Supplementary material for enriched mixtures of generalised Gaussian process experts

Charles W.L. Gadd
Dept. of Computer Science
Aalto University
Espoo, Finland
cwlgadd@gmail.com

Sara Wade
School of Mathematics
University of Edinburgh
Edinburgh, United Kingdom
sara.wade@ed.ac.uk

Alexis Boukouvalas
PROWLER.io
Cambridge, United Kingdom
alexis@prowler.io

1 GENERALISED GAUSSIAN PROCESS EXPERTS

Examples of generalised GP experts include:

Gaussian: for $y \in \mathbb{R}$, with identity link function,

$$p(y|x, \theta_j) = \mathrm{N}(y|m_j(x), \sigma_j^2).$$

Bernoulli: for $y \in \{0, 1\}$,

$$p(y|x, \theta_j) = \mathrm{Bern}(y|g^{-1}(m_j(x))),$$

where the link function maps $(0, 1)$ to the real line, e.g. logistic, probit. For the *logistic* link function,

$$\mathbb{P}(y = 1|x, \theta_j) = \frac{\exp(m_j(x))}{1 + \exp(m_j(x))}.$$

For the *probit* link function,

$$\mathbb{P}(y = 1|x, \theta_j) = \Phi(m_j(x)),$$

where Φ denotes the standard normal cumulative distribution function. In this case, the model can be equivalently formulated through a latent response $\tilde{y}$ that is Gaussian distributed with mean $m_j(x)$ and unit variance. In particular, $\tilde{y}|m_j(x) \sim \mathrm{N}(m_j(x), 1)$ and

$$p(y|\tilde{y}) = \begin{cases} \mathbf{1}(\tilde{y} \leq 0) & \text{if } l = 0 \\ \mathbf{1}(\tilde{y} > 0) & \text{if } l = 1 \end{cases}.$$

The probit model is recovered by marginalising the latent $\tilde{y}$.

Categorical: for y taking *unordered* values $l = 0, \ldots, L$,

$$p(y|x, \theta_j) = \mathrm{Cat}(y|g^{-1}(m_j(x))),$$

where the link function maps the L-dimensional simplex to $\mathbb{R}^L$. For the multivariate logistic link function,

$$\mathbb{P}(y = l|x, \theta_j) = \frac{\exp(m_{j,l}(x))}{1 + \sum_{l=1}^{L} \exp(m_{j,l}(x))},$$

for $l = 1, \ldots, L$. For the multinomial probit link function,

$$\mathbb{P}(y = l|x, \theta_j) = \mathbb{P}(\tilde{y}_l > \max(\tilde{y}_1, \ldots, \tilde{y}_{l-1}, \tilde{y}_{l+1}, \ldots, \tilde{y}_L, 0)),$$

for $l = 1, \ldots, L$, where $\tilde{y}$ takes values in $\mathbb{R}^L$ and has a multivariate Gaussian distribution with mean $m_j(x) = (m_{j,1}(x), \ldots, m_{j,L}(x))^T$ and covariance matrix Σ_j, which may be the identity matrix, or treated as a more general scale parameter (in this case, care should be taken to avoid identifiability issues). The prior on the vector-valued unknown function $m_j(x)$ can be extended to independent GPs across $l = 1, \ldots, L$ or, more generally, a matrix-variate GP.

Ordinal: for y taking *ordered* values $l = 0, \ldots, L$ and cutoffs $0 = \varepsilon_0 < \varepsilon_1 < \ldots < \varepsilon_{L-1}$,

$$\mathbb{P}(y \leq l|x, \theta_j) = g^{-1}(\varepsilon_l - m_j(x)),$$

where the link function maps $(0, 1)$ to the real line. Due to the nonparametric nature of the model we consider fixed cutoffs $\varepsilon_1, \ldots, \varepsilon_{L-1}$ (Kottas et al., 2005). For the logistic link function,

$$\mathbb{P}(y \leq l|x, \theta_j) = \frac{\exp(\varepsilon_l - m_j(x))}{1 + \exp(\varepsilon_l - m_j(x))}.$$

