method, the cutpoint method, in which identifiability has been proven to hold.

5.3 Stochastic vs. non-stochastic semiparametric EM

For Model (18) with $m = 2$ components, we compared the semiparametric stochastic EM (spSEM) algorithm of Bordes et al. (2007), which is discussed immediately following equation (8), with the deterministic semiparametric EM (spEM) algorithm that uses equation (19).

	λ	μ_1	μ_2	λ	μ_1	μ_2
True	0.25	-1	2	0.25	-1	2
	MSE			bias		
spSEM	0.0044	0.1880	0.0459	-0.0246	0.0413	-0.1003
spEM	0.0042	0.1154	0.0373	-0.0229	0.0056	-0.0898

Table 1: *Empirical mean squared error (MSE) and bias for* (λ, μ_1, μ_2)*, based on* $10,000$ *Monte-Carlo replications of Model (18) with* $f(\cdot)$ *taken to be standard normal and* $n = 100$*. The spSEM and spEM algorithms are run for 100 and 20 iterations each, respectively, starting from the true parameter values.*

The comparison is based on 10,000 Monte Carlo replications in which we selected the bandwidth h according to the formula used by Bordes et al. (2007), namely, $h = (4/3n)^{1/5}$, or $h = 0.422$ when $n = 100$. The spEM was allowed only 20 iterations, relative to the 100 iterations allowed the stochastic version, since its non-stochastic sequence of estimates requires fewer iterations to stabilize. Results are given in Table 1 and give empirical

evidence that the deterministic version is slightly more efficient than the stochastic version.

6 Discussion

The algorithm we propose in this article is the first algorithm we have seen in the literature for dealing with model (3) in its full generality. Furthermore, it is quite a bit easier to code than many if not all competing algorithms in the particular cases to which the latter are suited. Finally, we have given empirical evidence that our algorithm produces dramatically lower error rates than the inversion method of Hall et al. (2005) for the test cases used in that paper, and we have explained how our algorithm gives insight into the multivariate mixture structure of a particular dataset (the water-level dataset) that is not possible under the more restrictive assumption that each multivariate observation has conditionally i.i.d. coordinates.

As we point out in Section 2, the great flexibility of our method requires some caution, since it is very easy to apply the algorithm for arbitrary m (number of mixture components) and r (number of vector coordinates per observation) even when model (3) is not known to be identifiable. We know that model (3) is not identifiable for an arbitrary m and r; yet it is not yet known where the "identifiability boundary" might lie — i.e., for which values $\rho(m)$ it is true that $r \geq \rho(m)$ implies identifiability but $r \geq \rho(m) - 1$ does not. Hall and Zhou (2003) proved that $\rho(2) = 3$, and Hall et al. (2005) and Elmore et al. (2005) have made some progress towards a general solution, but so far such a solution remains elusive.

There are several questions about our algorithm that could be further investigated in addition to the identifiability question. For instance, Hall et al. (2005) introduce a further generalization of model (3). Namely, they allow some of the $f_{jb_k}(\cdot)$ to be *multivariate* densities whose coordinates are not independent. There is no difficulty in extending our algorithm to this case in principle, though to do so requires the use of multidimensional kernel density estimates. We have not explored this possibility yet.

Selection of an appropriate bandwidth is another area in which further work could shed some light. We have discussed this problem at length in Section 3.2. Indeed, selecting a bandwidth in a mixture setting like this one appears to be a fundamentally more complicated problem than the corresponding non-mixture case due to the fact that we do not have a sample from any of the individual mixture components *per se*, and we do not obtain information on the individual components until after the algorithm has

already been run. This suggests an iterative scheme as mentioned in Section 3.2, but we have not yet implemented such a scheme. Related to the bandwidth selection question is the question of whether our algorithm can be shown to be consistent for a fixed r and m; and if so, at what rate it converges. Preliminary empirical evidence suggests that this rate of convergence is slower than $n^{-1/2}$, but its comparison with the rate of convergence of standard kernel density estimation is not yet clear.

Finally, we reiterate that the analyses in this article may be reproduced using the publicly-available R package called `mixtools` (R Development Core Team, 2007; Young et al., 2007). Future revisions of this package may extend its capabilities to include some of the discussion items here.

