A Bayesian approach for clustering and exact finite-sample model selection in longitudinal data mixtures

M. Corneli[1,2], E. Erosheva[3], X. Qian[3], and M. Lorenzi[4]

[1]*Université Côte d'Azur, Laboratoire CEPAM, Nice, France*
[2]*Université Côte d'Azur, Inria, CNRS, Laboratoire J.A. Dieudonné, MAASAI team, Nice, France*
[3]*Department of Statistics, School of Social Work, and the Center for Statistics and the Social Sciences, University of Washington, Seattle, WA, 98195, USA*
[4]*Université Côte d'Azur, Inria, CNRS, EPIONNE team, Nice, France*

Abstract

We consider mixtures of longitudinal trajectories, where one trajectory contains measurements over time of the variable of interest for one individual and each individual belongs to one cluster. The number of clusters as well as individual cluster memberships are unknown and must be inferred. We propose an original Bayesian clustering framework that allows us to obtain an exact finite-sample model selection criterion. Our approach is more flexible and parsimonious than asymptotic alternatives such as Bayesian Information Criterion (BIC) or Integrated Classification Likelihood (ICL) criterion in the choice of the number of clusters. Moreover, our approach has other desirable qualities: i) it keeps the computational effort of the clustering algorithm under control and ii) it generalizes to several families of regression mixture models, from linear to purely non-parametric.

1 Introduction and background

The question of clustering longitudinal trajectories arises in a number of social sciences concerned with describing human behavior over time (see Erosheva et al., 2014, and references therein). Here, we focus on longitudinal data involving N individuals. Each individual is associated with one (and only one) hidden group.

Latent random variable z_i labels the group (*cluster*) of the i-th individual. Assuming that there is a variable of interest that is measured for all individuals, a vector $y_i \in \mathbb{R}^D$ keeps track of the measurements for the i-th individual, where D is the number of measurements. In the following, it is assumed that exactly D measurements are taken at the same times across N individuals and y_{ij} denotes the j-th measurement for the i-th individual at time t_j, with $j \leq D$. Assume the following generative model

$$y_{ij} = \phi(t_{ij})^T \beta_{z_i} + \sigma \epsilon_{ij}, \quad \forall i \leq N, j \leq D, \tag{1}$$

where $\phi(\cdot): \mathbb{R} \to \mathbb{R}^K$ is a user defined function (e.g., identity, polynomial, etc.), $\beta_1, \ldots, \beta_Q$ are parameters in $\mathbb{R}^K$ with Q the number of mixing components, and σ is the scalar standard deviation of the noise term. The residuals ϵ_{ij} are all i.i.d Gaussian $\mathcal{N}(0,1)$ distributed. The following more compact notation will also be employed

$$y_i = \Phi \beta_{z_i} + \sigma \epsilon_i, \tag{2}$$

where $\Phi \in \mathbb{R}^{D \times K}$ is the *design matrix*, whose j-th row is $\phi(t_{ij})^T$, y_i and ϵ_i are vectors in $\mathbb{R}^D$ and ϵ_i follows a multivariate isotropic Gaussian distribution $\mathcal{N}(0, \sigma^2 I_D)$, with I_D denoting the identity matrix of order D. In the following, $y_1, \ldots, y_N$ are assumed independent, given $z_1, \ldots, z_N$. Finally, the model can be extended straightforward by assuming that the standard deviation σ also depends on z_i. All the results reported in this paper are still valid.

1.1 A frequentist approach and some related drawbacks

For a given Q, assuming that $z_1, \ldots, z_N$ are i.i.d. random variables and such that

$$\mathbb{P}\{z_i = q\} = \pi_q, \qquad \forall q \in \{1, \ldots, Q\},$$

the observed data likelihood of the generative model described in the above section is

$$p(y_1, \ldots, y_N | \theta) = \sum_{i=1}^{N} \sum_{q=1}^{Q} \pi_q g(y_i; \Phi\beta_q, \sigma^2 I_D), \tag{3}$$

where $\theta := \{\beta_q, \pi_q, \sigma^2\}_{q \leq Q}$ is the set of the model parameters and $g(\cdot; \mu, \Sigma)$ denotes the probability density function of a multivariate Gaussian distribution with mean μ and covariance matrix Σ. Clearly, this generative model can be seen as a *constrained* Gaussian mixture model (see Appendix A for more details and a proof that time matters in this model).

A standard approach to estimate the model parameters (see, for instance, Nagin et al., 2005) would:

1. set $\phi(\cdot)$ to some polynomial of reasonable order (possibly one!), depending on the the data,

2. maximize the log-likelihood $\sum_{i=1}^{N} \log p(y_i|\theta, Q)$ with respect to θ to obtain $\hat{\theta}^{ML}$. Posterior distributions $p(z_i|y_i, \hat{\theta}^{ML}, Q)$ could then be computed for all i and they might used to cluster the observations.

Repeating steps 1 and 2 for different Q, the number of clustering components Q might be estimated via some model selection criterion (e.g. AIC or BIC) (see e.g. Nagin et al., 2005; Muthén and Asparouhov, 2008).

However, in practice, AIC and BIC are often not able to provide reasonable choice for choosing the number of longitudinal trajectory clusters. It is common that these goodness-of-fit criteria keep improving as the number of groups Q increases until models encounter convergence problems (Erosheva et al., 2014). In addition, when the choice of Q is perceived as large because the identified clusters do not represent substantively distinct trajectory patterns, the practical advice is to reduce Q to a lower number that is more meaningful in the applied context (e.g., Nagin et al., 2005).

We now describe some know drawbacks to using AIC/BIC for selecting the number of clusters in standard mixture data analysis. We conjecture that these drawbacks could provide a possible explanation as to why AIB/BIC often fail to produce meaningful number of clusters with longitudinal trajectories.

First, we note that BIC (as well as AIC) is an *asymptotic* criterion needing a large number of observations to be given and some relevant assumptions to be fulfilled by the data that may not be straightforward in case of longitudinal observations. In addition, BIC was shown to possibly overestimate of the number of clusters in mixture models (see for instance Biernacki et al., 2000; Baudry et al., 2010). This point is illustrated with an example in Figure 1. One thousand points were independently sampled from a mixture of $Q = 4$ Gaussian distributions in dimension two. The actual number of mixing components can be safely recovered by BIC model selection by some software in standard statistical libraries (e.g. mclust on R). However, Q is *not* necessarily the most meaningful number of clusters. Indeed, in Figure 1 there are 2 disjoint sets of points. In our simulations in Section 4, we show that the over-estimation problem illustrated in Figure 1 might exist also in longitudinal data mixtures.

Second, a more parsimonious alternative to AIC/BIC for model selection in mixture models is the Integrated Classification Likelihood (ICL, Biernacki et al., 2000). See Appendix B for a formal definition of ICL and the proof that ICL is a lower bound of BIC. Although ICL might select a smaller number of components than BIC, it still remains an asymptotic criterion. Moreover, whereas ICL was shown to solve the model selection issue described in Figure 1 (see for instance

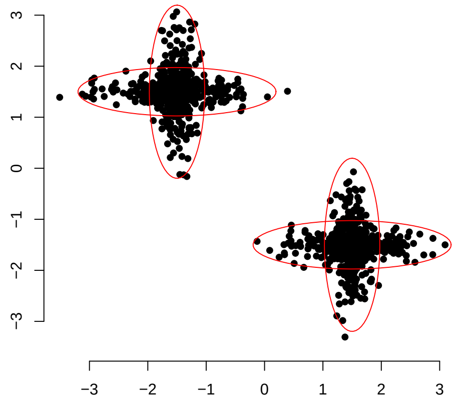


Figure 1: One thousand data points independently sampled from a mixture of $Q = 4$ bi-variate Gaussian distributions (identified by the red ellipses), forming two separated groups.

Baudry et al., 2010), it does not solve the overestimation problem for longitudinal data mixtures, as shown in Section 4.

In an attempt to address these issues, this paper focuses on a Bayesian clustering framework for the generative model detailed so far allowing us to compute an *exact* (i.e. a finite sample) version of the ICL by adopting prior distributions on the model parameters and integrating them out. A similar approach was adopted for Gaussian mixture models by Bertoletti et al. (2015) or by Côme and Latouche (2015); Corneli et al. (2016), in the context of graph data clustering.

We should note that non-parametric Bayesian approaches such as a Bayesian extension of the generative model in Eq. 3 to a Dirichlet process (DP) mixture model (see e.g. MacEachern and Müller, 1998) would represent an alternative way to analyze longitudinal mixtures. Bayesian non-parametric techniques as well as their extensions to grouped data sharing cluster memberships (Hierarchical DP, Teh et al., 2005) have been widely used in the context of mixture models and topic models (Griffiths and Steyvers, 2004). Apart from their flexibility, one of the main advantages of the DP mixture models is their ability to perform clustering and selection of the number of components *simultaneously* in one shot. In addition, DP mixture models can be used with a reduced computational cost thanks to variational inference (Blei et al., 2006). However, we do not pursue Bayesian non-parametric methods here because one of the main aims of this paper is to detail a model selection method that could be used on a given clustering *separately* from the clustering process. In this way, practitioners will have a choice of using their preferred clustering tool and just adopt our model selection criterion or using our complete approach of clustering and model selection in longitudinal data. This objective is difficult (and even unnatural) to be reached via DPMMs.

We organize the remainder of our paper as follows. Section 2 develops our Bayesian framework to clustering longitudinal trajectories. Section 3 develops the associated estimation routine whose computational cost remains reasonable. Section 4 presents a simulation study comparing our method to some state-of-the-art alternatives, in order to highlight the main features of our approach. A final discussion, in Section 6, concludes the paper.

