Nonlinear nonparametric mixed-effects models for unsupervised classification

Laura Azzimonti, Francesca Ieva & Anna Maria Paganoni

Computational Statistics

ISSN 0943-4062
Volume 28
Number 4

Comput Stat (2013) 28:1549-1570
DOI 10.1007/s00180-012-0366-5

Springer

Comput Stat (2013) 28:1549–1570
DOI 10.1007/s00180-012-0366-5

ORIGINAL PAPER

Nonlinear nonparametric mixed-effects models for unsupervised classification

Laura Azzimonti · Francesca Ieva · Anna Maria Paganoni

Received: 2 September 2011 / Accepted: 7 September 2012 / Published online: 27 September 2012

Abstract In this work we propose a novel EM method for the estimation of nonlinear nonparametric mixed-effects models, aimed at unsupervised classification. We perform simulation studies in order to evaluate the algorithm performance and we apply this new procedure to a real dataset.

Keywords Mixed-effects models · Nonparametric estimation · EM algorithm · Nonlinear models

1 Introduction

Nonlinear mixed-effects models (NLME models) are mixed-effects models in which at least one of the fixed or random effects appears nonlinearly in the model function. NLME models are increasingly used in several biomedical applications, especially in population pharmacokinetics, pharmacodynamic, immune cells reconstruction and epidemiological studies (see Sheiner and Beal 1980; Davidian and Gallant 1993; De Lalla et al. 2011; Ieva et al. 2012).
In these fields, statistical modeling based on NLME models takes advantage of tools that allow to distinguish overall population effects from drugs effects or unit specific influence.

L. Azzimonti · F. Ieva · A. Maria Paganoni (✉)
MOX, Dipartimento di Matematica, Politecnico di Milano, via Bonardi 9, 20133 Milan, Italy
e-mail: anna.paganoni@polimi.it

F. Ieva
e-mail: francesca.ieva@mail.polimi.it

L. Azzimonti
e-mail: laura.azzimonti@mail.polimi.it

Mixed-effects models include parameters associated with the entire population (fixed effects) and subject/group specific parameters (random effects). For this reason, mixed-effects models are able to describe the dynamics of the phenomenon under investigation, even in presence of high between subjects variability. When the random effects represent a deviation from the common dynamics of the population, mixed-effects models provide both estimates for the entire population's model and for each subject's one. In this work random effects have a different meaning, in fact they describe the common dynamics of different groups of subjects. In this framework, mixed-effects models provide only estimates for each group-specific model. Thanks to this property, it will be possible to consider mixed-effects models as an unsupervised clustering tool for longitudinal data and repeated measures. For this reason we focus our attention on the estimation of the distribution of the random effects $\mathcal{P}^*$.

A wide literature exists for parametric modeling of random effects distribution in linear and NLME models. In this framework, Maximum likelihood (ML) estimators are generally preferred because of their consistency and efficiency. However, due to the nonlinearity of the likelihood, we are not always able to provide explicitly the parameter estimators. A general and complete overview of linear multilevel models is given in Hox (1995). An analogous overview for nonlinear case is given in Gallant (1987). Fox (2002) shows how `R` and `S-plus` tools estimate linear and generalized linear mixed-effects models with parametric, in particular Gaussian, random effects. Concerning nonlinear models, in Goldstein (1991) a ML estimation of Gaussian random effects is provided for peculiar nonlinear forms. A stochastic approximation of traditional EM algorithm (SAEM) for estimating Gaussian random effects is suggested in Kuhn and Lavielle (2005), whereas an exact EM algorithm is described in Walker (1996). Finally, Wolfinger (1993) introduces a Laplace approximation for nonlinear random effects marginal distributions. However, parametric assumptions may sometimes result too restrictive to describe very heterogeneous or grouped populations. Moreover, when the number of measurements for unit is small, predictions for random effects are strongly influenced by the parametric assumptions. For these reasons nonparametric (NP) framework, which allow $\mathcal{P}^*$ to live in an infinite dimensional space, is attractive. It also provides in a very natural way a classification tool, as we will highlight later.

Methods for the estimation of linear NP random effects distribution in linear and generalized linear mixed-effects models have been proposed in Aitkin (1996; 1999), whereas Lai and Shih (2003), Davidian and Gallant (1993), Vermunt (2004), Antic et al. (2009), among others, deal with NP nonlinear models.

In this work we propose a novel estimation method for nonlinear NP mixed-effects models, aimed at unsupervised classification. Classification is performed through the estimation of the random effects distribution. The discreteness of the distribution, in fact, naturally clusters data in groups. The present algorithm is implemented in `R` program (version 2.14.0, R Development Core Team 2009) and the `R` source code is downloadable at the webpage http://mox.polimi.it/~azzimonti. To the best of our knowledge, this is the first example of free software for the estimation of nonlinear NP mixed-effects models.

In Sect. 2 the general framework of the work is sketched out and the algorithm for the estimation of nonlinear nonparametric random effects (NLNPEM) is described.

In Sect. 3 some simulation studies are presented, both for the linear and nonlinear case. We first compare the performance of our procedure with the already existing method in the linear framework, computing the Wasserstein distance between the true and the estimated distribution of random effects and the goodness of fit index $-2\log L$, then we test NLNPEM in the nonlinear case. Section 4 contains an application to real data. Concluding remarks and further developments of the present work are finally discussed in Sect. 5.

2 Methods

2.1 Model and framework

We consider the following NLME model for longitudinal data:

$$\begin{aligned} \mathbf{y}_i &= f(\boldsymbol{\beta}, \mathbf{b}_i, \mathbf{t}) + \boldsymbol{\epsilon}_i \quad i = 1, \ldots, N \\ \boldsymbol{\epsilon}_i &\sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbb{I}_n) \qquad \text{i.i.d.} \end{aligned} \tag{1}$$

where $\mathbf{y}_i \in \mathbb{R}^n$ is the response variable evaluated at times $\mathbf{t} \in \mathbb{R}^n$ and f is a general, real-valued and differentiable function with $p+q$ parameters. Each parameter of f is treated either as fixed or as random. Fixed effects are parameters associated with the entire population whereas random effects are subject-specific parameters that allow to identify clusters of subjects. $\boldsymbol{\beta} \in \mathbb{R}^p$ is a vector that contain all fixed effects and $\mathbf{b}_i \in \mathbb{R}^q$ is the vector for the ith subject random effects.

The function f is nonlinear at least in one component of the fixed or random effects. The errors ϵ_{ij} are associated with the jth measurement of the ith longitudinal data. They are normally distributed, independent between different subjects and independent within the same subject. In general, the proposed method could also take account of a different number of observations, located at different times, for different subjects. In (1) we chose not to consider this case in order to ease the notation, but the generalization is straightforward.

Usually random effects are assumed to be Normal distributed, $\mathbf{b}_i \sim \mathcal{N}_q(\mathbf{0}, \Sigma)$, with unknown parameters that, together with $\boldsymbol{\beta}$ and σ^2, can be estimated through methods based on the likelihood function (see Pinheiro and Bates 2000). In this parametric framework the ML estimators are generally preferred for their statistical properties, i.e., consistency and efficiency. Nevertheless the parametric assumptions could be too restrictive to describe highly heterogeneous or grouped data, so it might be necessary to move to a NP approach. In our case, we assume $\mathbf{b}_i$, for $i = 1, \ldots, N$, to be independent and identically distributed according to a probability measure $\mathcal{P}^*$ that belongs to the class of all probability measures on $\mathbb{R}^q$. $\mathcal{P}^*$ can then be interpreted as the mixing distribution that generates the density of the stochastic model in (1). We can face the problem of estimating $\mathcal{P}^*$ following the general theory of mixture likelihoods analysed form a geometrical point of view in Lindsay (1983a; 1983b). Looking for the ML estimator $\hat{\mathcal{P}}^*$ of $\mathcal{P}^*$ in the space of all probability measures on $\mathbb{R}^q$, the discreteness theorem proved in Lindsay (1983a), states that $\hat{\mathcal{P}}^*$ is a discrete measure with at most N support points. Moreover under suitable hypotheses on the distribution of the response

variable, satisfied for example by densities in the exponential family, the ML estimator is also unique as proved in Lindsay (1983b). Therefore the ML estimator of the random effects distribution can be expressed as a set of points $(\mathbf{c}_1, \ldots, \mathbf{c}_M)$, where $M \leq N$ and $\mathbf{c}_l \in \mathbb{R}^q$, and a set of weights $(\omega_1, \ldots, \omega_M)$, where $\omega_l \geq 0$ and $\sum_{l=1}^{M} \omega_l = 1$.