For the probit link function,

$$\mathbb{P}(y \leq l|x, \theta_j) = \Phi\left(\frac{\varepsilon_l - m_j(x)}{\sigma_j}\right), \tag{1}$$

with additional scale parameter σ_j^2 for $L \geq 2$. In this case, the model can be equivalently formulated through a latent response $\tilde{y}$ that is Gaussian distributed with mean $m_j(x)$ and variance σ_j^2. In particular, $\tilde{y}|m_j(x), \sigma_j^2 \sim \mathrm{N}(m_j(x), \sigma_j^2)$ and

$$p(y|\tilde{y}) = \begin{cases} \mathbf{1}(\tilde{y} \leq 0) & \text{if } l = 0 \\ \mathbf{1}(\varepsilon_{l-1} < \tilde{y} \leq \varepsilon_l) & \text{if } l = 1, \ldots, L-1 \\ \mathbf{1}(\tilde{y} > \varepsilon_{L-1}) & \text{if } l = L \end{cases}.$$

The ordered probit model is recovered by marginalising the latent $\tilde{y}$.

Poisson: for $y \in \{0, 1, 2, \ldots\}$,

$$p(y|x, \theta_j) = \text{Pois}(y|g^{-1}(m_j(x))),$$

where the link function maps $(0, \infty)$ to $\mathbb{R}$. For the log link function with $\lambda_j(x) = \exp(m_j(x))$,

$$\mathbb{P}(y = l|x, \theta_j) = \frac{\exp(-\lambda_j(x))\lambda_j(x)^l}{l!}.$$

Alternatively, a non-negative integer-valued output $y \in \{0, 1, 2, \ldots\}$ can be modelled through a discretised latent Gaussian as in (1), with fixed cutoffs $\epsilon_1 = 1, \epsilon_2 = 2, \ldots$.

2 LOCAL INPUT MODELS

Other types of inputs can be easily handled through the assumptions of local independence

$$p(x|\psi) = \prod_{d=1}^{D} p(x_d|\psi_d),$$

and that each parametric model $p(x_d|\psi_d)$ belongs to the exponential family, that is,

$$p(x_d|\psi_d) = \exp(\psi_d^T t_d(x_d) - a_d(\psi_d) + b_d(x_d)),$$

and t_d, a_d, and b_d are known functions specified by the choice within the exponential family. The standard conjugate prior for ψ assumes independence of ψ_d across $d = 1, \ldots, D$ with

$$\pi(\psi_d) \propto \exp(\psi_d^T \tau_d - \nu_d a_d(\psi_d)).$$

In this conjugate setting, the parameters ψ can be marginalised in each cluster analytically. Specifically, for the collapsed Gibbs sampler, we need 1) the marginal likelihood $h(x_n)$ and 2) the predictive likelihood $h(x_n|\mathbf{X}_{l|j}^{-n})$, where $\mathbf{X}_{l|j}^{-n}$ contains $x_{n'}$ such that $n' \neq n, z_{n'} = (j, l)$. Additionally, for the spilt and merge moves we require the the joint marginal likelihood of $h(\mathbf{X}_{l|j})$. We note that due to the assumption of local independence

$$\begin{aligned}
h(x_n) &= \int p(x_n|\psi)\pi(\psi)d\psi = \prod_{d=1}^{D} h(x_{n,d}),\\
h(x_n|\mathbf{X}_{l|j}^{-n}) &= \int p(x_n|\psi)\pi(\psi|\mathbf{X}_{l|j}^{-n})d\psi\\
&= \prod_{d=1}^{D} h(x_{n,d}|\mathbf{X}_{l|j,d}^{-n}),\\
h(\mathbf{X}_{l|j}) &= \int \prod_{n:z_n=(j,l)} p(x_n|\psi)\pi(\psi)d\psi\\
&= \prod_{d=1}^{D} h(\mathbf{X}_{l|j,d}).
\end{aligned}$$