2 A Bayesian perspective

In this section, we detail a Bayesian extension of the model in Eq. (2) allowing us to i) cluster the observations $y_1, \dots, y_N$ in Q groups and ii) select Q in a non asymptotic framework. The target probability distribution, that we would like to maximize with respect to the pair (Z, Q) is

$$p(Z, Q|Y) = \int p(Z, Q|Y, \theta)p(\theta)d\theta \tag{4}$$

where $Z := (z_1, \dots, z_N)$, $Y := (y_1, \dots, y_N)$ and the model parameters θ are seen as random variables and integrated out. Note that, from a full Bayesian perspective, the number of clusters Q is also viewed as a random variable in the above equation. In order to develop an estimation algorithm that is reasonably fast, we choose not to implement MCMC algorithms to simulate the above posterior distribution and present the following alternative strategy that relies on the Bayes rule

$$p(Z, Q|Y) = \frac{p(Y, Z|Q)p(Q)}{p(Y)}.$$

where $p(Q)$ is a prior distribution over Q. Since the denominator does not depend on (Z, Q), it holds that

$$\begin{aligned} \arg\max_{(Z,Q)} p(Z, Q|Y) &= \arg\max_{(Z,Q)} \Big(p(Y, Z|Q)p(Q)\ \Big) \\ &= \arg\max_{Q} \left(\arg\max_{Z|Q} \Big(p(Y, Z|Q)\ \Big)p(Q)\right). \end{aligned}$$

Note that, because $\max_{Z|Q} p(Y, Z|Q)$ does not depend on the functional form of $p(Q)$, the results shown in next sections, only involving $\max_{Z|Q} p(Y, Z|Q)$, remain valid for any choice of $p(Q)$. However, to simplify the exposition, we assume that Q is uniform distributed ($p(Q) \propto 1$) and the above equation then reduces to

$$\arg\max_{(Z,Q)} p(Z,Q|Y) = \arg\max_{(Z,Q)} p(Y,Z|Q). \tag{5}$$

Thus, computing the maximum posterior (MAP) estimates of Z and Q reduces to maximize the complete data integrated log-likelihood.

2.1 A closed form complete data integrated log-likelihood

The complete data integrated log-likelihood on the right hand side of Eq. (5) is

$$p(Y,Z|Q) = \int p(Y,Z|\theta,Q)p(\theta|Q)d\theta, \tag{6}$$

where we recall that $\theta := \{\beta_q, \pi_q, \sigma^2\}_q$.

We stress that, in general, this quantity is *not* tractable, since the integral on the right hand side of the above equality cannot be computed. However, some further assumptions allow us to obtain $p(Y,Z|Q)$ in a closed form, which is essential in order to develop the clustering/model selection strategy detailed in the next section. First, we assume that the prior distribution factorizes over the model parameters, namely

$$p(\beta,\sigma,\pi) = p(\beta,\sigma)p(\pi),$$

where, in order to keep the notation uncluttered, we denote $\beta := \{\beta_q\}_q$ and $\pi := \{\pi_q\}_q$. Thus, the integrated log-likelihood in Eq. (6) factorizes

$$p(Y,Z|Q) = p(Y|Z,Q)p(Z|Q). \tag{7}$$

Then, we assume prior conjugate distributions for β, σ and π. Specifically, conditionally on σ^2, assume that $\beta_1,\dots,\beta_Q$ follow independent Gaussian prior distributions

$$\beta_q \sim \mathcal{N}\left(0, \sigma^2\eta_q I_K\right), \tag{8}$$

where $\eta_1,\dots,\eta_Q$ are positive parameters and I_K is the identity matrix of order K. The β_qs are further assumed to be independent from ϵ_i, for all i. Likewise, assume σ^2 follows an Inverse Gamma prior distribution

$$\sigma^2 \sim \mathrm{I}\Gamma(a,b), \tag{9}$$

where $a,b > 0$. Finally, assume π follows a Dirichlet prior distribution

$$\pi \sim \mathrm{Dir}(\alpha_1,\dots,\alpha_Q), \tag{10}$$

where $\alpha_q > 0$ for all q.

The integration with respect to β is detailed in the next section, since it allows us to highlight a central feature of our Bayesian model. The details of the integration with respect to σ^2 and π are postponed in Appendices C and D.

2.1.1 Integrating with respect to β

By integrating out β_q in Eq. (2), we obtain the marginal conditional density of y_i

$$y_i|z_i, \sigma^2 \sim \mathcal{N}\left(0, \sigma^2\left(\eta_{z_i}\Phi\Phi^T + I_D\right)\right). \tag{11}$$

By the law of iterated expectations it follows that

$$\begin{aligned} \operatorname{Cov}(y_i, y_j|z_i, z_j, \sigma^2) &= \mathbb{E}\left[y_i y_j^T|z_i, z_j, \sigma^2\right] \\ &= \mathbb{E}\left[\Phi\beta_{z_i}\beta_{z_j}^T\Phi^T|z_i, z_j, \sigma^2\right] \\ &= \sigma^2\eta_{z_i}\Phi\Phi^T\mathbf{1}_{\{z_i=z_j\}}, \end{aligned}$$

where $\mathbf{1}_A(\cdot)$ denotes the indicator function over a set A. The above equation has an important consequence: after β_q is integrated out, the random vectors y_i sharing the same cluster are no longer independent, as they were in the frequentist model. Moreover, let us denote by

$$y^{(q)} := \{y_i|i \leq N, z_i = q\}, \tag{12}$$

the set of trajectories forming the q-th cluster, whose cardinality is denoted by C_q. Since all vectors in $y^{(q)}$ are Gaussian distributed, their joint conditional density can be specified as

$$Y_q|Z, \sigma^2 \sim \mathcal{N}\left(0, \sigma^2 G_q\right), \tag{13}$$

where $Y_q \in \mathbb{R}^{DC_q}$ is a column vector obtained by concatenating all the observations in cluster $y^{(q)}$ and $G_q \in \mathbb{R}^{DC_q \times DC_q}$ is a block matrix. The blocks on the main diagonal are of the form $\left(\eta_q\Phi\Phi^T + I_D\right)$, whereas the blocks outside the main diagonal look like $\left(\eta_q\Phi\Phi^T\right)$.

Remark. *Equation* (11) *shows that the generative model detailed so far can entirely be expressed in terms of* kernels. *In other terms, we can directly work with $\Phi\Phi^T$ without previous specifying the feature map $\Phi(\cdot)$. By doing that, the generative approach described so far embodies*

1. *mixtures of **linear** regression models, with $[\Phi\Phi^T]_{jl} := t_j t_l$,*
2. *mixtures of **polynomial** regression models, with $[\Phi\Phi^T]_{jl} := (1 + t_j t_l)^r$ being a polynomial kernel of degree r,*
3. *mixtures of **non-parametric** regression models, with $[\Phi\Phi^T]_{jl} := \exp\left(-\frac{(t_j - t_l)^2}{2\gamma}\right)$ being a radial basis function (RBF) kernel, for some positive scale parameter γ,*
4. *any other mixture of regression models depending on the choice of a symmetric, positive definite kernel matrix $\Phi\Phi^T$.*

3 Inference

The closed form expression of the integrated log-likelihood is reported in Eqs (21)-(22) in Appendices C-D. This quantity depends on the value of the hyper-parameters $\iota := \{\eta, a, b, \alpha\}$, that must be set based on the prior knowledge about the the data. If the number of groups Q is assumed to vary in a range $\{1, \ldots, Q_{\max}\}$, for a given Q, we aim at estimating

$$Z_Q^* := \arg\max_{Z|Q} \log p(Y, Z|Q).$$

Then, the Q leading to the highest value of $\log p(Y, Z_Q^*|Q)$ will be the estimated number of clusters (see Section 2)[1].

The estimation of Z_Q^* is challenging. The strategy that we propose relies on a *greedy* maximization of $\log p(Y, Z|Q)$. In simple terms, we compute changes in the integrated log-likelihood obtained by switching y_i from its current cluster to other proposal clusters, and retain the proposal leading to the highest increase in the log-likelihood. We stress that having a close form expression for $\log p(Y, Z|Q)$ is crucial to adopt such strategy. All vectors $y_1, \ldots, y_N$ are switched once. Notice that, since Q is given, if y_i is alone in its cluster, no movement is allowed. This **classification step (CS)** is repeated recursively until no further increase of $\log p(Y, Z|Q)$ is possible. We stress that the CS described so far is not guaranteed to converge to a global optimum. Indeed, CS is a *greedy* step. In order to reduce the risk of being trapped into local optima, one should either run the algorithm several times (for each Q), with different initialization, or run the algorithm with a "clever" initialization of Z, for instance the one obtained by k-means clustering. We adopt this latter solution in experiments in Section 4. However, notice that also in case of smart initialisation, it might be useful to run the CS several times, randomizing over the switch order of the observations.

The CS is described in more details in Appendix E, where some original results are reported in order to reduce the computational complexity of this step. Instead, we conclude this section with a couple of important remarks. First, we notice that the algorithm described in this section is reminiscent of the Classification EM algorithm (C-EM, Celeux and Govaert, 1991) that replaces the expectation step of the standard EM algorithm with a classification step similar to the one described here. However, while EM algorithms maximize the complete data log-likelihood with respect to the model parameters, here, those parameters are integrated away and the maximization is performed with respect to the cluster assignments alone. Second, it has been shown in the literature that, in latent variable models, likelihood maximization strategies involving a *greedy* search over the group labels

[1] More generally, when Q in not (a priori) uniformly distributed, the selected Q is the one maximizing $\log p(Y, Z_Q^*|Q) + \log p(Q)$, where the last term is independent of Z_Q^*.

can outperform alternative variational approaches in terms of classification accuracy (see for instance Bertoletti et al., 2015; Côme and Latouche, 2015; Corneli et al., 2016).

4 Experiments

Based on simulation studies, this section compares the Bayesian approach previously described[2] with some alternative methods, both in terms of pure clustering (Q fixed) and model selection. The methods considered for comparison are:

1. flexmix[3]: in the eponymous R package, it is a function fitting mixtures of generalized linear models. Here, it is used to fit the generative model in Eq. (3). Since it is the frequentist version of the generative model introduced in Section 2, we consider flexmix as our major competitor.

2. funHDDC[4]: it performs (in R) clustering and model selection of functional longitudinal data based on a latent mixture model which fits high dimensional data in group-specific sub-spaces (Bouveyron and Jacques, 2011; Schmutz et al., 2020).

3. BayesianGaussianMixture[5]: from the Python library scikit-learn, it is used in order to compare our approach with a very popular model based clustering technique. Here, longitudinal data are seen as unconstrained multivariate mixtures of Gaussian distributions (despite the discussion in Appendix A), possibly with non diagonal covariance matrices. A prior Dirichlet process, allowing for a potentially infinite number of components is considered and the cluster assignments are estimated via variational inference.