As mentioned above, in this paper we propose an algorithm for the joint estimation of $\boldsymbol{\beta}$, σ^2, $(\mathbf{c}_1, \ldots, \mathbf{c}_M)$ and $(\omega_1, \ldots, \omega_M)$ in the nonlinear framework of model (1). The estimation of the fixed effects $\boldsymbol{\beta}$, of the random effects $\mathbf{b}_i$ and of the variance σ^2 is performed through the maximization of the likelihood, mixture by the discrete distribution of the random effects

$$L(\boldsymbol{\beta}, \sigma^2 \Big| \mathbf{y}) = p(\mathbf{y}|\, \boldsymbol{\beta}, \sigma^2) = \sum_{l=1}^{M} \frac{\omega_l}{(2\pi\sigma^2)^{\frac{nN}{2}}} e^{-\frac{1}{2\sigma^2} \sum_{i=1}^{N} \sum_{j=1}^{n} (y_{ij} - f(\boldsymbol{\beta}, \mathbf{c}_l, t_j))^2} \quad (2)$$

with respect to the fixed effects $\boldsymbol{\beta}$, the error variance σ^2 and the random effects distribution $(\mathbf{c}_l, \omega_l)$, $l = 1, \ldots, M$. Each iteration of the algorithm described in Sect. 2.2 increases the likelihood in (2).

Concerning the distribution of random effects, for each $l = 1, \ldots, M$, $\mathbf{c}_l$ and ω_l represent the group specific parameters and the corresponding weights in the mixture (2), respectively. Notice that the number of support points M is computed by the algorithm as well and we do not have to fix it a priori. Since we don't have to specify a priori the number of support points and in consequence the number of groups, the NP mixed-effects model could be interpreted as an unsupervised clustering tool for longitudinal data. This tool could be very useful in order to identify the groups of subjects to be used in the analysis.

2.2 NLNPEM algorithm

The algorithm proposed for the estimation of the parameters of model (1) is an EM algorithm that arises from the framework described in Schumitzky (1991). The nonlinear nonparametric EM algorithm is an iterative algorithm that alternates an expectation step and a maximization step. During the expectation step we compute the conditional expectation of the likelihood function with respect to the random effects, given the observations and the parameters computed in the previous iteration, whereas during the maximization step we maximize the conditional expectation of the likelihood function. In this framework the random effects can be regarded as latent variables introduced in the model in order to take into account the overdispersion derived from the grouped structure of data. These variables are summed up in the likelihood during the E step as a mixing distribution.

At each iteration of the EM algorithm we obtain an update of the parameters that increases the likelihood (2), as proved in "Appendix A". The update is the following:

$$\omega_l^{(up)} = \frac{1}{N} \sum_{i=1}^{N} W_{il} \quad \text{for } l = 1, \ldots, M \quad (3)$$

$$\left(\boldsymbol{\beta}^{(up)}, \mathbf{c}_1^{(up)}, \ldots, \mathbf{c}_M^{(up)}, \sigma^{2(up)}\right) = \arg \max_{\boldsymbol{\beta}, \mathbf{c}_l, \sigma^2} \sum_{l=1}^{M} \sum_{i=1}^{N} W_{il} \ln p(\mathbf{y}_i | \boldsymbol{\beta}, \sigma^2, \mathbf{c}_l) \quad (4)$$

where

$$W_{il} = \frac{\omega_l p(\mathbf{y}_i | \boldsymbol{\beta}, \sigma^2, \mathbf{c}_l)}{\sum_{k=1}^{M} \omega_k p(\mathbf{y}_i | \boldsymbol{\beta}, \sigma^2, \mathbf{c}_k)} \quad (5)$$

and

$$p(\mathbf{y}_i | \boldsymbol{\beta}, \sigma^2, \mathbf{c}) = \frac{1}{(2\pi\sigma^2)^{n/2}} e^{-\frac{1}{2\sigma^2} \sum_{j=1}^{n} (y_{ij} - f(\boldsymbol{\beta}, \mathbf{c}, t_j))^2}.$$

The coefficients W_{il} represent the probability of $\mathbf{b}_i$ being equal to $\mathbf{c}_l$ conditionally to the observation $\mathbf{y}_i$ and given the fixed effects $\boldsymbol{\beta}$ and the variance σ^2, that is

$$W_{il} = p(\mathbf{b}_i = \mathbf{c}_l | \mathbf{y}_i, \boldsymbol{\beta}, \sigma^2)$$

in fact,

$$W_{il} = \frac{p(\mathbf{b}_i = \mathbf{c}_l) p(\mathbf{y}_i | \boldsymbol{\beta}, \sigma^2, \mathbf{c}_l)}{p(\mathbf{y}_i | \boldsymbol{\beta}, \sigma^2)} = \frac{p(\mathbf{y}_i, \mathbf{b}_i = \mathbf{c}_l | \boldsymbol{\beta}, \sigma^2)}{p(\mathbf{y}_i | \boldsymbol{\beta}, \sigma^2)} = p(\mathbf{b}_i = \mathbf{c}_l | \mathbf{y}_i, \boldsymbol{\beta}, \sigma^2).$$

Due to the high dimensionality of the parameter space in the maximization step, we compute the *arg–max* in (4) iteratively until convergence. In the first step of this iterative procedure we compute the *arg–max* in (4) with respect to the support points of the random effects distribution, setting $\boldsymbol{\beta}$ and σ^2 equal to the last computed values. Keeping $\boldsymbol{\beta}$ and σ^2 fixed enables us to maximize the expected loglikelihood with respect to all the support points $\mathbf{c}_l$ separately, that means

$$\mathbf{c}_l^{(up)} = \arg \max_{\mathbf{c}} \sum_{i=1}^{N} W_{il} \ln p(\mathbf{y}_i | \boldsymbol{\beta}, \sigma^2, \mathbf{c}) \quad l = 1, \ldots, M. \quad (6)$$

In the second step we fix the support points of the random effects distribution computed in the previous step and we compute the *arg–max* in Eq. (4) with respect to $\boldsymbol{\beta}$ and σ^2.

In order to compute the point estimate $\hat{\mathbf{b}}_i$ of $\mathbf{b}_i$ for each subject $i = 1, \ldots, N$, we maximize the conditional probability of $\mathbf{b}_i$ given the observations $\mathbf{y}_i$, the fixed effects $\boldsymbol{\beta}$ and the error variance σ^2. For this reason the estimation of the random effects, $\hat{\mathbf{b}}_i$, is obtained maximizing W_{il} over l, that is

$$\hat{\mathbf{b}}_i = \mathbf{c}_{\tilde{l}} \quad \text{if } \tilde{l} = \arg \max_{l} W_{il}.$$

The algorithm, given a starting discrete distribution with N support points for the random effects and a starting estimate for the fixed effects, updates the parameters through Eqs. (3) and (4) until convergence. The convergence of EM algorithms is

usually local but in this case we can obtain global convergence since the likelihood (2) has a unique maximum. Technical details together with the sketch of the algorithm are reported in "Appendix B".

During the iterations of the EM algorithm we can also reduce the support of the discrete distribution, in order to cluster the support of the random effects distribution. This support reduction consists in both making points very close to each other collapse and removing points with very low weight and not associated with any subject. In particular if two points are too close, that means $\|\mathbf{c}_l - \mathbf{c}_k\| < D$, where D is a tuning tolerance parameter, than we replace $\mathbf{c}_l$ and $\mathbf{c}_k$ with a new point $\mathbf{c}_{\min\{l,k\}} = (\mathbf{c}_l + \mathbf{c}_k)/2$ with weight $\omega_{\min\{l,k\}} = \omega_l + \omega_k$. Otherwise, if $\omega_l < \tilde{\omega}$, where $\tilde{\omega}$ is another tuning tolerance parameter, and the subset $\{i : \hat{\mathbf{b}}_i = \mathbf{c}_l\}$ is empty, we remove the point $\mathbf{c}_l$. The thresholds D and $\tilde{\omega}$ are two complexity parameters that affect the estimation of the NP distribution; the higher D is set, the lower is the number of groups. For this reason the two complexity parameters define a trade off between bias and high number of groups. In this work we prefer setting D small (i.e., much smaller than the standard deviation between groups) in order to obtain an higher number of groups and, in case, cluster them later. In general, the definition of the correct number of groups is an hard task, and it is strongly connected with thresholding. In the following, some rules of thumb will be provided for performing a suitable setting of parameters D and $\tilde{\omega}$.