Examples (used in this paper) include:

Gaussian: for continuous input $x_{n,d}$ taking values in $\mathbb{R}$ with

$$p(x_{n,d}|\psi_d) = \text{N}(x_{n,d}|u_d, s_d^2),$$

where $\psi_d = (u_d, s_d^2)$. The standard conjugate prior is the normal-inverse gamma distribution,

$$u_d|s_d^2 \overset{ind}{\sim} \text{N}(u_{0,d}, c_d^{-1}s_d^2), \quad s_d^2 \overset{ind}{\sim} \text{IG}(a_{x,d}, b_{x,d}),$$

which we denote by $(u_d, s_d^2) \overset{ind}{\sim} \text{NIG}(u_{0,d}, c_d, a_{x,d}, b_{x,d})$. In this case, marginally $x_{n,d}$ has a non-central t-distribution,

$$h(x_{n,d}) = t\left(x_{n,d}|u_{0,d}, \frac{b_{x,d}}{a_{x,d}}\frac{c_d + 1}{c_d}, 2a_{x,d}\right).$$

The predictive distribution of $x_{n,d}$ given $z_n = (j, l)$ is a non-central t-distribution,

$$h(x_{n,d}|\mathbf{X}_{l|j,d}^{-n}) = t\left(x_{n,d}|\widehat{u}_{l|j,d}^{-n}, \frac{\widehat{b}_{x,l|j,d}^{-n}}{\widehat{a}_{x,l|j,d}^{-n}}\frac{\widehat{c}_{l|j,d}^{-n} + 1}{\widehat{c}_{l|j,d}^{-n}}, 2\widehat{a}_{x,l|j,d}^{-n}\right),$$

with $\widehat{c}_{l|j,d}^{-n} = c_d + N_{l|j}^{-n}$, $\widehat{a}_{x,l|j,d}^{-n} = a_{x,d} + N_{l|j}^{-n}/2$,

$$\widehat{u}_{l|j,d}^{-n} = \frac{1}{c_d + N_{l|j}^{-n}}(c_d u_{0,d} + N_{l|j}^{-n}\bar{x}_{l|j,d}^{-n}),$$

$$\begin{aligned}
\widehat{b}_{x,l|j,d}^{-n} = b_{x,d} + \frac{1}{2}\Bigg(&c_d u_{0,d}^2 - \widehat{c}_{l|j,d}^{-n}(\widehat{u}_{l|j,d}^{-n})^2\\
&+ \sum_{n'\neq n:z_{n'}=(j,l)} x_{n',d}^2\Bigg),
\end{aligned}$$

and $\bar{x}_{l|j,d}^{-n} = 1/N_{l|j}^{-n}\sum_{n'\neq n:z_{n'}=(j,l)} x_{n',d}$. The joint marginal likelihood of $\mathbf{X}_{l|j,d}$ follows a multivariate t with mean $u_{0,d}1_{N_{l|j}}$, variance matrix

$$\Sigma_{l|j,d} = \frac{b_{x,d}}{a_{x,d}}\left(I_{N_{l|j}} - \frac{1}{c_d + N_{l|j}}1_{N_{l|j}}1'_{N_{l|j}}\right)^{-1},$$

and degrees of freedom $2a_{x,d}$, that is

$$h(\mathbf{X}_{l|j,d}) = t(\mathbf{X}_{l|j,d}|u_{0,d}1_{N_{l|j}}, \Sigma_{l|j,d}, 2a_{x,d}).$$