4. mclust[6]: in the eponymous R package, it fits mixtures of Gaussian distributions to the data, in a frequentist framework. Definitely a gold standard in model based clustering.

4.1 Setup I

Aim. One of the major claims of this paper is that, in longitudinal data mixture analysis, our Bayesian approach might outperform asymptotic model selection cri-

[2] The Python code implementing the clustering/model selection method described in this paper is publicly available at https://github.com/marcogenni/BayesianLongitudinal.

[3] https://cran.r-project.org/web/packages/flexmix/index.html.

[4] https://cran.r-project.org/web/packages/funHDDC/index.html.

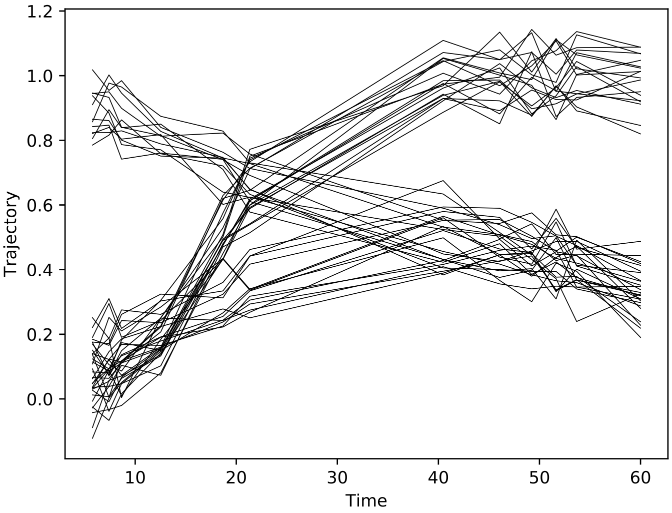
[5] https://scikit-learn.org/stable/modules/mixture.html#variational-bayesian-gaussian-mixture.

[6] https://cran.r-project.org/web/packages/mclust/index.html.

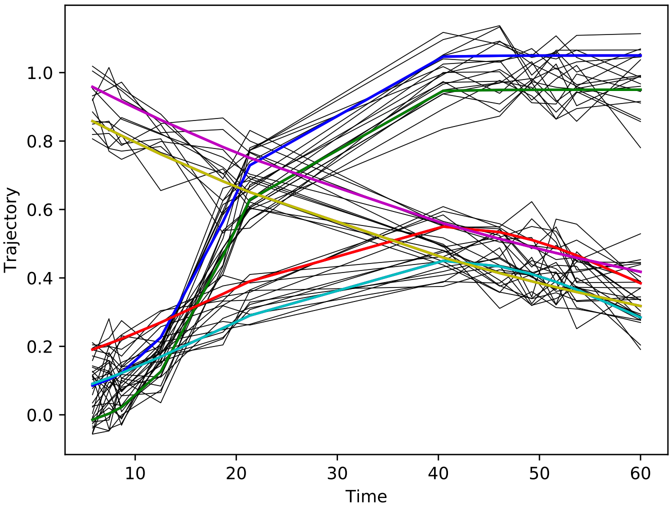
teria (BIC and ICL) in some scenarios. This might happen either because BIC tends to overestimate the number of components and/or the number of observations is not large enough to ensure the convergence of the asymptotic criteria to the corresponding posterior probabilities. This section focuses on such scenarios.

Settings. An increasing number of simulated trajectories, around $Q = 6$ mean signals, is considered. The mean signals are fixed (coloured curves in Figure 2b). Each trajectory is assigned to a group uniformly at random ($\pi_q = 1/6$ for all q) and the number of simulated trajectories increases from $N = 20$ to $N = 190$. Each simulated trajectory is obtained via perturbations around the corresponding signal. We set $D = 12$ points per trajectory, not equally spaced. Figure 2a shows 40 (linearly interpolated) sampled trajectories over the time horizon $[5, 60]$. The same number of sampled trajectories and the underlying signals are shown in Figure 2b. For each value of N, we simulated 10 independent data sets (around the same 6 signals) and each model was fitted on each data set testing a value of Q from 1 to 8. In order to avoid convergence to local maxima, each algorithm run 10 times, one for each data set and for each value of Q, provided with a k-means initialization of Z. The estimates leading to the highest value of the objective function were finally retained. Our model and flexmix were equipped with a polynomial kernel of order 4, the choice of the hyper-parameters for our model is discussed later in this section. Finally, we tested two different families of random perturbations, Gaussian and Gumbel, respectively, both centered and with standard deviation $\sigma = 0.5$. The results for model selection are reported in Figure 3. Note that the legend reports our method as “Bayes" and BayasianGaussianMixture as “NPBGMM". For each value of N we report the mean value of the selected Q by each model (over 10 runs). For all models, except Bayes and NPBGMM, the reported Q was selected via ICL, whose calculation was supported by different packages. Interestingly, the differences with respect to BIC were absolutely negligible, knowing that $Q_{BIC} \geq Q_{ICL}$. In case of flexmix we always found that $Q_{BIC} = Q_{ICL}$.

Discussion. Although in a context of mixture density estimation it is of course desirable to be able to select $Q = 6$ mixture densities, in a *clustering* perspective it is fully reasonable to desire a model selection criterion selecting three main groups of trajectories, especially for small N, as it can be understood when looking at Figure 2a (three groups of well separated trajectories and not six). This point was raised and discussed in more general terms in Section 1.1. Thus, when looking for “clusters", both flexmix and mclust might be overestimating the number of groups for small values of N both in case of Gaussian and Gumbel perturbations (in the Gaussian case the overestimation is more striking). Not surprisingly, a Gaussian mixture model is unfit to model longitudinal data with Gumbel residuals and this



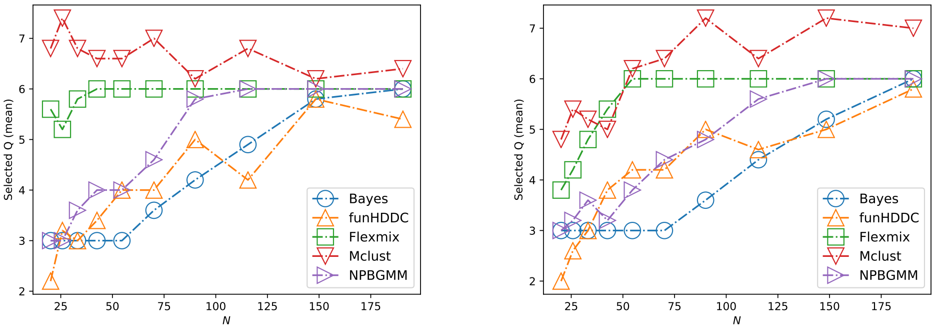
(a) Sampled trajectories



(b) Sampled trajectories with underlying signals

Figure 2: Figure 2a shows $N = 40$ random trajectories distributed around $Q = 6$ signals (not showed). Figure 2b shows another sample of $N = 40$ trajectories as well as the underlying signals. In both cases, the random trajectories are obtained by means of centered Gaussian perturbations around the mean signals ($\sigma = 0.05$).

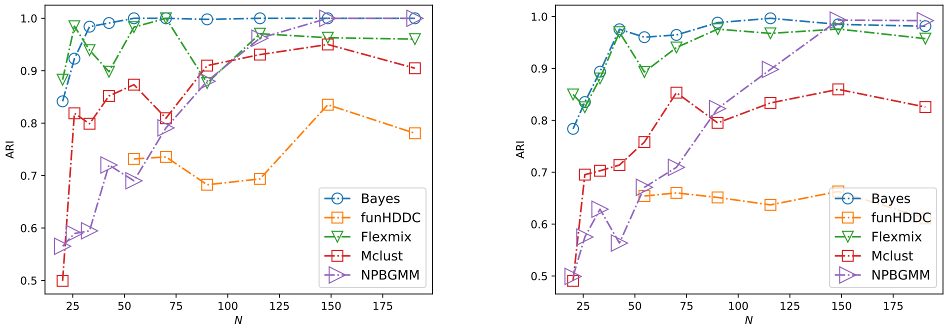
might explain why the actual number of components is not fully recovered by



(a) Gaussian perturbations

(b) Gumbel perturbations

Figure 3: The mean Q selected by each model over ten runs in case of Gaussian residuals (Figure 3a) and Gumbel residuals (Figure 3b).



(a) Gaussian perturbations

(b) Gumbel perturbations

Figure 4: Adjusted Rand indexes obtained by the clustering algorithms for $Q = 6$. Gaussian residuals in Figure 4a, Gumbel residuals in Figure 4b.

mclust, in Figure 3b, even for large N. flexmix always selects more groups than our model and, interestingly, BIC and ICL produce the very same selected Q. This suggests that also the over penalization induced by ICL (see Appendix B) fails to produce a more conservative model selection criterion than BIC. In our model (Bayes), the role of the hyper-parameters is crucial. We will come onto this point in a while. The other competitor approaches (funHDDC and NPBGMM) seem to suffer less overestimation issues for small N. However, the low values of Q estimated by funHDDC are certainly dues to a poor clustering more then to a more penalizing model selection design. This can be seen in Figure 4. It reports the mean Adjusted Rand indexes (ARIs, Rand, 1971) (over 10 runs) obtained by each method when fixing $Q = 6$. The first four values for funHDDC are not reported since, for $Q = 6$, the log-likelihood went to minus infinity and the model produced NA. Starting from $N > 50$ the ARIs can always be computed but they are lower then those obtained by other methods. We think that funHDDC might need more observations per trajectory to work properly. On the opposite, our clustering method outperforms the competitors most of the time. We stress that, for values of N around 50 trajectories, the ARIs are almost one, meaning that the model can perfectly recover the $Q = 6$ mixture components. Still, if it is allowed to choose the number of clusters, it selects $Q = 3$. NPBGMM has good clustering performances for high N. The fact that it needs more observations to recover the true clustering is consistent with the discussion in Appendix A. Note, however, that NPBGMM is not capable of performing clustering with a fixed number of groups. This might penalize it for smaller values of N when it selects $Q < 6$. Several values of the hyper-parameters were tested both for the mean precision and the weight concentration priors in NPBGMM but the best results were always obtained with the default parameters provided by the function.