3 Simulation studies

In order to validate the proposed estimation algorithm and to compare it with different procedures, we perform two simulation studies. Since we are mainly interested in classifying curves in an unsupervised framework, we focus our attention on the estimation of random effects distribution.

In the first simulation study (Sect. 3.2), we test our algorithm in a linear framework, in order to compare results of our procedure with those obtained with the algorithm introduced in Aitkin (1996) and implemented in the `npmlreg` `R`-package (see Einbeck et al. 2009). In the second one (Sect. 3.3), we consider two classic nonlinear functions f in (1): the exponential and the logistic growth curves. For each case we consider a test set of simulated curves (details are provided in "Appendix C") and we evaluate the algorithm performance in the estimation of the random effects computing the Wasserstein distance (defined in Sect. 3.1) between the true and the estimated distribution of the random effects.

3.1 Wasserstein distance

Evaluating the goodness of the estimation of a discrete distribution is not a straightforward task. Indeed, ways of comparing the true and the estimated probability distribution of the random effects have to take in account both support locations and weights, for this reason we adopt a multidimensional version of Wasserstein distance (see Gibbs and Su 2002). The Wasserstein distance between two probability measures $\mu,\ \nu$ on a subset Ω of the metric space $\mathbb{R}$ is defined as

$$d_W(\mu, \nu) = \int_{-\infty}^{\infty} |F(x) - G(x)| dx, \tag{7}$$

where F and G are the cumulative distribution functions of μ and ν respectively. The generalization to the q-dimensional case is straightforward. When the probability measures μ and ν are discrete, the computation of the integral in (7) is very easy, even in the q-dimensional case. It is known that the Wasserstein metric assumes values in $[0, |\Omega|]$, where $|\Omega|$ is the Euclidean measure of the support space Ω. For this reason, the Wasserstein distance divided by $|\Omega|$ is a good performance index for the evaluation of the estimates in the simulations study.

3.2 Linear cases

In this subsection, a simulation study for linear models is considered, therefore f in model (1) is linear. The general model, for $i = 1, \ldots, N$, include three different cases, that are:

$$\mathbf{y}_i = \begin{cases} \alpha + d_i \mathbf{t} + \boldsymbol{\epsilon}_i & \text{(random-slope case)} \\ a_i + \delta \mathbf{t} + \boldsymbol{\epsilon}_i & \text{(random-intercept case)} \\ a_i + d_i \mathbf{t} + \boldsymbol{\epsilon}_i & \text{(fully random case)} \end{cases}$$

where $\boldsymbol{\epsilon}_i$ are i.i.d. from $\mathcal{N}(\mathbf{0}, \sigma^2 \mathbb{I}_n)$ and $\mathbf{t}$ is the vector of sampling times. Intercept and slope are treated as fixed or random effects according to the different cases. In the fully random case, both slope and intercept parameters are considered random, i.e., $\mathbf{b}_i = (d_i, a_i)$, whereas in the random-slope and random-intercept case, $b_i = d_i$ and $b_i = a_i$ respectively. The interest is focused on random effects estimation, because our main goal is to test the performance of our algorithm in identifying the correct number of groups in simulated data and in estimating properly location and weights of different groups. Testing the linear case enables us to compare results of our algorithm with those carried out by the `R` algorithm `npmlreg`, which implements Aitkin (1996) procedure of NP random effects estimation. To be noticed is that our method is not efficient in the linear case, since it doesn't take advantage of the linearity of the problem. However it doesn't need any a priori specification of the number of distinct support points of the random effects distribution. Even if we don't specify the exact number of groups beforehand, the proposed method is able to estimate well the random effects distribution, as we will show in the following.

We simulated 8 datasets of linear growth curves grouped in a different number of balanced or unbalanced clusters (from 2 to 10 clusters). Parameters specification and details of each set of curves are reported in "Appendix C". All these datasets represent typical situations in which fitting a parametric mixed-effects model could be wrong because random effects are not normally distributed. On these datasets, we fitted models with both the NLNPEM method and the nonparametric ML approach introduced in Aitkin (1996).

The method introduced in Aitkin (1996) is a method for fitting overdispersed generalized linear models: the idea is to approximate the unknown and unspecified distribution of the random effects by a discrete mixture of densities from exponential family. This approximation leads to a simple expression of the marginal likelihood that can be maximized using a standard EM algorithm. Once specified the model and the number of random effects groups k, the `R` package `npmlreg` fits a linear mixed-effects model using nonparametric ML. Since we are testing the proposed method in a simulation setting, when `npmlreg` method is used we provide the correct number of groups, whereas, when NLNPEM is used, we don't have to. The N starting points for random effects distribution are randomly chosen in a proper range and the starting fixed effects are estimated through linear least squares. Finally, the tolerance D is set equal to 0.05 and $\tilde{\omega}$ equal to 0.05 in all cases. These two parameters are problem driven; $\tilde{\omega}$ for example is linked to the size of the smallest group that we want to detect, while D represents the minimum allowed distance between different points of the discrete random effects distribution. In particular, in the real case it is useful to perform a sensitivity analysis to set these two parameters, since the proposed method is an explorative tool for the detection of the real number of groups. A rule of thumb for setting these threshold parameters is the following: D may be much smaller than the standard deviation; on the other hand, $\tilde{\omega}$ may be set of the same order of the inverse of the total number of the observation in the dataset.

According to the dimension of the random effects ($q = 1$ for random-slope or random-intercept case, $q = 2$ when both effects are random), we properly define the model in `npmlreg` and NLNPEM algorithms. Notice that `npmlreg` does not allow to select one dimensional random effect in the case of random-slope only, but provides a random effects estimation for both intercept and slope parameters. In this case, in order to correctly compare the two methods, we have set also in the NLNPEM method both slope and intercept to be random in the random-slope case. Of course, in the NLNPEM method, random effects only for the slope may be selected by the user, if necessary.

Figure 1 shows the classification obtained applying the NLNPEM method to the lin2I, lin3S, lin9SI (first row), lin10I and lin10S (second row) datasets respectively. In each simulation a suitable version of the NLNPEM method is used in order to compare the results with the `npmlreg` method, i.e., random-intercept ($q = 1$) for lin2I and lin10I, random-slope and -intercept ($q = 2$) for lin3S, lin9SI and lin10S.

In Tables 4, 5 and 6 of "Appendix D", results of `npmlreg` and NLNPEM algorithms for three representative cases are compared, i.e., the estimations of random effects in terms of points and weights are reported and compared with the corresponding true distributions. Observing the estimated values reported in these tables , it can be argued that both methods estimate well both the discrete random components of the model and the fixed effects when a small number of groups is considered. Increasing the number of groups, the two algorithms show a different behavior. In particular we notice that, for large number of groups, `npmlreg` doesn't detect some points of the NP distribution, whereas NLNPEM performs better, even ignoring the true number of groups.

In terms of misclassification rate, the NLNPEM method performs better than the competitor. The mean misclassification rate (MMR) over all the simulations settings is

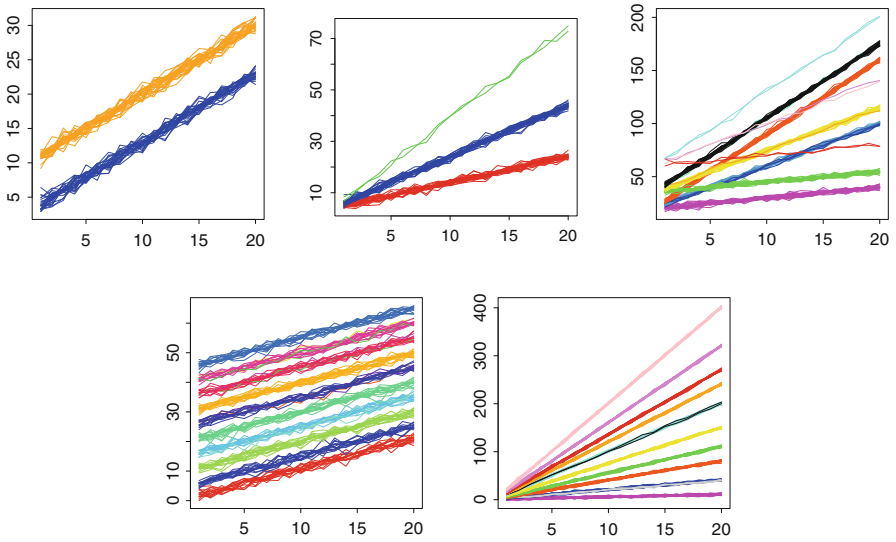

Fig. 1 NLNPEM classification of curves belonging to lin2I, lin3S, lin9SI (*first row*), lin10I and lin10S (*second row*) datasets, respectively