Categorical: for discrete inputs $x_{n,d}$ taking *unordered* values $g = 0, 1, \dots, G_d$ with

$$p(x_{n,d}|\psi_d) = \psi_{d,x_{n,d}},$$

where ψ_d is a $G_d + 1$ vector of probabilities such that $\sum_{g=0}^{G_d} \psi_{d,g} = 1$. The standard conjugate prior is the Dirichlet distribution with parameter $\gamma_d = (\gamma_{d,0}, \dots, \gamma_{d,G_d})$. In this case, the marginal likelihood is the Dirichlet-multinomial with

$$h(x_{n,d}) = \frac{\Gamma\left(\sum_{g=0}^{G_d} \gamma_{d,g}\right)}{\Gamma\left(\sum_{g=0}^{G_d} \gamma_{d,g} + 1\right)} \frac{\Gamma\left(\gamma_{d,x_{n,d}} + 1\right)}{\Gamma\left(\gamma_{d,x_{n,d}}\right)}.$$

The predictive likelihood of $x_{n,d}$ given $z_n = (j, l)$ is the Dirichlet-multinomial with

$$h(x_{n,d}|\mathbf{X}_{l|j,d}^{-n}) = \frac{\Gamma\left(\sum_{g=0}^{G_d} \gamma_{d,g} + N_{l|j}^{-n}\right)}{\Gamma\left(\sum_{g=0}^{G_d} \gamma_{d,g} + N_{l|j}^{-n} + 1\right)} \frac{\Gamma\left(\gamma_{d,x_{n,d}} + N_{l|j,x_{n,d}}^{d,-n} + 1\right)}{\Gamma\left(\gamma_{d,x_{n,d}} + N_{l|j,x_{n,d}}^{d,-n}\right)},$$

where $N_{l|j,g}^{d,-n} = \sum_{n' \neq n : z_{n'} = (j,l)} \mathbf{1}(x_{n',d} = g)$. The joint marginal likelihood of $\mathbf{X}_{l|j,d}$ follows a Dirichlet-multinomial with

$$h(\mathbf{X}_{l|j,d}) = \frac{\Gamma\left(\sum_{g=0}^{G_d} \gamma_{d,g}\right)}{\Gamma\left(\sum_{g=0}^{G_d} \gamma_{d,g} + N_{l|j}\right)} \prod_{g=0}^{G_d} \frac{\Gamma\left(\gamma_{d,g} + N_{l|j,g}^{d}\right)}{\Gamma\left(\gamma_{d,g}\right)},$$

and $N_{l|j,g}^{d} = \sum_{n : z_n = (j,l)} \mathbf{1}(x_{n,d} = g)$.

Binomial: for discrete inputs $x_{n,d}$ taking *ordered* values $g = 0, 1, \dots, G_d$ with

$$p(x_{n,d}|\psi_d) = \begin{pmatrix} G_d \\ x_{n,d} \end{pmatrix} \psi_d^{x_{n,d}} (1 - \psi_d)^{G_d - x_{n,d}},$$

where $\psi_d \in (0, 1)$. The standard conjugate prior is the beta distribution with parameter $\gamma_d = (\gamma_{d,0}, \gamma_{d,1})$. In this case, the marginal likelihood is the beta-binomial with

$$h(x_{n,d}) = \begin{pmatrix} G_d \\ x_{n,d} \end{pmatrix} \frac{\Gamma\left(\gamma_{d,0} + \gamma_{p,1}\right)}{\Gamma\left(\gamma_{d,0}\right) \Gamma\left(\gamma_{d,1}\right)} \frac{\Gamma\left(\gamma_{d,0} + x_{n,d}\right) \Gamma\left(\gamma_{d,1} + G_d - x_{n,d}\right)}{\Gamma\left(\gamma_{d,0} + \gamma_{d,1} + G_d\right)}.$$