The experiments reported so far were performed with the simple settings showed in Table 1. To simplify the hyper-parameter choice we assumed a symmetric Dirchlet distribution in Eq. 10 and fixed $\eta_q = \eta$ for all q. In this setup, the value of α was not relevant when the algorithm was provided with a "smarter" k-means initialization. On the contrary, we observed that with a random initialization of Z, the choice of the Dirichlet parameter α had an impact on the final clustering (not reported). In more detail, we noticed that the Classification Step detailed in Appendix E tended to switch too many observations at once, thus emptying some clusters. A similar remark was previously made by Côme and Latouche (2015); Corneli et al. (2016) in the context of graph clustering. However, such a tendency to over-switching can be resisted by picking a large value for the initial α (see Ch.4 of Fruhwirth-Schnatter et al., 2019, for a detailed discussion about the Dirichlet prior hyper-parameters in mixture analysis.). A value of $\alpha = 10$ or even higher can make the job in this framework. We noticed that the value of the hyper-parameters a and b did not impact on the final result. Instead, the hyper-parameter η has an important role. We performed again model selection in the same settings described at the beginning of this section, with Gaussian residuals, but testing several values

Parameter	Value
η	1.0
a	1.0
b	1.0
α	10

Table 1: Values of the prior hyper-parameters.

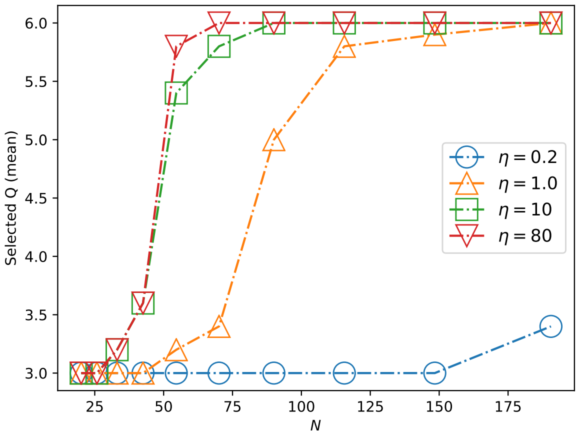


Figure 5: Selection of Q performed by our model, with Gaussian perturbations, increasing number of trajectories and for different values of the prior hyper-parameter η.

of η. As it can be observed in Figure 5, our model selection criterion could even be more parsimonious for small values of η, thus needing higher values of N to select highers Q's. On the opposite, higher values of η allow us to select a higher number of clustering components for smaller values of N. However, the “convergence" of our model selection criterion toward its asymptotic counterpart (ICL) is never fulfilled when $N < 30$: the selected Q is still 3.

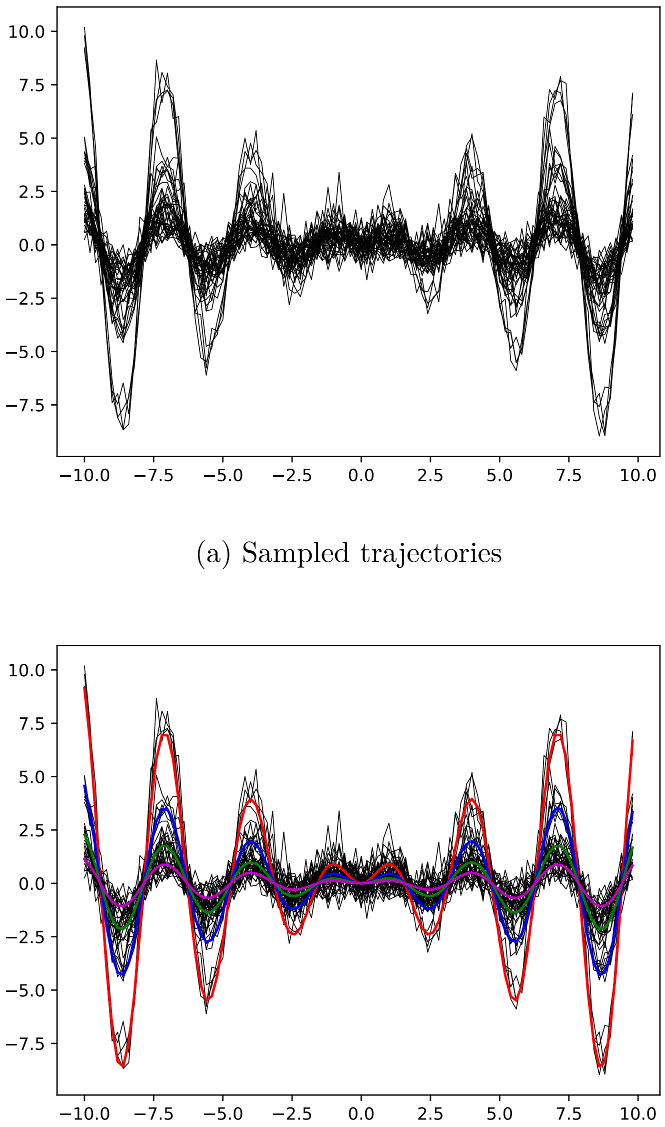
4.2 Setup II

Aim. As stated in Section 2.1, the Bayesian formulation of the generative model in Eq. 2 allows us to directly fit mixtures of linear, polynomial or non-parametric regression models to the data. A non-parametric scenario is considered in this section.

Settings. A fixed number of $N = 40$ (random) trajectories is considered. The trajectories are obtained by means of centred random Gumbel perturbations around $Q = 4$ signals

$$s_b(x) := \frac{\sin(3x)}{bx}, \qquad b = \{1, 2, 4, 8\}.$$

Each trajectory consists of $D = 100$ measurements and the standard deviation σ of the Gumbel perturbations varies in $[0.5, 1.2]$. Gumbel perturbations were preferred to Gaussian residuals in order to make the experiment more challenging.



(a) Sampled trajectories

(b) Sampled trajectories with underlying signal

Figure 6: Figure 6a shows $N = 40$ random trajectories distributed around $Q = 4$ signals (not showed). Figure 6b show the same sample with the underlying signals. The random trajectories are obtained by means of centered Gumbel perturbations around the mean signals ($\sigma = 0.5$).

Figures 6a and 6b show a sample of trajectories over the time horizon $[-10, 10]$, with and without the underlying signals, respectively, for a value of $\sigma = 0.5$. Unlike the previous setup, here trajectories are not assigned to clusters in equal proportions, but according to the following scheme: 4 trajectories around $s_1(\cdot)$, 8 trajectories around $s_2(\cdot)$, 12 trajectories around $s_3(\cdot)$ and, finally, 16 trajectories around $s_4(\cdot)$. Note that, such a setting is more challenging with respect to the equally balanced case, since most of the trajectories are affected to $s_3(\cdot)$ and $s_4(\cdot)$ who are the nearest signals. As long as σ increases, it becomes impossible to separate these two clusters. For different values of σ, 10 independent samples of N trajectories each simulated. Our model was equipped with a RBF kernel

$$[\Phi\Phi^T]_{jl} = \exp\left(-\frac{(t_j - t_l)^2}{2\gamma}\right)$$

with $\gamma = 0.04$, selected by cross validation. The hyper-parameters are set as in the previous setting, except for η, whose role is discussed at the end of this section. We stress that, once fitted to the data, our approach implicitly provides estimates for the mean signals in Figure 6b. Indeed, the estimated parameters (as well as the estimated Z) can be used to compute the mean and variance of the (Gaussian) predictive distribution $p(y_i(t)|y_i, \hat{z}_i, \hat{\sigma}^2, \iota)$, where t is a *new* time point corresponding to a potential measurement for the i-th individual. In Figure 7 the estimated mean signals (the mean values of the predictive distributions over time) are plotted together with the original simulated data clustered by our algorithm. Dashed lines delimit 95% empirical confidence regions, obtained via sampling according to the estimated predictive distributions. All models were fitted 5 independent times on each simulated dataset, provided with a k-means initialisation. The run leading to the highest value of the objective function was finally retained. The results for clustering and model selection are reported in Figure 8.

Discussion. As it can clearly be seen in Figure 8a, our algorithm and flexmix outperform other approaches. In case of mclust and NPBGMM this is not surprising: in this setup, the number of points for trajectory ($D = 100$) is higher than the number of observations ($N = 40$). Thus, both these approaches work in very high dimension (see also Appendix A). For NPBGMM we tried several options for the maximum number of iterations, the mean precision prior and the weight concentration prior. The best results are reported in Figure 8 and correspond to the default values of the hyper-parameters provided by the Python function.

We also report that, since flexmix does not support mixture of non-parametric regressions, we needed to perform progressively higher order polynomial regressions (up to order 6) in order to obtain the best results showed in Figure 8a. Still, our method slightly outperforms flexmix. For completeness, results of model selection are shown in Figure 8b. As in the previous section, the model selection via ICL

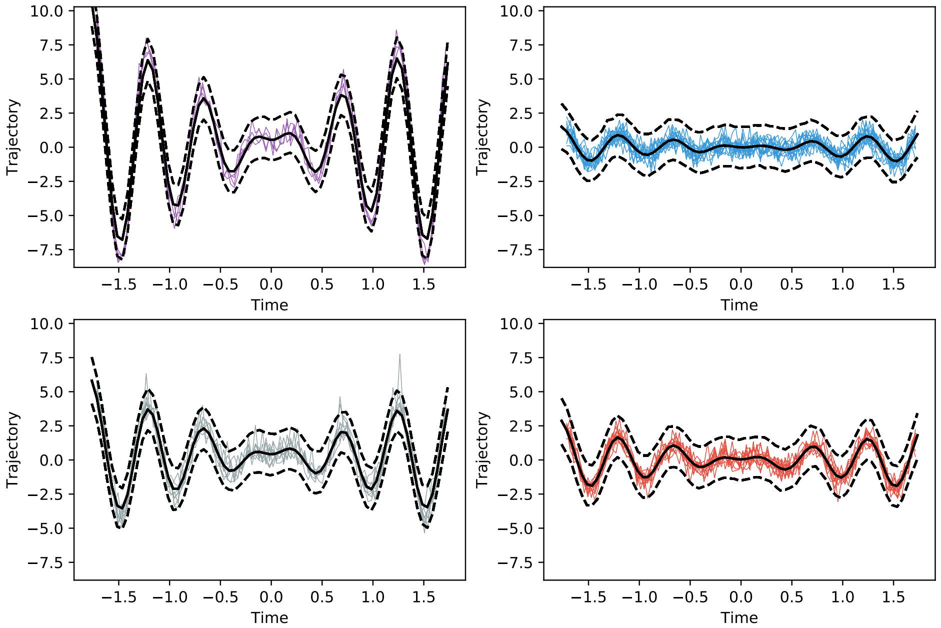
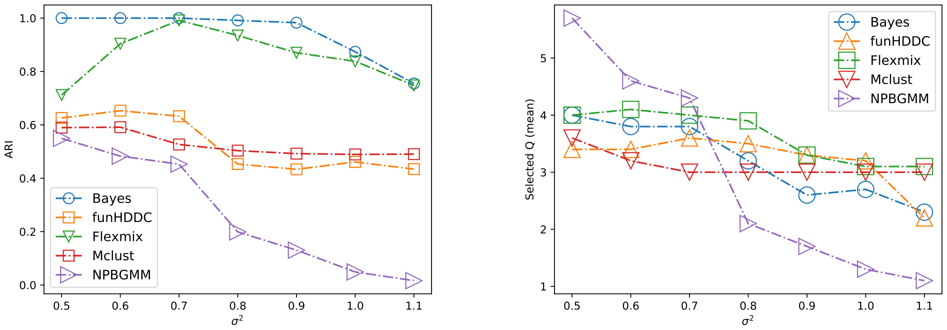