equal to 4.57 % using NLNPEM and to 5.84 % using `npmlreg`. Moreover, comparing the results obtained with the two methods in each simulation set, we notice that the bigger is the number of groups, the more significant is the difference between misclassification rates (for example, in lin10I model, NLNPEM misclassifies 14 % of the 150 curves vs. 20 % of `npmlreg`). It has also to be noticed that misclassification in NLNPEM method is caused by the detection of too many groups; in this case curves belonging to the same original group are assigned to different groups characterized by location points very close one to each other. This behavior can then be corrected properly tuning the tolerance parameters. The number of groups computed by the NLNPEM algorithm, in fact strongly depends on the tuning tolerances D and $\tilde{\omega}$, introduced in Sect. 2.2. Since the NLNPEM method is an explorative tool, different parameters D and $\tilde{\omega}$ can be used to investigate the presence of groups. Even if the number of points is greater than the real number, the points tend to cluster near the true ones. Moreover, summing the weights of the points in each cluster, we obtain results similar to the exact weights. Anyway, within the context of explorative analyses carried out in an unsupervised framework, the NLNPEM method provides information on the number of the groups that effective is. The clues concerning the number of groups provided by NLNPEM algorithm are effective and useful when clustering is the main goal of the analysis. The proposed method is also capable of detecting outlier groups, whereas the `npmlreg` method is able to detect them only in presence of small number of groups. In general, we notice that sometimes `npmlreg` method performs poorly in estimation or even doesn't reach the convergence, whereas NLNPEM does. This is clear, for example, observing what happens when there are 10 groups for intercept, in "lin10I" dataset (see Table 6).

In order to resume the goodness of fit of NLNPEM and the `npmlreg` methods, we finally compare the normalized Wasserstein distances between the true discrete random

Table 1 Normalized Wasserstein distances and $-2\log L$ index for `npmlreg` and NLNPEM algorithm respectively in the simulated linear cases

Model	Wasserstein distance		$-2\log L$	
	`npmlreg`	NLNPEM	`npmlreg`	NLNPEM
lin2S	0.01357	0.01357	2861.2	954.0
lin2I	0.00454	0.01001	2097.7	190.5
lin4SI	0.00812	0.00812	5974.4	2021.4
lin3S	0.00304	0.00451	2839.8	900.2
lin3I	0.00345	0.00345	2938.3	1017.2
lin9SI	0.01776	0.00403	16127.0	5358.6
lin10S	0.03363	0.00065	76716.1	18093.0
lin10I	0.02305	0.00155	12795.8	2949.8

effects distribution and the estimated one through the two methods, for each simulated set of linear curves. Results are reported in Table 1, together with the goodness of fit index $-2\log L$.
To be noticed is that, in the case of Wasserstein distance, results are similar for all datasets where both algorithms perform well. On the other hand, significant differences exist in cases with a large number of groups, where NLNPEM performance is much better both in terms of Wasserstein distance and $-2\log L$.

3.3 Nonlinear cases

In this subsection we describe two nonlinear case studies: the exponential and the logistic growth model. These two models are among the most used in nonlinear mixed-effects framework because they find application in several areas like pharmacokinetics and epidemiological studies.
Since other nonlinear NP methods are not available for free software, we are not able to compare the NLNPEM results with those obtained with other methods; for this reason we will only test NLNPEM performance, providing the normalized Wasserstein distance between the true distribution and the estimated one.

3.3.1 Exponential growth model

We first describe the exponential case, in which we consider the following nonlinear function in model (1):

$$f(t) = \alpha\left(1 - e^{-\lambda t}\right)$$

which is nonlinear in λ. The two parameters α and λ represent respectively the asymptote and the growth rate.

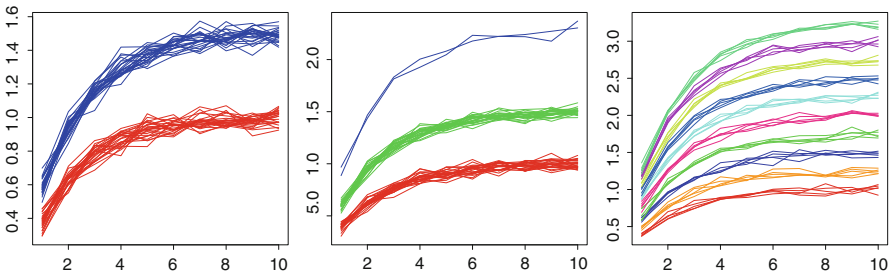

Fig. 2 NLNPEM classification in exp2A, exp3A and exp10A datasets, respectively

Table 2 Normalized Wasserstein distances for NLNPEM algorithm in the simulated exponential cases

Model	Wasserstein distance
exp2A	0.00752
exp3A	0.00264
exp10A	0.00348

In this case study we consider only random effects for the asymptote, that means that the mixed-effects model becomes

$$\mathbf{y}_i = a_i \left(1 - e^{-\lambda \mathbf{t}}\right) + \boldsymbol{\epsilon}_i$$

where $\boldsymbol{\epsilon}_i \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbb{I}_n)$ are i.i.d. errors, a_i are the random effects for the asymptote ($b_i = a_i$) and λ is the fixed effect for the growth rate ($\beta = \lambda$).

We simulated 3 datasets of exponential growth curves, described in Appendix C in which only asymptote varies. The starting random effects distribution has N support points, randomly chosen in a proper range, and the starting fixed effects are estimated through nonlinear least squares. The tuning tolerance parameter D is set equal to 0.01 and $\tilde{\omega}$ equal to 0.05. Figure 2 shows original datasets, where each curve is colored according to the group estimated by NLNPEM method.

The number of groups is larger than the real one in all the three cases; however the estimated random effects create the right number of clusters located close to the correct points. In the exp3A case the NLNPEM method is also able to identify the *outlier* group estimating well the locations and the weights of the random effects. Only few curves are misclassified, in fact the MMR is equal to 0.67 %. Table 2 shows the normalized Wasserstein distance for each case.

3.3.2 Logistic growth model

The second nonlinear model tested is the logistic growth model. In this case, the nonlinear function is:

$$f(t) = \frac{\alpha}{1 + e^{-\frac{t-\delta}{\gamma}}}$$

where α represent the asymptote, δ is the inflection point, which corresponds to the time at which the growth curve reaches the half of the asymptote, and γ is the time elapsed between δ and the time at which the growth curve reaches 3/4 of the asymptote level. The parameter γ will always be treated as a fixed effect while the asymptote and the inflection point will be treated either as fixed or as random effect according to different cases. The general model, which is nonlinear in δ and γ, includes then three different cases:

$$\mathbf{y}_i = \begin{cases} \dfrac{a_i}{1+e^{-\frac{\mathbf{t}-\delta}{\gamma}}} + \boldsymbol{\epsilon}_i & \text{(random-asymptote case)} \\ \dfrac{\alpha}{1+e^{-\frac{\mathbf{t}-d_i}{\gamma}}} + \boldsymbol{\epsilon}_i & \text{(random-inflection case)} \\ \dfrac{a_i}{1+e^{-\frac{\mathbf{t}-d_i}{\gamma}}} + \boldsymbol{\epsilon}_i & \text{(random-asymptote and inflection case)} \end{cases} \tag{8}$$

where $\boldsymbol{\epsilon}_i \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbb{I}_n)$ are i.i.d. errors, a_i and d_i represent the random effects for the asymptote and the inflection point, while α, δ and γ represent the fixed effects. In particular in the random-asymptote case $b_i = a_i$ and $\beta = (\delta, \gamma)$, in the random-inflection case $b_i = d_i$ and $\beta = (\alpha, \gamma)$ and in the random-asymptote and -inflection case $\mathbf{b}_i = (a_i, d_i)$ and $\beta = \gamma$.

We simulated 8 datasets of logistic growth curves that include all the cases resumed in (8). Each dataset is composed by a different number of balanced or unbalanced groups (from 2 to 10 clusters) similar to those presented in the linear framework. Details are provided in "Appendix C".

Since the NLNPEM method is able to fit all three models resumed in (8), we fit the right model for each dataset. The starting random effects distribution has N support points, randomly chosen in a proper range, and the starting fixed effects are estimated through nonlinear least squares. We set the tolerance D equal to 0.05 and $\tilde{\omega}$ equal to 0.05.