The predictive likelihood of $x_{n,d}$ given $z_n = (j, l)$ is the beta-binomial with

$$h(x_{n,d}|\mathbf{X}_{l|j,d}^{-n}) = \begin{pmatrix} G_d \\ x_{n,d} \end{pmatrix} \frac{\Gamma\left(\gamma_{d,0} + \gamma_{d,1} + G_d N_{l|j}^{-n}\right)}{\Gamma\left(\widehat{\gamma}_{d,0,l|j}\right) \Gamma\left(\widehat{\gamma}_{d,1,l|j}\right)} \frac{\Gamma\left(\widehat{\gamma}_{d,0,l|j} + x_{n,d}\right) \Gamma\left(\widehat{\gamma}_{d,1,l|j} + G_d - x_{n,d}\right)}{\Gamma\left(\gamma_{d,0} + \gamma_{d,1} + G_d (N_{l|j}^{-n} + 1)\right)},$$

where $\widehat{\gamma}_{d,0,l|j} = \gamma_{d,0} + N_{l|j}^{-n} \bar{x}_{l|j,d}^{-n}$ and $\widehat{\gamma}_{d,1,l|j} = \gamma_{d,1} + N_{l|j}^{-n} (G_d - \bar{x}_{l|j,d}^{-n})$. The joint marginal likelihood of $\mathbf{X}_{l|j,d}$ follows a Beta-binomial with

$$h(\mathbf{X}_{l|j,d}) = \left[\prod_{n : z_n = (j,l)} \begin{pmatrix} G_d \\ x_{n,d} \end{pmatrix} \right] \frac{\Gamma\left(\gamma_{d,0} + \gamma_{d,1}\right)}{\Gamma\left(\gamma_{d,0}\right) \Gamma\left(\gamma_{d,1}\right)} \frac{\Gamma\left(\gamma_{d,0} + N_{l|j} \bar{x}_{l|j,d}\right) \Gamma\left(\gamma_{d,1} + N_{l|j} (G_d - \bar{x}_{l|j,d})\right)}{\Gamma\left(\gamma_{d,0} + \gamma_{d,1} + N_{l|j} G_d\right)}.$$

3 POSTERIOR INFERENCE

We present the algorithm for a general setting, when the observed outputs y_n are a deterministic function of latent Gaussian outputs $\tilde{y}_n$. This includes the probit, ordered probit and multinomial probit, as well as the Gaussian example with $y = \tilde{y}$, among others. The MCMC algorithm targets the posterior

$$\begin{aligned} &\pi(z_{1:N}, \sigma_{1:k}^2, \beta_{0,1:k}, \lambda_{1:k}, \alpha_\theta, \alpha_{\psi,1:k}, \tilde{y}_{1:N} \mid y_{1:N}, x_{1:N}) \propto \\ &\quad \prod_{j=1}^{k} h(\tilde{\mathbf{Y}}_j | \sigma_j^2, \beta_{0,j}, \lambda_j) \prod_{l=1}^{k_j} h(\mathbf{X}_{l|j}) \prod_{n=1}^{N} p(y_n | \tilde{y}_n) \\ &\quad * \frac{\Gamma(\alpha_\theta)}{\Gamma(\alpha_\theta + N)} \alpha_\theta^k \pi(\alpha_\theta) \prod_{j=1}^{k} \alpha_{\psi,j}^{k_j} \frac{\Gamma(\alpha_{\psi,j}) \Gamma(N_j)}{\Gamma(\alpha_{\psi,j} + N_j)} \\ &\quad * \pi(\sigma_j^2) \pi(\beta_{0,j}) \pi(\lambda_j) \pi(\alpha_{\psi,j}) \prod_{l=1}^{k_j} \Gamma(N_{l|j}), \end{aligned}$$

where we make use of the notation $\tilde{\mathbf{Y}}_j$ to denote the latent outputs $\tilde{y}_n$ such that $z_{y,n} = j$ and $\mathbf{X}_{l|j}$ to denote the inputs x_n such that $z_n = (j, l)$. The marginal likelihood of $\tilde{\mathbf{Y}}_j$ given $\beta_{0,j}$, λ_j and σ_j^2, obtained from marginalising the unknown functions m_j, is Gaussian, e.g. for the ordered probit,