Figure 7: Trajectories clustered via our algorithm ($\sigma = .5$). The solid dark lines are the means of the predictive distributions on time. The dashed lines delimit 95% empirical confidence regions.

criterion was reported for flexmix, Mclust and funHDDC: the differences between ICL and BIC were negligible. Being the number of mixing components smaller in this setup, model selection was less challenging. Basically all methods tend to select $Q = 4$ groups for the smallest value of $\sigma = 0.5$. Then, trajectories around $s_3(\cdot)$ and $s_4(\cdot)$ are less and less separated and models tend to select $Q = 3$ groups and finally $Q = 2$ (our approach still selects $Q = 3$ for some samples, for $\sigma = 1.1$).

Finally, Figure 9 reports the effect of the hyper-parameter η on the selection of Q. Similarly to the previous experiment, smaller values of η can be used to have a more parsimonious model selection. Notice that the value $\eta = 1.2$ is the one used to obtain the curve in Figure 8b. Moreover, as in the previous scenario, different values of η affect the model selection but not the clustering: the four values of η always lead to the Bayes ARIs reported in Figure 8a (not reported).



(a) Clustering

(b) Model Selection

Figure 8: Adjusted Rand indexes obtained by the clustering algorithms for $Q = 4$ and selected number of clusters.

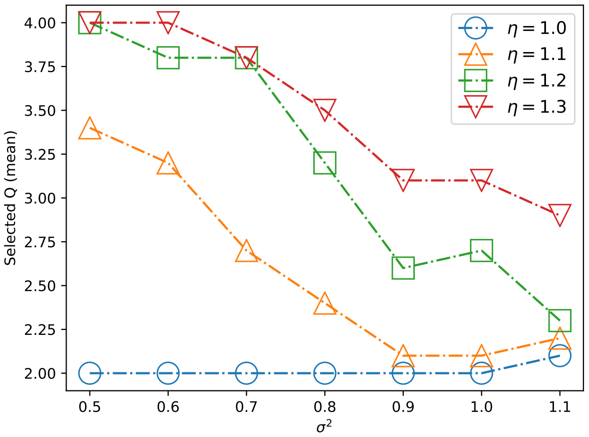
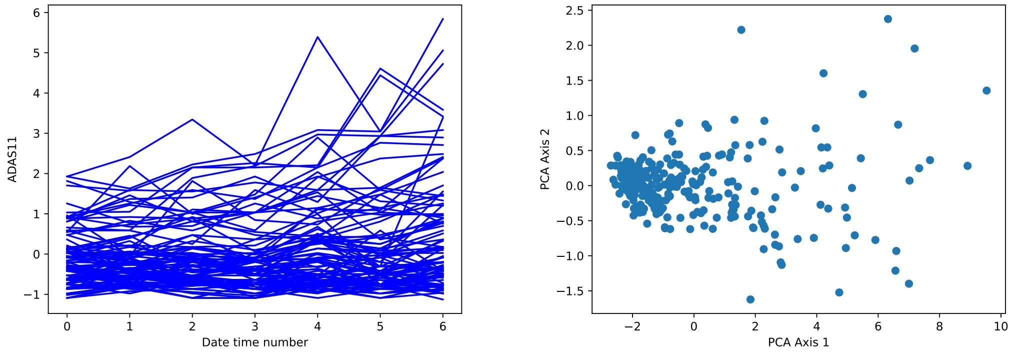


Figure 9: The estimated number of groups selected by Bayes with decreasing η.

5 Alzheimer disease dataset

The longitudinal dataset that we consider (Marco Lorenzi)... Since our generative model accounts for univariate longitudinal data, sharing the same number of measurements for each individual we pre-processed the dataset in the following manner. First, we only considered the feature **ADAS11** known to be highly discriminant for Dementia's diagnoses. Second, since all patients do not share the same number of records (i.e. they were not examined the same number of times) we selected a sub-sample of patients actually having the same number of records. Under this constraint, the value allowing us to maximize the number of



(a) Longitudinal data.

(b) PCA.

Figure 10: Two views of the considered dataset. In 10a, each trajectory reports 7 records of the variable **ADAS11** for a patient. In 10b the same trajectory is reduced to a 2D point via PCA.

observations (i.e. patients) was $D = 7$, corresponding to a sample of $N = 274$ patients. In general, for each patient, records were taken every six months, but exceptions exist in the data. We simply assumed records are taken at the same time for each patient and time dates, in months, (i.e. 6, 12, 18, ...) were centered and scaled to compute the kernel matrix. Thus, the corresponding longitudinal data are collected into a $N \times D$ matrix Y which is the input of the classification steps, as described in Section 3. Another categorical data matrix $N \times D$, say X, associating a diagnosis (NL, NL to MCI, MCI, MCI to Dementia, Dementia) to each patient for each record will be considered in a second time for a qualitative assessment of the estimated clusters. About 30% of the data (in X and Y) is missing and was imputed via a k-nearest neighbours (KNN) algorithm. For Y, KNN was equipped with an Euclidean distance and number of neighbors equal to 2. For X, two different approaches were tested : i) in row X_i we imputed the NAs via the corresponding entries in X_j, with j being of the nearest (Euclidean) neighbour of i, in Y or ii) directly performing KNN on X, provided with the following dissimilarity measure

$$d(X_i, X_j) := \sum_k \frac{\mathbf{1}_{\{X_i \neq X_j\}}}{l(X_i, X_j)}, \tag{14}$$

where the sum is taken over all the indexes corresponding to observed entries (not NA) for both X_i and X_j and $l(X_i, X_j)$ counts the number of such entries. We point out that, in order for the two strategies i) and ii) to be well defined, X_i and X_j need to be *compatible*. Namely, they need to have at least one observed value at the same position and if we want to impute X_{ik}, then X_{jk} must be observed and not missing.

Thus, the neighbors search was always performed among compatible arrays. The KNN approach based on the dissimilarity in Eq. (14) generally provided us with more consistent imputations, provided the degenerative nature of the pathologies being diagnosed. Some examples can be seen in Tables 2-3-4.

0	1	2	3	4	5	6
NL	NL	NL	NaN	NL	NaN	NL
MCI	MCI	MCI	MCI	MCI	NaN	MCI
NL	NL	NL	NaN	NL	NaN	NL
MCI	MCI to Dem	Dem	Dem	Dem	NaN	Dem
NL	NL	NL	NaN	NL	NaN	NL

Table 2: Original data set, first 5 patients.

0	1	2	3	4	5	6
NL	NL	NL	MCI	NL	MCI	NL
MCI	MCI	MCI	MCI	MCI	MCI	MCI
NL	NL	NL	NL	NL	NL	NL
MCI	MCI to Dem	Dem	Dem	Dem	MCI to Dem	Dem
NL	NL	NL	NL	NL	NL to MCI	NL

Table 3: Imputation with method i), first 5 patients.

0	1	2	3	4	5	6
NL	NL	NL	NL	NL	NL	NL
MCI	MCI	MCI	MCI	MCI	MCI to Dem	MCI
NL	NL	NL	NL	NL	NL	NL
MCI	MCI to Dem	Dem	Dem	Dem	Dem	Dem
NL	NL	NL	NL	NL	NL	NL

Table 4: Imputation with method ii), first 5 patients.

The whole data sample can be seen in Figure 10. In Figure 10a, each trajectory reports 7 records of the variable **ADAS11** for a patient. As it can be seen, the trajectories are not clearly separable. This can also be seen in Figure 10b, where each trajectory is reduced to a 2D point via PCA.