Figure 3 shows some of the simulated datasets, where each curve is colored according to the group estimated by NLNPEM method. Even if we don't specify a priori the correct number of groups, we are able to cluster correctly the curves, in cases characterized by both few and many groups, as proved by the MMR for the 8 simulated logistic datasets, which is equal to 2.16 %. The method is also able to capture correctly outliers; in all the unbalanced cases the proposed method recognizes the outliers and estimates well both the locations and the weights of random effects.

In order to test the NLNPEM method we can compare these results with those obtained considering always a model with both asymptote and inflection point as random effects. For the two random-asymptote and -inflection cases we have obviously fitted only the model with two random effects. The normalized Wasserstein distances are shown in Table 3; the first column represents the normalized Wasserstein distance for a random-inflection model ($q = 1$), the second one for a random-asymptote model ($q = 1$), and the third one represents the same distance for models with two random effects ($q = 2$).

We first notice that the normalized Wasserstein distances are always very low, that means that the NLNPEM method is able to estimate well both random and fixed effects even in presence of a high number of groups. We also notice that in the NLNPEM

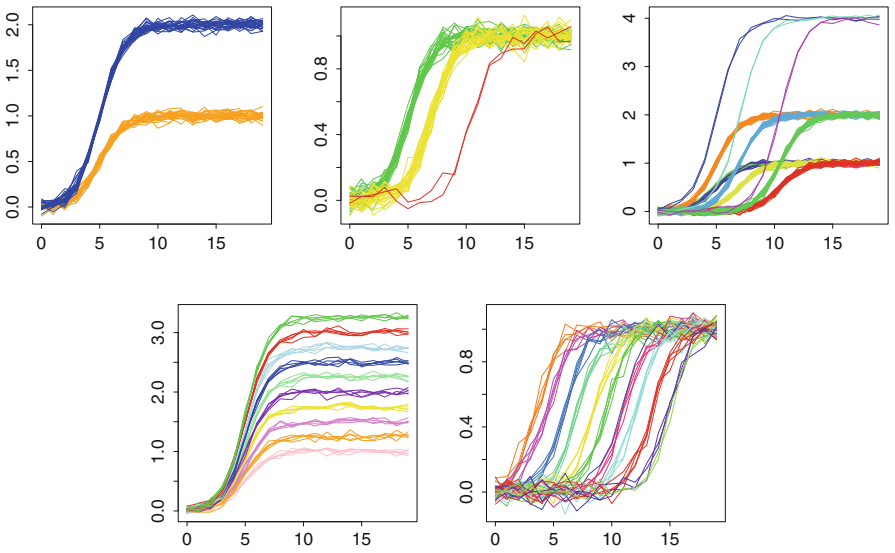

Fig. 3 NLNPEM classification of curves belonging to logis2A, logis3I, logis9AI (*first row*), logis10A and logis10I (*second row*) datasets, respectively

Table 3 Normalized Wasserstein distances for NLNPEM algorithm in the simulated logistic cases

Model	Wasserstein distance		
	Random Inflection ($q = 1$)	Random Asymptote ($q = 1$)	Fully Random ($q = 2$)
logis2A	–	0.00045	0.00730
logis2I	0.01065	–	0.00114
logis4AI	–	–	0.00422
logis3A	–	0.00063	0.00094
logis3I	0.00967	–	0.00171
logis9AI	–	–	0.00324
logis10A	–	0.00151	0.00027
logis10I	0.00521	–	0.00017

method we are allowed to consider more parameters as random effects than needed, without damaging the parameter estimation. In particular this approach could be useful when we don't know which are the parameters to be considered random. For this purpose we could perform a first analysis considering all parameters as random effects and then fit a second model fixing the parameters that show a very low variability. This approach could be performed with the NLNPEM method because it can handle both random and fixed effects whereas other previous methods cannot.

4 Application to NON STEMI data

In this section we study a dataset concerning Hospital Discharges of patients affected by acute myocardial infarction (AMI) without ST-segment elevation (NON-STEMI).

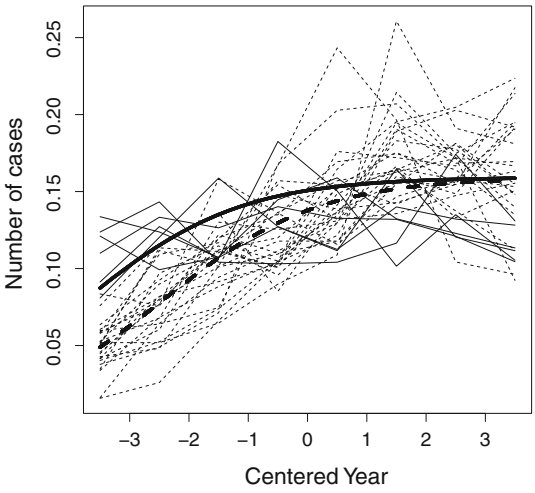

Fig. 4 Standardized number of AMI without ST-segment elevation diagnoses in the period 2000–2007 in the 30 largest clinical institutions of Regione Lombardia. The year has been centered and normalization has been carried out standardizing the yearly number of diagnoses for each hospital by total number of diagnoses in the time window 2000–2007. Real data are depicted according to the NLNPEM clusters and NLNPEM fitted models are superimposed

These data have already been studied in Ieva et al. (2012). Figure 4 represents the normalized number of NON-STEMI diagnoses along the time period 2000–2007 grouped by hospital and relative to the 30 largest clinical institutions of Regione Lombardia. For each hospital the yearly number of diagnoses has been standardized by the hospital total number of diagnoses in the time period 2000–2007.

As pointed out in Ieva et al. (2012), the random-inflection case in model (8) seems to capture the common "S-shaped" growing pattern. The NLNPEM algorithm clusters the hospitals in $M = 2$ different groups, according to the estimated discrete distribution of the random effect for the inflection point (see Fig. 4). The estimated fixed effects are $\hat{\alpha} = 0.16$ and $\hat{\gamma} = 1.45$, the estimated discrete measure $\hat{\mathcal{P}}^*$ is concentrated on $(\hat{\mathbf{c}}_1, \hat{\mathbf{c}}_2) = (-3.61, -2.47)$ with weights $(\hat{\omega}_1, \hat{\omega}_2) = (0.16, 0.84)$ and the estimated variance is $\hat{\sigma}^2 = 8.1 \times 10^{-4}$. This analysis, performed with $D = 0.1$ and $\tilde{\omega} = 0.05$, backs up the presence of two groups of hospitals according to different inflection points and automatically detects an unsupervised cluster structure. Even if clinical best practice maintains that there is no evidence for a greater incidence of NON-STEMI in this period it is known that since the early 2000s a new diagnostic procedure—the *troponin* exam—has been introduced and this could have produced an increased number of positive diagnoses, by easing NON-STEMI detection. Hence, the presence of 2 clusters could be a consequence of the different hospital timings in the introduction and adoption of this practice. This hypothesis cannot be validated directly since the timings of adoption of the troponin exam by the 30 different hospitals included in the analysis are not available.

The good agreement with previous results detailed in Ieva et al. (2012) together with the great advantage of a NP approach advocates the real profit in using this new estimation algorithm.

5 Conclusions

In this work, we present a new estimation method for nonlinear NP mixed-effects models, aimed at unsupervised classification; this method, named NLNPEM, is based on an EM algorithm and can be considered a flexible tool for investigating the presence of groups in data.
We first tested this procedure in a linear framework against the already existing tool for NP random effects estimation (the `npmlreg` `R` package), in order to compare the performance of the new method in terms of random effects distribution estimation. Results show that it performs well both in terms of Wasserstein distance and $-2\log L$ index, even ignoring the real number of groups, and that it always reaches convergence, even in those cases where several groups are present. Then we tested NLNPEM algorithm also in simulated test set within nonlinear frameworks of exponential and logistic growth. In both these cases, the number of groups and distribution of random effects are correctly and effectively identified. An application to real data of NON-STEMI is also presented in the end, where the potential of our method in unsupervised clustering analysis is highlighted.
NLNPEM may be successfully adopted for investigating the grouped nature of data also when the random effects distribution is not discrete. For example, we carried out some simulations in the case of random effects distribution arising from different mixtures of Gaussian distributions, centered in the same location points of the exponential and logistic cases treated in the paper. Also in this setting, NLNPEM proved to be able to detect true locations of the mixture and to reconstruct the true distribution of the random effects.