$$h(\tilde{\mathbf{Y}}_j | \sigma_j^2, \beta_{0,j}, \lambda_j) = \mathrm{N}(\tilde{\mathbf{Y}}_j \mid \beta_{0,j} 1_{N_j}, \sigma_j^2 I_{N_j} + K_{\lambda_j}),$$

where K_{λ_j} denotes the N_j by N_j matrix of the kernel function evaluated at every pair of inputs in y-cluster j. The marginal likelihood of $\mathbf{X}_{l|j}$, obtained from marginalising $\psi_{l|j}$, is also available in closed form and factorises over D, with examples in Section 2. The term $p(y_n|\tilde{y}_n)$ represents the deterministic function specifying the observed output y_n given the latent Gaussian output $\tilde{y}_n$; examples are provided in Section 1.

The algorithm is a Gibbs sampler, which alternatively samples each set of parameters, 1) the allocation variables $z_{1:N}$, 2) the unique cluster parameters $(\sigma_j^2, \beta_{0,j}, \lambda_j)_{j=1}^{k}$, 3) the concentration parameters α_θ and $\alpha_{\psi,1:k}$ and 4) the latent outputs $\tilde{y}_{1:N}$ (if needed). Computations involving the GP are evaluated using **GPy** in **Python** GPy (since 2012).

Allocation variables. A non-conjugate collapsed Gibbs sampler is employed, combining Algorithm 3, when cluster parameters can be integrated, and Algorithm 8, when cluster parameters cannot be integrated, of Neal (2000), and extending this for the nested partitioning scheme. This consists of N Gibbs steps, where the allocation variable z_n for each data point is updated conditioned on all others $z_1, \ldots, z_{n-1}, z_{n+1}, \ldots, z_N$. This procedure allows local changes to the allocation variables, and to improve mixing in high-dimensional input spaces, we additionally develop two novel split-merge updates for global changes to the nested partition. Throughout, we make use of the superscript notation $-n$ to denote the data points, parameters, and latent variables with the n^{th} data point removed.

The **local updates** are described in the following steps:

1. Remove singleton cluster:
   - Singleton y-cluster: If $z_{y,n} \neq z_{y,n'}$ for all $n' \neq n$, i.e. data point n is in a singleton y-cluster, remove that cluster and set $(\sigma^2_{k^{-n}+1}, \beta_{0,k^{-n}+1}, \lambda_{k^{-n}+1}, \alpha_{\psi,k^{-n}+1})$ equal to the values of the singleton cluster parameters.
   - Singleton x-cluster within a non-singleton y-cluster: If $z_{y,n} = z_{y,n'}$ for some $n' \neq n$ and $z_{x,n} \neq z_{x,n'}$ for all $n' \neq n$ such that $z_{y,n} = z_{y,n'}$, i.e. data point n is in a singleton x-cluster within a non-singleton y-cluster, remove that cluster.
2. Calculate the allocation probability for each occupied cluster: $j \in \{1, \ldots, k^{-n}\}$ and $l \in \{1, \ldots, k_j^{-n}\}$

$$p(z_n = (j,l) | z^{-n}_{1:N}, \ldots) \propto \frac{N_j^{-n} N_{l|j}^{-n}}{\alpha_{\psi,j} + N_j^{-n}} h(\tilde{y}_n | \tilde{\mathbf{Y}}_j^{-n}, \sigma_j^2, \lambda_j, \beta_{0,j}) h(x_n | \mathbf{X}_{l|j}^{-n}).$$

3. Calculate the allocation probability for a new x-cluster within each occupied y-cluster: $j \in \{1, \ldots, k^{-n}\}$

$$p(z_n = (j, k_j^{-n}+1) | z^{-n}_{1:N}, \ldots) \propto \frac{N_j^{-n} \alpha_{\psi,j}}{\alpha_{\psi,j} + N_j^{-n}} h(\tilde{y}_n | \tilde{\mathbf{Y}}_j^{-n}, \sigma_j^2, \lambda_j, \beta_{0,j}) h(x_n).$$