References

Anderson, J. A. (1979), Multivariate logistic compounds, *Biometrika*, **66**: 17–26.

Bordes, L., Mottelet, S., and Vandekerkhove, P. (2006), Semiparametric estimation of a two-component mixture model, *Annals of Statistics*, **34**, 1204–1232.

Bordes, L., Chauveau, D., and Vandekerkhove, P. (2007), An EM algorithm for a semiparametric mixture model, *Computational Statistics and Data Analysis*, **51**: 5429–5443.

Cruz-Medina, I. R., Hettmansperger, T. P. and Thomas, H. (2004), Semiparametric mixture models and repeated measures: the multinomial cut point model, *Journal of the Royal Statistical Society, Series C*, **53**: 463–474.

Cruz-Medina, I. R. and Hettmansperger, T. P. (2004), Nonparametric estimation in semi-parametric univariate mixture models, *Journal of Statistical Computation and Simulation*, **74**: 513–524.

Dempster, A., Laird, N. and Rubin, D. (1977). Maximum likelihood from incomplete data via the EM algorithm (with discussion). *J. Roy. Stat. Soc.*, B, **39**, 1–38.

Elmore, R. T. and Wang, S. (2003), Identifiability and estimation in finite mixture models with multinomial coefficients, Penn State Department of Statistics Technical Report #03-04.

Elmore, R. T., Hettmansperger, T. P., and Thomas, H. (2004), Estimating component cumulative distribution functions in finite mixture models, *Communications in Statistics: Theory and Methods*, **33**: 2075–2086.

Elmore, R. T., Hall, P. and Neeman, A. (2005), An application of classical invariant theory to identifiability in nonparametric mixtures, *Annales de l'Institut Fourier*, **55**, 1: 1–28.

Elmore, R.T., Hettmansperger, T.P., and Xuan, F. (2006). A fully nonparametric test for one-way layouts, *Australian and New Zealand Journal of Statistics*, to appear.

Hall, P. and Zhou, X. H. (2003) Nonparametric estimation of component distributions in a multivariate mixture, *Annals of Statistics*, **31**: 201–224.

Hall, P., Neeman, A., Pakyari, R., and Elmore, R. (2005), Nonparametric inference in multivariate mixtures, *Biometrika*, **92**: 667–678.

Hettmansperger, T. P. and Thomas, H. (2000), Almost nonparametric inference for repeated measures in mixture models, *Journal of the Royal Statistical Society, Series B*, **62**, 811–825.

Hunter, D. R., Wang, S., and Hettmansperger, T. P. (2007), Inference for mixtures of symmetric distributions, *Annals of Statistics*, **35**: 224–251.

Leung, D. H.-Y. and Qin, J. (2006), Semi-parametric inference in a bivariate (multivariate) mixture model, *Statistica Sinica*, **16**: 153–163.

McLachlan, G. and Peel, D. A. (2000) *Finite Mixture Models*, New York: Wiley.

R Development Core Team (2007). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL `http://www.R-project.org`.

Silverman, B. W. (1986), *Density Estimation*, London: Chapman and Hall.

Thomas, H., Lohaus, A., and Brainerd, C.J. (1993). Modeling Growth and Individual Differences in Spatial Tasks, *Monographs of the Society for Research in Child Development*, **58**, 9: 1–190.

Titterington, D. M., Smith, A. F. M., and Makov, U. E. (1985). *Statistical Analysis of Finite Mixture Distributions*, Chichester, Great Britain: Wiley.

Qin, J. and Leung, D. H.-Y. (2006), Semiparametric analysis in conditionally independent multivariate mixture models, unpublished manuscript.

Young, D. S., Benaglia, T., Chauveau, D., Elmore, R. T., Hettmansperger, T. P., Hunter, D. R., Thomas, H., and Xuan, F. (2007) mixtools: Tools for mixture models, R package version 0.3.0.

Corresponding author
Didier Chauveau
Laboratoire MAPMO - UMR 6628 - Fédération Denis Poisson
Université d'Orléans
BP 6759, 45067 Orléans cedex 2, FRANCE.
Email: `didier.chauveau@univ-orleans.fr`