Acknowledgments The case study in Sect. 4 is within the Strategic Program "Exploitation, integration and study of current and future health databases in Lombardia for Acute Myocardial Infarction" supported by "Ministero del Lavoro, della Salute e delle Politiche Sociali" and by "Direzione Generale Sanità - Regione Lombardia".

Appendix A: Proof of increasing likelihood property of the EM algorithm

In Sect. 2.2 we propose an EM algorithm for the estimation of the parameters of model (1). The update of the parameter described by Eqs. (3) and (4) provides an increasing of the likelihood function (2), that is

$$L\left(\boldsymbol{\beta}^{(up)}, \sigma^{2(up)} \middle| \mathbf{y}\right) \geq L\left(\boldsymbol{\beta}, \sigma^2 \middle| \mathbf{y}\right)$$

where $\beta^{(up)}$ and $\sigma^{2(up)}$ are the updated fixed effects and error variance. The likelihood $L(\boldsymbol{\beta}^{(up)}, \sigma^{2(up)}|\mathbf{y})$ is computed summing up the random effects with respect to the updated discrete distribution $(\mathbf{c}_l^{(up)}, \omega_l^{(up)})$ for $l = 1, \ldots, M$.
Thanks to the definition of likelihood function (2) we have that

$$\log\left[\frac{L\left(\boldsymbol{\beta}^{(up)}, \sigma^{2(up)} \middle| \mathbf{y}\right)}{L\left(\boldsymbol{\beta}, \sigma^2 \middle| \mathbf{y}\right)}\right] = \sum_{i=1}^{N} \log\left[\frac{p(\mathbf{y}_i | \boldsymbol{\beta}^{(up)}, \sigma^{2(up)})}{p(\mathbf{y}_i | \boldsymbol{\beta}, \sigma^2)}\right]$$

All these terms can be bounded above by the quantity

$$\log\left[\frac{p(\mathbf{y}_i|\,\boldsymbol{\beta}^{(up)},\sigma^{2(up)})}{p(\mathbf{y}_i|\,\boldsymbol{\beta},\sigma^2)}\right] \geq Q_i(\theta^{(up)},\theta) - Q_i(\theta,\theta) \tag{9}$$

where

$$Q_i(\theta^{(up)},\theta) = \sum_{l=1}^{M}\left(\frac{\omega_l\, p(\mathbf{y}_i|\,\boldsymbol{\beta},\sigma^2,\mathbf{c}_l)}{p(\mathbf{y}_i|\,\boldsymbol{\beta},\sigma^2)}\right)\log\left(\omega_l^{(up)}\, p(\mathbf{y}_i|\,\boldsymbol{\beta}^{(up)},\sigma^{2(up)},\mathbf{c}_l^{(up)})\right).$$

$Q_i(\theta,\theta)$ is analogously defined, and $\theta = (\boldsymbol{\beta},\mathbf{c}_1,\ldots,\mathbf{c}_M,\omega_1,\ldots,\omega_M,\sigma^2)$. This bound can be found thanks to the convexity of the logarithm since

$$\begin{aligned}
\log\left[\frac{p(\mathbf{y}_i|\,\boldsymbol{\beta}^{(up)},\sigma^{2(up)})}{p(\mathbf{y}_i|\,\boldsymbol{\beta},\sigma^2)}\right] &= \log\sum_{l=1}^{M}\frac{\omega_l^{(up)}\, p(\mathbf{y}_i|\,\boldsymbol{\beta}^{(up)},\sigma^{2(up)},\mathbf{c}_l^{(up)})}{p(\mathbf{y}_i|\,\boldsymbol{\beta},\sigma^2)}\\
&= \log\sum_{l=1}^{M}\left(\frac{\omega_l\, p(\mathbf{y}_i|\,\boldsymbol{\beta},\sigma^2,\mathbf{c}_l)}{p(\mathbf{y}_i|\,\boldsymbol{\beta},\sigma^2)}\right)\left(\frac{\omega_l^{(up)}\, p(\mathbf{y}_i|\,\boldsymbol{\beta}^{(up)},\sigma^{2(up)},\mathbf{c}_l^{(up)})}{\omega_l\, p(\mathbf{y}_i|\,\boldsymbol{\beta},\sigma^2,\mathbf{c}_l)}\right)\\
&\geq \sum_{l=1}^{M}\left(\frac{\omega_l\, p(\mathbf{y}_i|\,\boldsymbol{\beta},\sigma^2,\mathbf{c}_l)}{p(\mathbf{y}_i|\,\boldsymbol{\beta},\sigma^2)}\right)\log\left(\frac{\omega_l^{(up)}\, p(\mathbf{y}_i|\,\boldsymbol{\beta}^{(up)},\sigma^{2(up)},\mathbf{c}_l^{(up)})}{\omega_l\, p(\mathbf{y}_i|\,\boldsymbol{\beta},\sigma^2,\mathbf{c}_l)}\right)\\
&= Q_i(\theta^{(up)},\theta) - Q_i(\theta,\theta)
\end{aligned}$$

Defining

$$Q(\theta^{(up)},\theta) = \sum_{i=1}^{N} Q_i(\theta^{(up)},\theta) \quad \text{and} \quad Q(\theta,\theta) = \sum_{i=1}^{N} Q_i(\theta,\theta)$$

we obtain, thanks to Eq. (9), an upper bound for the quantity of interest

$$\log\left[\frac{L\left(\boldsymbol{\beta}^{(up)},\sigma^{2(up)}\middle|\,\mathbf{y}\right)}{L\left(\boldsymbol{\beta},\sigma^2\middle|\,\mathbf{y}\right)}\right] \geq Q(\theta^{(up)},\theta) - Q(\theta,\theta)$$

We have now to show that $\forall\theta$

$$Q(\theta^{(up)},\theta) \geq Q(\theta,\theta)$$

In order to prove this result we can show that, $\forall\theta$ fixed, $\theta^{(up)}$ is defined as

$$Q(\theta^{(up)},\theta) = \arg\max_{\tilde{\theta}} Q(\tilde{\theta},\theta)$$

Defining W_{il} as in Eq. (5) we obtain

$$\begin{aligned} Q(\tilde{\theta}, \theta) &= \sum_{i=1}^{N}\sum_{l=1}^{M}\left(\frac{\omega_l\, p(\mathbf{y}_i|\,\boldsymbol{\beta}, \sigma^2, \mathbf{c}_l)}{p(\mathbf{y}_i|\,\boldsymbol{\beta}, \sigma^2)}\right)\log\left(\tilde{\omega}_l\, p(\mathbf{y}_i|\,\tilde{\boldsymbol{\beta}}, \tilde{\sigma}^2, \tilde{\mathbf{c}}_l)\right) \\ &= \sum_{i=1}^{N}\sum_{l=1}^{M} W_{il}\log\left(\tilde{\omega}_l\, p(\mathbf{y}_i|\,\tilde{\boldsymbol{\beta}}, \tilde{\sigma}^2, \tilde{\mathbf{c}}_l)\right) \\ &= \sum_{i=1}^{N}\sum_{l=1}^{M} W_{il}\log\tilde{\omega}_l + \sum_{i=1}^{N}\sum_{l=1}^{M} W_{il}\log p(\mathbf{y}_i|\,\tilde{\boldsymbol{\beta}}, \tilde{\sigma}^2, \tilde{\mathbf{c}}_l) \\ &= J_1(\tilde{\omega}_1, \ldots, \tilde{\omega}_M) + J_2(\tilde{\boldsymbol{\beta}}, \tilde{\mathbf{c}}_1, \ldots, \tilde{\mathbf{c}}_M, \tilde{\sigma}^2) \end{aligned}$$

The functionals J_1 and J_2 can be maximized separately. The update (3) for the weights of the random effects distribution $\omega_1, \ldots, \omega_M$ is obtained in closed form maximizing the functional J_1.
The functional J_1 can be written as

$$J_1(\tilde{\omega}_1, \ldots, \tilde{\omega}_M) = \sum_{l=1}^{M-1}\sum_{i=1}^{N} W_{il}\ \log\tilde{\omega}_l + \sum_{i=1}^{N} W_{iM}\ \log\left(1 - \sum_{l=1}^{M-1}\tilde{\omega}_l\right)$$

Imposing the gradient of the functional J_1 equal to zero we obtain

$$\frac{\partial J_1}{\partial \tilde{\omega}_l} = \frac{\sum_{i=1}^{N} W_{il}}{\tilde{\omega}_l} - \frac{\sum_{i=1}^{N} W_{iM}}{\tilde{\omega}_M} = 0 \qquad \forall l = 1, \ldots, M-1$$

that is equivalent to

$$\frac{\sum_{i=1}^{N} W_{il}}{\tilde{\omega}_l} = \frac{\sum_{i=1}^{N} W_{ik}}{\tilde{\omega}_k} \qquad \forall l, k = 1, \ldots, M$$

Since $\sum_{l=1}^{M} W_{il} = 1$, we obtain $\omega_l^{(up)} = \sum_{i=1}^{N} W_{il}/N$.
On the other hand the update (4) for the fixed effects $\boldsymbol{\beta}$, the error variance σ^2 and the support points of the random effects distribution $\mathbf{c}_1, \ldots, \mathbf{c}_M$ is obtained maximizing the functional J_2 in an iterative way, described in Sect. 2.2 and in "Appendix B".