4. Calculate the allocation probability for m new y-clusters: sample m new parameters (or $m-1$ new parameters if $z_{y,n}$ was in a singleton y-cluster) from the prior $(\sigma^2_{k^{-n}+j}, \beta_{0,k^{-n}+j}, \lambda_{k^{-n}+j}, \alpha_{\psi,k^{-n}+j}) \sim \pi(\sigma^2)\pi(\beta_0)\pi(\lambda)\text{Gam}(u_\psi, v_\psi)$. Then, for $j = k^{-n}+1, \ldots, k^{-n}+m$, compute

$$p(z_n = (j,1) | \sigma_j^2, \beta_{0,j}, \lambda_j, \alpha_\theta, \tilde{y}_n, x_n) \propto \frac{\alpha_\theta}{m} h(\tilde{y}_n | \sigma_j^2, \beta_{0,j}, \lambda_j) h(x_n).$$

5. Update the allocation variable z_n using the allocation probabilities. All empty clusters are removed, and if one of the m new clusters is selected, set $z_n = (k^{-n}+1, 1)$ and the parameters $(\sigma^2_{k^{-n}+1}, \beta_{0,k^{-n}+1}, \lambda_{k^{-n}+1}, \alpha_{\psi,k^{-n}+1})$ equal to the parameters of the selected new cluster.

After the full Gibbs sweep for the N allocation variables, two Metropolis-Hastings steps are performed to improve mixing and allow global changes to the allocation variables. The first proposes to move an x-cluster to be nested within a different or new y-cluster and is a 'smarter' version of the move described in Wade et al. (2014), by proposing moves that are more likely to be accepted. This step is separated into three possible moves: 1) an x-cluster, among those within y-clusters with more than one x-cluster, is moved to a different y-cluster; 2) an x-cluster, among those within y-clusters with more than one x-cluster, is moved to a new y-cluster; 3) an x-cluster, among those within y-clusters with only one x-cluster, is moved to a different y-cluster. Define

$$k_{x,2+} = \sum_{j=1}^{k} k_j \mathbf{1}(k_j > 1) \quad \text{and} \quad k_{x,1} = \sum_{j=1}^{k} \mathbf{1}(k_j = 1).$$

At every iteration, Move 1 is performed if $k_{x,2+} > 0$. Next, with probability 1/2, Move 2 is performed, otherwise, Move 3 is performed (with the exception that when $k_{x,1} = 0$, Move 2 is performed with probability 1, or when $k_{x,2+} = 0$, Move 3 is performed with probability 1).

The **global updates** to the **y-clusters** are described in the following steps:

1. **Move 1:** an x-cluster (nested within a y-cluster with more than one x-cluster) is uniformly selected with probability $k_{x,2+}^{-1}$ and moved to be nested within a different y-cluster selected with probability proportional to the conditional marginal likelihood. Specifically, suppose x-cluster l in y-cluster j is first selected, then it is moved to be nested within y-cluster h with probability proportional to $h(\tilde{\mathbf{Y}}_{l|j} | \tilde{\mathbf{Y}}_h, \sigma_h^2, \beta_{0,h}, \lambda_h)$. Let $z^*_{1:N}$ denote the proposed allocations defined by moving x-cluster l in y-cluster j to be nested within y-cluster h for $h \in \{1, \ldots, j-1, j+$

$1, \ldots, k\}$. The acceptance probability is $\min(1, p)$, where

$$p = \frac{\Gamma(N_j - N_{l|j})\Gamma(N_h + N_{l|j})}{\Gamma(N_j)\Gamma(N_h)} * \frac{\Gamma(\alpha_{\psi,j} + N_j)\Gamma(\alpha_{\psi,h} + N_h)}{\Gamma(\alpha_{\psi,j} + N_j - N_{l|j})\Gamma(\alpha_{\psi,h} + N_h + N_{l|j})} * \frac{\alpha_{\psi,h}}{\alpha_{\psi,j}} \frac{k_{x,2+}}{k^*_{x,2+}} \frac{\sum_{h' \neq j} h(\tilde{\mathbf{Y}}_{l|j} | \tilde{\mathbf{Y}}_{h'}, \sigma^2_{h'}, \beta_{0,h'}, \lambda_{h'})}{\sum_{h' \neq h} h(\tilde{\mathbf{Y}}_{l|j} | \tilde{\mathbf{Y}}^*_{h'}, \sigma^2_{h'}, \beta_{0,h'}, \lambda_{h'})},$$