Given the relatively small value of $D = 7$ as well as the simple shape of the trajectories (they basically are constant or increasing functions), we fit out generative model to the data by adopting a polynomial kernel of order 3. The number Q of potential mixtures component was selected in the range $\{1, \ldots, 9\}$, and for any

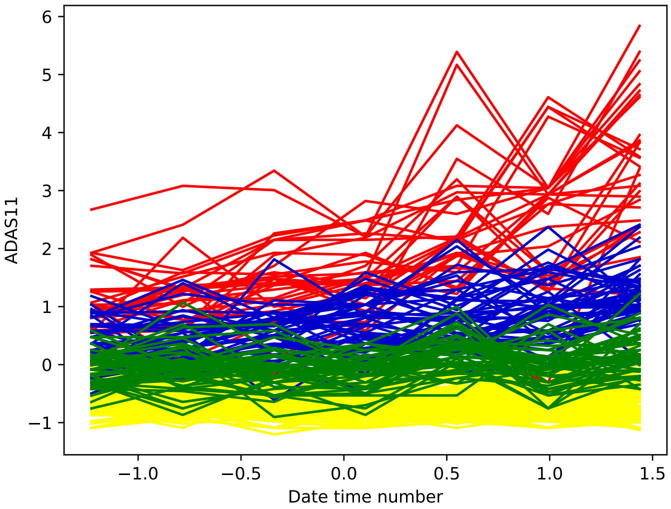


Figure 11: Clustered trajectories, $Q = 4$ groups with hyper-parameters leading to the highest Silhouette score (see Table 5).

value Q, a k-means initialisation was adopted. As discussed in previous sections, the choice of the hyper-parameters is crucial and can lead to very different (numbers of) clusters. In general, that choice should be guided by a prior knowledge or need. For instance, as discussed in Section 4, large values of $\eta > 1.0$ and small values of $\alpha < .5$ (symmetric Dirichlet prior) favor a less parsimonious model selection as well as highly imbalanced clusters. Thus, such values could be suited to detect anomalous trajectories/clinical progressions (Should we show an example of fit?) Here, we adopt a more agnostic approach, guided by the data, via cross validation. Based on the intuitive notion that a good clustering provides users with compact and well separated groups, we adopt the Silhouette score (Rousseeuw, 1987) to select the hyper-parameters. In more detail, we set $a = 1$ and let b, η and α (symmetric Dirichlet prior) vary in $\{0.5, 1, 2, 4, 8\}$, $\{0.01, 0.05, 0.1, 0.5, 1\}$ and $\{0.5, 1, 2, 4, 8\}$, respectively. For each one of the corresponding 125 settings, we fit our model to the data and select the optimal Q in the range $\{1, \dots, 9\}$. For that value of Q the Silhouette score was computed. We recall that the Silhouette score ranges in [-1,1] and higher values indicate that most points are well ranged in their clusters. The three best settings are reported in Table 5. The best setting led to $Q = 4$ groups of trajectories, as shown in Figure 11. In general, the highest Silhouette scores were obtained via large values of b (4 or 8), corresponding to flatter priors for σ_q, *always* and small values of η.

To better inspect the composition of each cluster, we identified each trajectory

a	b	η	α	*Silhouette*
1.0	8.0	0.01	1.0	**0.438**
1.0	8.0	0.05	0.5	0.395
1.0	8.0	0.1	2.0	0.394

Table 5: Silhouette scores corresponding to different values of the hyper-parameters.

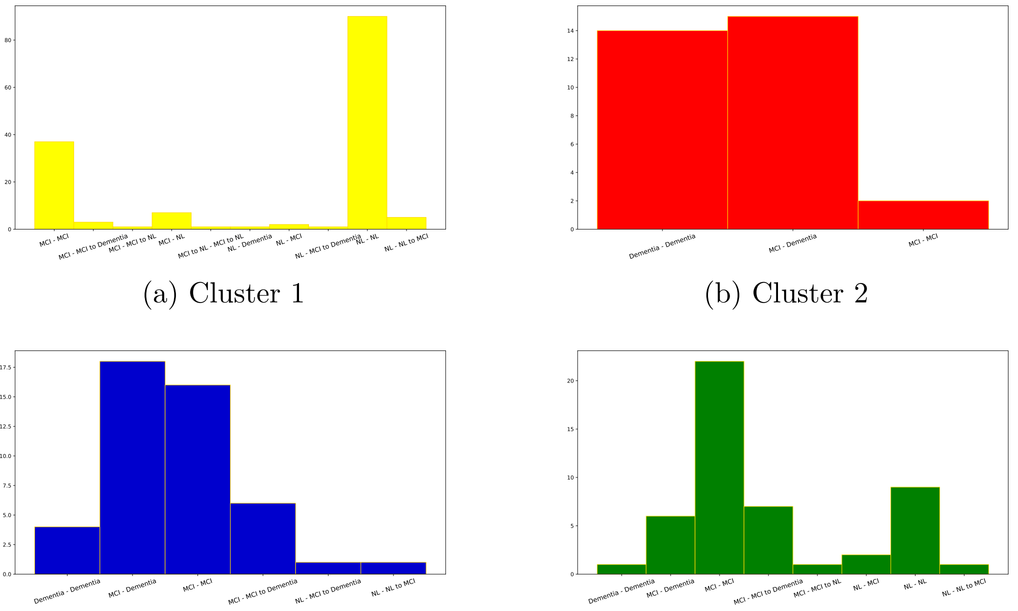
with the diagnoses associated with the first and last record for the corresponding patient. So, for instance, a trajectory labelled as “MCI - Dementia" corresponds to a patient who was diagnosed with MCI after the first record and whose disease degenerated to Dementia. Thus, in Figure 12a we see that Cluster 1 (yellow) mostly contains sane patients (NL-NL) or border line patients (MCI - NL or NL - NL to MCI). A non-negligible number of MCI-MCI patients (almost 40) is the cluster and this is not surprising since, as previously discussed the data are not clearly separable. On the contrary, Cluster 2 (red) contains the "highest" trajectories, mostly corresponding to patients that are affected from Dementia at the end of the trial (either Dementia - Dementia or MCI - Dementia). Cluster 3 (blue) and Cluster 4 (green) group all the in between cases of patients that substantially are affected by MCI and degenerating toward Dementia (Cluster 3) or being stable (Cluster 4).

We point out that our aim is not to classify patients in a non supervised way, but just to show that our approach is flexible enough to provide reasonable groups based on a ground truth information.

6 Discussion and perspectives

The experiments in the previous section suggest that our approach can be used as a viable alternative as a pure clustering and model selection tool in applications. Indeed, it outperforms the state-of-the art when performing clustering (with a fixed Q) of longitudinal data and provides the user with a model selection criterion which is by far more flexible than BIC and ICL. Our criterion can be very parsimonious depending on the choice of the hyper-parameter η. As usual in Bayesian statistics, this parameter must be set by the statistician based on the prior knowledge he/she has about the phenomenon under observation. Smaller prior variances should be preferred when the aim is to detect very well separated clusters of trajectories, higher prior variances when one is interested in the number of Gaussian mixing components needed to fit the data distribution. The higher the number of observations the less crucial the choice.

Although both the proposed Bayesian modeling (based on conjugated prior dis-



(a) Cluster 1

(b) Cluster 2

(c) Cluster 3

(d) Cluster 4

Figure 12: Caption

tributions on the model parameters) and the inference strategy (based on a greedy search over trajectory labels) can easily be extended to multivariate regression models not involving time or to longitudinal data mixtures whose measurements are not taken at the same times, we recall that, on those cases, the results presented in Appendix E would no longer be valid. Future researches might focus on strategies to keep under control the complexity of the estimation procedure under less restrictive assumptions. Also, it would be interesting to extend the generative model considered here in order to account for individual effects and/or autoregressive components.

References

Baudry, J.P., Raftery, A.E., Celeux, G., Lo, K., Gottardo, R., 2010. Combining mixture components for clustering. Journal of computational and graphical statistics 19, 332–353.

Bertoletti, M., Friel, N., Rastelli, R., 2015. Choosing the number of clusters in a finite mixture model using an exact integrated completed likelihood criterion. Metron 73, 177–199.

Biernacki, C., Celeux, G., Govaert, G., 2000. Assessing a mixture model for clustering with the integrated completed likelihood. IEEE Trans. Pattern Anal. Machine Intel 7, 719–725.

Blei, D.M., Jordan, M.I., et al., 2006. Variational inference for dirichlet process mixtures. Bayesian analysis 1, 121–143.

Bouveyron, C., Jacques, J., 2011. Model-based clustering of time series in group-specific functional subspaces. Advances in Data Analysis and Classification 5, 281–300.

Celeux, G., Govaert, G., 1991. A classification em algorithm for clustering and two stochastic versions. Computational Statistics Quaterly 2, 73–82.

Côme, E., Latouche, P., 2015. Model selection and clustering in stochastic block models based on the exact integrated complete data likelihood. Statistical Modelling 15, 564–589.

Corneli, M., Latouche, P., Rossi, F., 2016. Exact icl maximization in a non-stationary temporal extension of the stochastic block model for dynamic networks. Neurocomputing 192, 81–91.

Erosheva, E.A., Matsueda, R.L., Telesca, D., 2014. Breaking bad: Two decades of life-course data analysis in criminology, developmental psychology, and beyond. Annual Review of Statistics and Its Application 1, 301–332.

Fruhwirth-Schnatter, S., Celeux, G., Robert, C.P., 2019. Handbook of mixture analysis. Chapman and Hall/CRC.

Griffiths, T.L., Steyvers, M., 2004. Finding scientific topics. Proceedings of the National Academy of Sciences 101, 5228–5235.

MacEachern, S.N., Müller, P., 1998. Estimating mixture of dirichlet process models. Journal of Computational and Graphical Statistics 7, 223–238.

Muthén, B., Asparouhov, T., 2008. Growth mixture modeling: Analysis with non-gaussian random effects. Longitudinal data analysis 143165.

Nagin, D.S., NAGIN, D., et al., 2005. Group-based modeling of development. Harvard University Press.

Rand, W.M., 1971. Objective criteria for the evaluation of clustering methods. Journal of the American Statistical association 66, 846–850.

Rousseeuw, P.J., 1987. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. Journal of computational and applied mathematics 20, 53–65.

Schmutz, A., Jacques, J., Bouveyron, C., Cheze, L., Martin, P., 2020. Clustering multivariate functional data in group-specific functional subspaces. Computational Statistics , 1–31.

Silvester, J.R., 2000. Determinants of block matrices. The Mathematical Gazette 84, 460–467.

Teh, Y.W., Jordan, M.I., Beal, M.J., Blei, D.M., 2005. Sharing clusters among related groups: Hierarchical dirichlet processes, in: Advances in neural information processing systems, pp. 1385–1392.