Appendix B: Details on NLNPEM Algorithm

The NLNPEM is the following:

1. Define a starting discrete distribution for random effects with support on $M = N$ points $(\mathbf{c}_l^{(0)}, \omega_l^{(0)})$ for $l = 1, \ldots, M$, a starting estimate for the fixed effects $\boldsymbol{\beta}^{(0)}$ and for $\sigma^{2(0)}$ and the tolerance parameters D and $\tilde{\omega}$;

2. given $(\mathbf{c}_l^{(k-1)}, \omega_l^{(k-1)})$ for $l = 1, \ldots, M$, $\boldsymbol{\beta}^{(k-1)}$ and $\sigma^{2(k-1)}$, update the weights $\omega_1^{(k)}, \ldots, \omega_M^{(k)}$ of the random effect distribution, according to Eq. (3);
3. (a) initialize $\mathbf{c}_l^{(k,0)} = \mathbf{c}_l^{(k-1)}$ for $l = 1, \ldots, M$, $\boldsymbol{\beta}^{(k,0)} = \boldsymbol{\beta}^{(k-1)}$ and $\sigma^{2(k,0)} = \sigma^{2(k-1)}$;
   (b) given $(\mathbf{c}_l^{(k,j-1)}, \omega_l^{(k-1)})$ for $l = 1, \ldots, M$, $\boldsymbol{\beta}^{(k,j-1)}$ and $\sigma^{2(k,j-1)}$ update the M support points $\mathbf{c}_1^{(k,j)}, \ldots, \mathbf{c}_M^{(k,j)}$ according to Eq. (6);
   (c) given $(\mathbf{c}_l^{(k,j)}, \omega_l^{(k-1)})$ for $l = 1, \ldots, M$, $\boldsymbol{\beta}^{(k,j-1)}$ and $\sigma^{2(k,j-1)}$ maximize Eq. (4) with respect to $\boldsymbol{\beta}$ and σ^2 obtaining $\boldsymbol{\beta}^{(k,j)}$ and $\sigma^{2(k,j)}$;
   (d) iterate steps 3(b) and 3(c) until convergence and set $\mathbf{c}_l^{(k)} = \mathbf{c}_l^{(k,j)}$ for $l = 1, \ldots, M$, $\boldsymbol{\beta}^{(k)} = \boldsymbol{\beta}^{(k,j)}$ and $\sigma^{2(k)} = \sigma^{2(k,j)}$;
4. iterate steps 2 and 3 until convergence;
5. reduce the support of the discrete distribution, according with the tuning parameters D and $\tilde{\omega}$;
6. iterate steps 2, 3 and 5 until convergence.

The algorithm reaches convergence when parameters and discrete distribution stop changing or when there is no variation in the log-likelihood function.

Appendix C: Details on simulation study

Appendix C.1: The linear case

We simulated 8 datasets of linear curves grouped in a number of clusters that vary form 2 to 10. Different values of the error variance σ^2 have been chosen for each test set, in order to obtain noisy observations for each curve. Some examples of simulated data are shown in left panels of Fig. 1. Datasets addressed with the name "S" contain groups in which only slopes is random, "I" datasets contain groups where only intercept is random and "SI" datasets contain curves where both slope and intercept are random. The simulated datasets are then:

- lin2S: 2 balanced groups, each one composed by 25 curves, with the same intercept (equal to 4), 2 different slopes ($\mathbf{c} = (c_1, c_2) = (1, 2)$) and $\sigma = 1$;
- lin2I: 2 balanced groups, each one composed by 25 curves with the same slope (equal to 1), 2 different intercept ($\mathbf{c} = (c_1, c_2) = (3, 10)$) and $\sigma = 0.65$;
- lin4SI: 4 balanced groups, each one composed by 25 curves, where location points $\mathbf{c} = (\mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3, \mathbf{c}_4)$ are obtained from all possible combinations of 2 different slopes (equal to 1 and 3) and 2 different intercepts (equal to 40 and 60), i.e., $\mathbf{c}_1 = (1, 40)$, $\mathbf{c}_2 = (1, 60)$, $\mathbf{c}_3 = (3, 40)$ and $\mathbf{c}_4 = (3, 60)$ with $\sigma = 1$;
- lin3S: 3 unbalanced groups, composed by 24, 24 and 2 curves respectively, with the same intercept (equal to 4), 3 different slopes ($\mathbf{c} = (c_1, c_2, c_3) = (1, 2, 3.5)$) and $\sigma = 1$;
- lin3I: 3 unbalanced groups, composed by 24, 24 and 2 curves respectively, with the same slope (equal to 1), 3 different intercepts ($\mathbf{c} = (c_1, c_2, c_3) = (2, 7, 14)$) and $\sigma = 1$;

- lin9SI: 9 unbalanced groups, 6 of whom containing 24 curves and 3 containing 2 curves, where location points $\mathbf{c} = (c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8, c_9)$ are obtained from all possible combinations of 3 different slopes (equal to 1, 4 and 7) and 3 different intercept (equal to 20, 35 and 60) with $\sigma = 1.5$;
- lin10S: 10 balanced groups, each one composed by 50 curves with the same intercept (equal to 1), 10 different slopes ($\mathbf{c} = (c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8, c_9, c_{10}) = (0.5, 2, 4, 5.5, 7.5, 10, 12, 13.5, 16, 20)$) and $\sigma = 1.5$;
- lin10I: 10 balanced groups, each one composed by 15 curves with the same slope (equal to 1), 10 different intercepts ($\mathbf{c} = (c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8, c_9, c_{10}) = (1, 5, 10, 15, 20, 25, 30, 35, 40, 45)$) and $\sigma = 1$.

Appendix C.2: The exponential case

We simulated 3 datasets of exponential growth curves where only asymptote varies and is considered as random. All datasets are then addressed with the name "A". They are:

- exp2A: 2 balanced groups, each one composed by 25 curves, with the same growth rate ($\lambda = 0.5$), 2 different asymptotes ($\mathbf{c} = (c_1, c_2) = (1, 1.5)$) and $\sigma = 0.04$;
- exp3A: 3 unbalanced groups of 24, 24 and 2 curves respectively, with the same growth rate ($\lambda = 0.5$), 3 different asymptotes ($\mathbf{c} = (c_1, c_2, c_3) = (1, 1.5, 2.3)$) and $\sigma = 0.04$;
- exp10A: 10 balanced groups, each one composed by 5 curves, with the same growth rate ($\lambda = 0.5$), 10 different asymptotes ($\mathbf{c} = (c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8, c_9, c_{10}) = (1, 1.25, 1.5, 1.75, 2, 2.25, 2.5, 2.75, 3, 3.25)$) and $\sigma = 0.04$.