A Relation with multivariate Gaussian mixture models

To simplify the exposition, let us consider here the simple case where $\phi(\cdot)$ is the identity function. Then, Eq. (2) reduces to

$$y_i = \bar{t}\beta_{z_i} + \sigma\epsilon_i,$$

where β_{z_i} is a scalar and $\bar{t} = (t_1, \ldots, t_D)^T \in \mathbb{R}^D$. Thus

$$y_i | z_i \sim \mathcal{N}\left(\bar{t}\beta_{z_i}, \sigma^2 I_D\right).$$

First, we stress that, since a Gaussian distributions is fully characterized by its mean and variance matrix, a permutation of the times inside $\bar{t}$ induces a different probability distribution on y_i. Thus, unless $\beta_1 = \cdots = \beta_Q = 0$, time matters. Second, recalling that $y_1, \ldots y_N$ are assumed independent, the above equation defines a **constrained** version of a multivariate Gaussian mixture model, since

$$p(y_i|\pi, \beta, \sigma^2) = \sum_{q=1}^{Q} \pi_q g(y_i; \bar{t}\beta_q, \sigma^2 I_D), \tag{15}$$

where $\pi := (\pi_1, \ldots, \pi_Q)^T$ are the mixing proportions, $g(\cdot; \mu, \Sigma)$ denotes the pdf of a multivariate Gaussian distribution with mean μ and variance Σ and z_i was integrated out. Since, y_i follows a multivariate Gaussian mixture distribution, would it be reasonable to attack the problem of clustering $y_1, \ldots, y_N$ via standard *unconstrained* Gaussian mixtures machinery (thus ignoring the data dependence on time)? Not really. It is worth noticing that the generative model induced by the above equation has $2Q$ free parameters to estimate. On the contrary, if also we saw that model as a not constrained Gaussian mixture model with spherical shared variance (the most parsimonious version), namely

$$p(y_i|\pi, \beta, \sigma^2) = \sum_{q=1}^{Q} \pi_q g(y_i; \mu_q, \sigma^2 I_D),$$

with $\mu_q \in \mathbb{R}^D$, unknown, the above equation would count $Q(D+1)$ free parameters to estimate. Thus, as long as $D \geq 2$ the longitudinal formulation is more parsimonious. In particular, in a high dimensional framework, the difference between the two approaches would be dramatic.

B ICL and BIC

We consider a standard mixture model where the N observed variables are denoted by $X := (x_1, \ldots, x_N)$ and the corresponding latent variables by $Z = (z_1, \ldots, z_N)$.

The unknown number of mixing components is K. The (asymptotic) **BIC** and **ICL** criteria are defined as follows

$$BIC_K := \max_\theta \log p(X|\theta, K) - \frac{\nu(K)}{2} \log N \tag{16}$$

and

$$ICL_K := \max_\theta \log p(X, Z|\theta, K) - \frac{\nu(K)}{2} \log N, \tag{17}$$

where θ denotes the set of the model parameters, $\nu(K)$ is the number of model parameters and $\log p(\cdot)$ denotes the log density of the observations.

Remark. *The ICL criterion in Eq.* (17) *is an approximation of the marginal log-likelihood*

$$\log p(X, Z|K) = \log \int_\theta p(X, Z|\theta, K) p(\theta|K) d\theta, \tag{18}$$

*where the model parameters are integrated out (Biernacki et al., 2000). If the prior distribution $p(\theta|K)$ is conjugated, the quantity on the left hand side of the above equation can be computed explicitly. We sometimes call it **exact ICL**.*

The following notations are adopted

$$\hat{\theta} = \arg\max_\theta \log p(X|\theta, K), \tag{19}$$

$$\overline{\theta} = \arg\max_\theta \log p(X, Z|\theta, K) \tag{20}$$

and we stress that, in general, $\hat{\theta} \neq \overline{\theta}$.

The following Proposition formally shows that ICL_K in Eq. (17) is a lower bound of BIC_K. This result was mentioned in (Biernacki et al., 2000; Baudry et al., 2010) but not formally proven.

Proposition 1.

$$ICL_K \leq BIC_K.$$

Proof.

$$\begin{aligned} ICL_K - BIC_K &= \log p(X, Z|\overline{\theta}) - \log p(X|\hat{\theta}) \\ &= \log \frac{p(X, Z|\overline{\theta})}{p(X|\hat{\theta})} \\ &= \log \frac{p(X, Z|\overline{\theta}) p(X|\overline{\theta})}{p(X|\overline{\theta}) p(X|\hat{\theta})} \\ & \log p(Z|X, \overline{\theta}) + \log \frac{p(X|\overline{\theta})}{p(X|\hat{\theta})} \leq 0, \end{aligned}$$

where the dependence on K was omitted for simplicity and the last inequality comes from the discrete nature of the random variables $z_1, \ldots, z_N$ and the definitions of $\hat{\theta}$ and $\overline{\theta}$. □

C Integrating with respect to σ^2

In Section 2.1.1 we saw that the marginal conditional density $p(Y|Z, \sigma^2, Q)$ is

$$\begin{aligned} p(Y|Z, \sigma^2, Q) &= \prod_{q=1}^{Q} p(Y_q|Z, \sigma^2) \\ &= \frac{1}{(2\pi\sigma^2)^{\frac{DN}{2}} \prod_{q=1}^{Q} \sqrt{\det(G_q)}} \exp\left(-\frac{1}{2\sigma^2} \sum_{q=1}^{Q} Y_q^T (G_q)^{-1} Y_q\right), \end{aligned}$$

where $G_q \in \mathbb{R}^{DC_q \times DC_q}$ is the block matrix introduced in Eq. (13). Since, Eq. (9) states that

$$p(\sigma^2|a, b) = \frac{b^a}{\Gamma(a)} \left(\frac{1}{\sigma^2}\right)^{a-1} \exp\left(-\frac{b}{\sigma^2}\right) \mathbf{1}(\sigma^2)_{]0,\infty[},$$

when looking at the joint conditional density $p(Y, \sigma^2|Z, Q)$ as a function of σ^2 we recognize the pdf of an Inverse Gamma distribution $\text{I}\Gamma\left(a + \frac{DN}{2}, b + \frac{1}{2}\sum_{q=1}^{Q} Y_q^T G_q^{-1} Y_q\right)$. Therefore, σ^2 can be integrated out to obtain

$$p(Y|Z, Q) = \frac{1}{(2\pi)^{\frac{ND}{2}} \prod_q \sqrt{\det(G_q)}} \frac{b^a}{\Gamma(a)} \frac{\Gamma\left(\frac{DN}{2} + a\right)}{\left(b + \frac{1}{2}\sum_q Y_q^T (G_q^{-1}) Y_q\right)^{\frac{DN}{2}+a}}. \tag{21}$$

D Integrating with respect to π

The second integral on the right hand side of Eq. (7) can be computed in a similar fashion. Due to Eq. (10), the posterior distribution $p(\pi|Z, Q)$ is still a Dirichlet. It can easily be seen that

$$p(Z|Q) = \frac{\Gamma\left(\sum_{q=1}^{Q} \alpha_q\right)}{\prod_{q=1}^{Q} \Gamma(\alpha_q)} \frac{\prod_{q=1}^{Q} \Gamma(C_q + \alpha_q)}{\Gamma\left(N + \sum_{q=1}^{Q} \alpha_q\right)}, \tag{22}$$

where we recall that C_q is the number of trajectories in cluster q.

E Classification step in depth

Consider the observation $y_i \in \mathbb{R}^D$ and assume it currently belongs to the q-th cluster, namely $z_i = q$. The change in the integrated log-likelihood due to a switch of y_i from the q-th cluster to the l-th cluster can be computed as

$$\begin{aligned} \Delta_{ll}^{i:q\to l} &:= \log p(Y, Z^a|Q) - \log p(Y, Z^b|Q) \\ &= \log p(Y|Z^a, Q) - \log p(Y|Z^b, Q) \\ &+ \log p(Z^a|Q) - \log p(Z^a|Q), \end{aligned} \tag{23}$$

where Z^a and Z^b denote the configurations *after* and *before* the switch, respectively. As it can be seen by looking at Eqs. (21)-(22), the calculation of $\Delta_{ll}^{i:q\to l}$ basically boils down to compute i) the determinant of G_q and ii) the quadratic term $Y_q^T G_q^{-1} Y_q$, for all $q \leq Q$. We report in the following some results allowing us to speed up the calculation of these two terms.

First term. First, recall that

$$G_q = \left(\begin{array}{c|c|c|c} B_q & A_q & \dots & A_q \\ \hline A_q & B_q & \dots & A_q \\ \hline A_q & A_q & \ddots & A_q \\ \hline A_q & \dots & \dots & B_q \end{array}\right) \tag{24}$$

where $A_q = \eta_q \Phi\Phi^T \in \mathbb{R}^{D\times D}$ and $B_q = A_q + I_D$. Since the size of G_q changes whenever an observation is switched from one cluster to another, we need a fast way to compute $\det(G_q)$[7]. Theorem 1 in Silvester (2000), whose precise formulation is reported in Appendix F, can help us. This theorem basically allows us to compute $\det(G_q)$ in two steps:

1. a first intermediate determinant (ID_q) is computed as if A_q, B_q in Eq. (24) where numbers
$$ID_q := \overline{\det}(G_q) \in \mathbb{R}^{D\times D}, \tag{25}$$
where the over line is used to differentiate this determinant from the real one, that we are actually trying to compute. Then,

2. since ID_q is itself a matrix, $\det(G_q)$ is computed as
$$\det(G_q) = \det(ID_q). \tag{26}$$

According to Theorem 1 in Silvester (2000), the above equality holds as long as all the possible matrix products between two blocks in G_q are commutative. This condition is fulfilled as stated in the following

Proposition 2. *The product between each pair of blocks of G_q is commutative.*

Proof. For simplicity, in this proof the subscript q is removed from A_q and B_q, since not needed. Clearly the products AA and BB are commutative. Moreover,

$$AB = A(A + I_D) = AA + A = (A + I_D)A = BA$$

and the proposition is proven. □

[7] For instance, the cost of computing $\det(G_q)$ via an LU decomposition is $\mathcal{O}(D^3N^3)$ and this approach is used by most linear algebra libraries.