Appendix C.3: The logistic case

We simulated 8 datasets of logistic growth curves. Datasets addressed with the name "A" represent random asymptote cases, "I" datasets contain groups where only inflection point is random and "AI" ones contain curves where both asymptote and inflection point are random. We then have:

- logis2A: 2 balanced groups, each one composed by 25 curves, with $\delta = 6$, $\gamma = 1$, 2 different asymptotes ($\mathbf{c} = (c_1, c_2) = (1, 2)$) and $\sigma = 0.04$;
- logis2I: 2 balanced groups, each one composed by 25 curves, with $\alpha = 1$, $\gamma = 1$, 2 different inflection points ($\mathbf{c} = (c_1, c_2) = (6, 8)$) and $\sigma = 0.04$;
- logis4AI: 4 balanced groups, each one composed by 25 curves, where location points $\mathbf{c} = (\mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3, \mathbf{c}_4)$ are obtained from all possible combinations of 2 different asymptotes (equal to 1 and 2) and 2 different inflection points (equal to 6 and 10), i.e., $\mathbf{c}_1 = (1, 6)$, $\mathbf{c}_1 = (1, 10)$, $\mathbf{c}_1 = (2, 6)$ and $\mathbf{c}_4 = (2, 10)$ with $\gamma = 1$ and $\sigma = 0.04$;
- logis3A: 3 unbalanced groups of 24, 24 and 2 curves respectively, with $\delta = 6$, $\gamma = 1$, 3 different asymptotes ($\mathbf{c} = (c_1, c_2, c_3) = (1, 2, 3.5)$) and $\sigma = 0.04$;
- logis3I: 3 unbalanced groups of 24, 24 and 2 curves respectively, with $\alpha = 1$, $\gamma = 1$, 3 different inflection points ($\mathbf{c} = (c_1, c_2, c_3) = (6, 8, 11.5)$) and $\sigma = 0.04$;

- logis9AI: 9 unbalanced groups of curves (6 of whom containing 24 curves and 3 containing 2 curves), where location points $\mathbf{c} = (c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8, c_9)$ are obtained from all possible combinations of 3 different asymptotes (equal to 1, 2 and 4) and 3 different inflection points (equal to 6, 8 and 11.5) with $\gamma = 1$ and $\sigma = 0.04$;
- logis10A: 10 balanced groups, each one composed by 5 curves, with $\delta = 6$, $\gamma = 1$, 10 different asymptotes ($\mathbf{c} = (c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8, c_9, c_{10}) = (1, 1.25, 1.5, 1.75, 2, 2.25, 2.5, 2.75, 3, 3.25)$) and $\sigma = 0.04$;
- logis10I: 10 balanced groups, each one composed by 5 curves, with $\alpha = 1$, $\gamma = 1$, 10 different inflection points ($\mathbf{c} = (c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8, c_9, c_{10}) = (4.5, 5.5, 7, 8, 9.5, 10.5, 12, 13, 14.5, 16)$) and $\sigma = 0.04$.

Appendix D: Comparison of results

Comparison of estimates carried out by `npmlreg` and NLNPEM method are reported here, for some cases of interest mentioned in the paper (Tables 4, 5, 6).

- Linear case—Random-intercept case (lin2I)
- Linear case—Random-slope case (lin3S)
- Linear case—Random-intercept case (lin10I)

Table 4 Estimates carried out by `npmlreg` and NLNPEM method on lin2I dataset, where intercept is considered as random, with 2 balanced groups

Effects		True	`npmlreg`	NLNPEM
Fixed	Slope	1	1.0021	1.0021
Random	Intercept 1	3	2.9382	2.9382
	(weight 1)	(0.5)	(0.5)	0.5
	Intercept 2	10	10.0150	10.0149
	(weight 2)	(0.5)	(0.5)	0.5

Table 5 Estimates carried out by `unpmlreg` and NLNPEM method on lin3S dataset, where slope is considered as random, with 3 unbalanced groups

Effects		True	`npmlreg`	NLNPEM
Random	Slope 1	1	1.0107	1.0029
	(weight 1)	(0.48)	(0.48)	(0.48)
	Slope 2	2	1.9982	2.0105 1.9530
	(weight 2)	(0.48)	(0.48)	(0.45) (0.03)
	Slope 3	3.5	3.5250	3.5251
	(weight 3)	(0.04)	(0.04)	(0.04)
Random	Intercept	4	3.9326	4.0259
	Intercept	4	4.0751	3.8974 4.8376
	Intercept	4	3.3717	3.7174

Table 6 Estimates carried out by `npmlreg` and NLNPEM method on lin10I dataset, where intercept is considered as random, with 10 balanced groups

Effects		True	`npmlreg`	NLNPEM
Fixed	Slope	1	1.0019	1.0019
Random	Intercept 1	1	0.9114 0.9114	0.9114
	(weight 1)	(0.1)	(0.00050) (0.09949)	(0.1)
	Intercept 2	5	5.0257	5.0258
	(weight 2)	(0.1)	(0.1)	(0.1)
	Intercept 3	10	–	10.0409
	(weight 3)	(0.1)	–	(0.1)
	Intercept 4	15	12.5442	14.7748 15.0869
	(weight 4)	(0.1)	(0.2)	(0.013) (0.087)
	Intercept 5	20	19.9818	19.9818
	(weight 5)	(0.1)	(0.1)	(0.1)
	Intercept 6	25	27.4750	24.9252 25.1609
	(weight 6)	(0.1)	(0.2)	(0.038) (0.062)
	Intercept 7	30	–	29.8789
	(weight 7)	(0.1)	–	(0.1)
	Intercept 8	35	35.0050	34.9155 35.1641
	(weight 8)	(0.1)	(0.1)	(0.064) (0.036)
	Intercept 9	40	39.9516	39.7483 39.9701 40.4362
	(weight 9)	(0.1)	(0.1)	(0.030) (0.060) (0.010)
	Intercept 10	45	45.0017 45.0017	45.0018
	(weight 10)	(0.1)	(0.09949) (0.000507)	(0.1)

References

Aitkin M (1996a) A general maximum likelihood analysis of overdispersion in generalized linear models. Stat Comput 6:251–262

Aitkin M (1999b) A general maximum likelihood analysis of variance components in generalized linear models. Biometrics 55:117–128

Antic J, Laffont CM, Chafaï D, Concordet D (2009) Comparison of nonparametric methods in nonlinear mixed effect models. Comput Stat Data Anal 53(3):642–656

Davidian M, Gallant AR (1993) The nonlinear mixed effects model with a smooth random effects density. Biometrika 80(3):475–488

De Lalla C, Rinaldi A, Montagna D, Azzimonti L, Bernardo ME, Sangalli LM, Paganoni AM, Maccario R, Zecca M, Locatelli F, Dellabona P, Casorati G (2011) Invariant natural killer T-cell reconstitution in pediatric leukemia patients given HLA-haploidentical stem cell transplantation defines distinct CD4+ and CD4- subset dynamics and correlates with remission state. J Immunol 186(7):4490–4499

Einbeck J, Darnell R, Hinde J (2009) npmlreg: Nonparametric maximum likelihood estimation for random effect models. [Online] http://CRAN.R-project.org/package=npmlreg

Fox J (2002) Linear mixed models, appendix to an R and S-PLUS companion to applied regression

Gallant AR (1987) Nonlinear statistical models. Wiley, New York

Gibbs AL, Su FE (2002) On choosing and bounding probability metrics. Int Stat Rev 70(3):419–435

Goldstein H (1991) Nonlinear multilevel models, with an application to discrete response data. Biometrika 78(1):45–51

Hox JJ (1995) Applied multilevel analysis. TT-Publikaties, Amsterdam

Ieva F, Paganoni AM, Secchi P (2012) Mining administrative health databases for epidemiological purposes: a case study on acute myocardial infarctions diagnoses. In: Pesarin F, Torelli S (eds) Accepted for publication in advances in theoretical and applied statistics. Springer, Berlin. [Online] http://mox.polimi.it/it/progetti/pubblicazioni/quaderni/45-2010.pdf

Kuhn E, Lavielle M (2005) Maximum likelihood estimation in nonlinear mixed effect models. Comput Stat Data Anal 49(4):1020–1038

Lai TL, Shih MC (2003) Nonparametric estimation in nonlinear mixed-effects models. Biometrika 90(1):1–13

Lindsay BG (1983a) The geometry of mixture likelihoods: a general theory. Ann Stat 11(1):86–94

Lindsay BG (1983b) The geometry of mixture likelihoods, part II: the exponential family. Ann Stat 11(3):783–792

Pinheiro JC, Bates DM (2000) Mixed-effects models in S and S-plus. Springer, New York

R Development Core Team (2009) R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna. [Online] http://www.R-project.org

Sheiner LB, Beal SL (1980) Evaluation of methods for estimating population pharmacokinetic parameters. III. Monoexponential model: routine clinical pharmacokinetic data. J Pharmacokinet Pharmacodyn 11(3):303–319

Schumitzky A (1991) Nonparametric EM algorithms for estimating prior distributions. Appl Math Comput 45(2):143–157

Vermunt JK (2004) An EM algorithm for the estimation of parametric and non-parametric hierarchical nonlinear models. Statistica Neerlandica 58(2):220–233

Walker S (1996) An EM algorithm for nonlinear random effects models. Biometrics 52(3):934–944

Wolfinger R (1993) Laplace's approximation for nonlinear mixed models. Biometrika 80(4):791–795