Now we can state the following Theorem

Theorem 1. *The determinant of G_q can be computed in $\mathcal{O}(D)$ as*

$$\det(G_q) = \prod_{j=1}^{D}(1 + C_q\lambda_j^{(q)}), \tag{27}$$

where $\lambda_1^{(q)}, \dots, \lambda_D^{(q)}$ are the eigenvalues of A_q and C_q is the cardinality of the q-th cluster (see Eq. (12)*).*

Proof. The proof of this Theorem relies on Lemma 1 in Appendix F, stating that ID_q in Eq. (25) is

$$ID_q = I_D + C_qA_q,$$

which is a matrix in $\mathbb{R}^{D\times D}$. Now, since A_q is symmetric it admits a diagonal representation $A_q = Q_q\Lambda_qQ_q^T$, where $\Lambda_q \in \mathbb{R}^{D\times D}$ is a diagonal matrix whose non null entries are the eigenvalues of A_q and Q_q is an orthogonal matrix whose columns are the corresponding eigenvectors. Thus

$$\begin{aligned}\det(ID_q) &= \det(I_D + C_qQ_q\Lambda_qQ_q^T)\\ &= \det\left(Q_q(I_D + C_q\Lambda_q)Q_q^T\right)\\ &= \det\left(I_D + C_q\Lambda_q\right)\\ &= \prod_{j=1}^{D}(1 + C_q\lambda_j^{(q)}).\end{aligned}$$

□

Second term. The quadratic form

$$Y_q^TG_q^{-1}Y_q \tag{28}$$

is now considered. Adopting the notation in Eq. (24), Theorem 3 in Appendix F states that the inverse matrix G_q^{-1} is also a diagonal block matrix

$$G_q^{-1} = \left(\begin{array}{c|c|c|c} V_q & W_q & \dots & W_q \\ \hline W_q & V_q & \dots & W_q \\ \hline W_q & W_q & \ddots & W_q \\ \hline W_q & \dots & \dots & V_q \end{array}\right)$$

where

$$\begin{aligned}W_q &:= -(I_D + C_qA_q)^{-1}A_q,\\ V_q &:= W_q + I_D.\end{aligned} \tag{29}$$

Thus, to compute the quadratic form in Eq. (28) it is not required to invert the whole G_q but only the matrix $(I_D + C_q A_q)$. Notice that the computational cost of this operation is independent of the number of observations N[8]. Moreover

$$\begin{aligned}
Y_q^T G_q^{-1} Y_q &= \sum_{i=1}^{C_q} y_i^{(q)^T} V_q y_i^{(q)} + \sum_{i=1}^{C_q} \sum_{\substack{j=1 \\ j \neq i}}^{C_q} y_i^{(q)^T} W_q y_j^{(q)} \\
&= \sum_{i=1}^{C_q} y_i^{(q)^T} I_d y_j^{(q)} + \sum_{i=1}^{C_q} \sum_{j=1}^{C_q} y_i^{(q)^T} W_q y_j^{(q)} \\
&= \|Y_q\|_2^2 + \left(\sum_{i=1}^{C_q} y_i^{(q)^T} \right) W_q \left(\sum_{j=1}^{C_q} y_j^{(q)} \right),
\end{aligned}$$

where $\|\cdot\|_2$ denotes the Euclidean norm and we denoted by $y_i^{(q)} \in \mathbb{R}^D$ the i-th column vector in cluster $y^{(q)}$. Since the sum of all observations in cluster $y^{(q)}$ (namely $\sum_{j=1}^{C_q} y_j^{(q)}$) can be pre-computed before and after each switch and W_q does not depend on i, the last term on the right hand side of the above equation can be computed in $\mathcal{O}(D^2)$ that can be sensibly smaller than $\mathcal{O}(N^2 D^2)$ needed for a direct calculation of $Y_q^T G_q^{-1} Y_q$.

F Linear algebra results

Theorem 2 (Silvester (2000)). *Let R be a commutative subring of $F^{n \times n}$, where F is a field and $F^{n \times n}$ denotes the set of matrices $n \times n$ over F. Let $M \in R^{m \times m}$. Then*

$$\det_F M = \det_F \left(\det_R (M) \right).$$

Lemma 1. *Consider a $\mathbb{R}^{N \times N}$ square matrix A such that*

$$A_{ij} = \begin{cases} a + \epsilon & \textit{if} \quad i = j \\ a & \textit{otherwise} \end{cases}$$

where a, ϵ are two real constants. Then

$$\det(A) = \epsilon^{N-1}(\epsilon + Na). \tag{30}$$

Proof. We proceed by recurrence. For $N = 1$, $A = (a + \epsilon)$ and Eq. (30) is verified. Now, let us assume that Eq. (30) holds for all $i \leq N$. The case where N is an even

[8] For instance, relying on the Gauss-Jordan elimination, the computational cost of the inversion would be $\mathcal{O}(D^3)$, which is smaller than $\mathcal{O}(ND^3)$ required to invert G_q.

number is considered at first. Thus

$$
\begin{aligned}
\det M_{N+1} &= \det \underbrace{\begin{pmatrix} a+\epsilon & a & \dots & a \\ a & a+\epsilon & \dots & a \\ \vdots & & \ddots & \vdots \\ a & a & \dots & a+\epsilon \end{pmatrix}}_{\text{N +1 columns}} \\
&= (a+\epsilon)\det(M_N) - aN \det \underbrace{\begin{pmatrix} a & a & \dots & a \\ a & a+\epsilon & \dots & a \\ \vdots & & \ddots & \vdots \\ a & a & \dots & a+\epsilon \end{pmatrix}}_{\text{N columns}} \\
&= (a+\epsilon)\det(M_N) - a^2 N \det(M_{N-1}) + a^2 N(N-1) \det \underbrace{\begin{pmatrix} a & a & \dots & a \\ a & a+\epsilon & \dots & a \\ \vdots & & \ddots & \vdots \\ a & a & \dots & a+\epsilon \end{pmatrix}}_{\text{N - 1 columns}}
\end{aligned}
$$

and pursuing the recursion we obtain

$$
\begin{aligned}
\det M_{N+1} = {} & (a+\epsilon)\det(M_N) \\
& - a^2 \left(\sum_{i=0}^{N-2} (-a)^i \frac{N!}{(N-i-1)!} \det(M_{N-i-1}) \right) \\
& - a^{N+1} N!,
\end{aligned}
\tag{31}
$$

where the sign of the last term on the right hand side (r.h.s.) of the equality is due to the assumption of an even N. Thanks to the inductive assumption, the first term on the r.h.s of the equality is

$$
\begin{aligned}
(a+\epsilon)\det(M_N) &= (a+\epsilon)\epsilon^{N-1}(\epsilon + Na) \\
&= a\epsilon^{N-1}(\epsilon + Na) + \epsilon^N(\epsilon + Na) \\
&= \epsilon^N(\epsilon + (N+1)a) + a^2 N \epsilon^{N-1}.
\end{aligned}
$$

Similarly, the inductive assumption can bu used to replace $\det(M_{N-i-1}) = \epsilon^{N-i-2}(\epsilon - (N-i-1)a)$ in the second term on the r.h.s. of Eq. (31). Thus, by developing the sum over i we obtain

$$
\begin{aligned}
\det(M_{N-1}) &= \epsilon^N(\epsilon + (N+1)a) \\
&\quad + \cancel{a^2 N \epsilon^{N-1}} - \cancel{a^2 N \epsilon^{N-1}} \\
&\quad + \cdots + \cancel{a^N \epsilon N!} - \cancel{a^N \epsilon N!} \\
&= \epsilon^N(\epsilon + (N+1)a).
\end{aligned}
$$

The case where N is odd is analogous and the lemma is proven. $\square$

Lemma 2. *Consider a $\mathbb{R}^{N\times N}$ square matrix A such that*

$$A_{ij} = \begin{cases} a+\epsilon & \text{if} \quad i=j \\ a & \text{otherwise} \end{cases}$$

where a, ϵ are two real constants. Then the inverse A^{-1} is

$$A^{-1}_{ij} = \begin{cases} \frac{\epsilon+(N-1)a}{\epsilon(\epsilon+Na)} & \text{if} \quad i=j \\ -\frac{a}{\epsilon(\epsilon+Na)} & \text{otherwise} \end{cases} \tag{32}$$

Proof. It suffices to verify that, given A^{-1} in Eq. (32), $AA^{-1} = I_N$. $\square$

Theorem 3. *Consider an invertible square block matrix $M \in \mathbb{R}^{DN\times DN}$ such that*

$$M = \left(\begin{array}{c|c|c|c} B & A & \dots & A \\ \hline A & B & \dots & A \\ \hline A & A & \ddots & A \\ \hline A & \dots & \dots & B \end{array}\right) \tag{33}$$

where the non diagonal blocks $A \in \mathbb{R}^{D\times D}$ are symmetric matrices and the diagonal blocks are $B = A + I_D$. Then, the inverse matrix M^{-1} is still a block matrix

$$M^{-1} = \left(\begin{array}{c|c|c|c} V & W & \dots & W \\ \hline W & V & \dots & W \\ \hline W & W & \ddots & W \\ \hline W & \dots & \dots & V \end{array}\right)$$

where the non-diagonal blocks W are

$$W = -(I_D + NA)^{-1}A, \tag{34}$$

and the diagonal blocks V are

$$V = (I_D + NA)^{-1}(I_D + (N-1)A) = W + I_D. \tag{35}$$

Proof. We need to prove that MM^{-1} is a block matrix whose non-diagonal blocks are matrices in $\mathbb{R}^{D\times D}$ having zero everywhere (henceforth denoted by $\mathbf{0}_{\mathbb{R}^{D\times D}}$) and whose diagonal blocks are I_D. We observe that $(I_D+NA)^{-1} = \left(\overline{\det}(M)\right)^{-1}$, where $\overline{\det}(\cdot)$ is defined in Eq. (25) and $(I_D + NA)$ is invertible thanks to the assumption of invertible M combined with Lemma 1 and Theorem 1 in Silvester (2000). Now, the following notation is introduced

$$D_M := (I_D + NA)^{-1}$$

to simplify the exposition. Moreover, with a slight abuse of notation we denote by $(MM^{-1})_{ij}$ the block (and not the real entry!) at position (i, j) in MM^{-1}. Then, for $j \neq i$

$$\begin{aligned}(MM^{-1})_{ij} &= -BD_MA + AD_M(I_D + (N-1)A) \\ &- (N-2)AD_MA \\ &= -D_MA + AD_M \\ &= \mathbf{0}_{\mathbb{R}^{D\times D}},\end{aligned}$$

where the last equality comes from

$$\begin{aligned}A(I_D + NA) = (I_D + NA)A \qquad &\Rightarrow \\ A = (I_D + NA)A(I_D + NA)^{-1} &\Rightarrow \\ (I_D + NA)^{-1}A = A(I_D + NA)^{-1}&.\end{aligned} \tag{36}$$

Similarly, for $i = j$

$$(MM^{-1})_{ii} = BD_M(I_d + (N-1)A) - (N-1)AD_MA$$

and some easy calculations show that the above quantity is equal to I_D. □