where $\tilde{\mathbf{Y}}^*_{h'}$ contains the outputs under the proposed allocation with $z^*_{y,n} = h'$, e.g. $\tilde{\mathbf{Y}}^*_j$ contains the $N_j - N_{l|j}$ outputs with the $N_{l|j}$ points removed from y-cluster j. The notation $k^*_{x,2+}$ represents the number of x-clusters within a y-cluster with more than one x-cluster under the proposed partition, i.e. $k^*_{x,2+} = k_{x,2+} - \mathbf{1}(k_j = 2) + \mathbf{1}(k_h = 1)$.

2. **Move 2**: an x-cluster (nested within a y-cluster with more than one x-cluster) is uniformly selected with probability $k^{-1}_{x,2+}$ and moved to be nested within a new y-cluster. In this case, we propose new parameters $(\sigma_{k+1}, \beta_{0,k+1}, \lambda_{k+1}, \alpha_{\psi,k+1})$ for the new y-cluster from the prior. The acceptance probability is $\min(1, p)$, where

$$p = \frac{\Gamma(N_j - N_{l|j})\Gamma(N_{l|j})}{\Gamma(N_j)} * \frac{\Gamma(\alpha_{\psi,j} + N_j)\Gamma(\alpha_{\psi,k+1})}{\Gamma(\alpha_{\psi,j} + N_j - N_{l|j})\Gamma(\alpha_{\psi,k+1} + N_{l|j})} * \alpha_\theta \frac{\alpha_{\psi,k+1}}{\alpha_{\psi,j}} \frac{k_{x,2+}}{k_{x,1}} \frac{h(\tilde{\boldsymbol{Y}}^*_{k+1} | \sigma^2_{k+1}, \beta_{0,k+1}, \lambda_{k+1})}{\sum_{h=1}^k h(\tilde{\boldsymbol{Y}}_{l|j} | \tilde{\boldsymbol{Y}}^*_h, \sigma^2_h, \beta_{0,h}, \lambda_h)},$$

where $k^*_{x,1} = k_{x,1} + 1 + \mathbf{1}(k_j = 2)$ represents the number of x-clusters within a y-cluster with only one x-cluster under the proposed partition.

3. **Move 3**: an x-cluster (nested within a y-cluster with only one x-cluster) is uniformly selected with probability $k^{-1}_{x,1}$ and moved to be nested within a different y-cluster selected with probability proportional to the conditional marginal likelihood. Specifically, suppose x-cluster l in y-cluster j is first selected, then it is moved to be nested within y-cluster h with probability proportional to $h(\tilde{\boldsymbol{Y}}_j | \tilde{\boldsymbol{Y}}_h, \sigma^2_h, \beta_{0,h}, \lambda_h)$. Let $z^*_{1:N}$ denote the proposed allocations defined by moving x-cluster l in y-cluster j to be nested within y-cluster h for $h \in \{1, \ldots, j-1, j+1, \ldots, k\}$. The acceptance probability is $\min(1, p)$, where

$$p = \frac{\Gamma(N_h + N_j)}{\Gamma(N_h)\Gamma(N_j)} \frac{\Gamma(\alpha_{\psi,j} + N_j)\Gamma(\alpha_{\psi,h} + N_h)}{\Gamma(\alpha_{\psi,h} + N_h + N_j)\Gamma(\alpha_{\psi,j})} \frac{1}{\alpha_\theta} \frac{\alpha_{\psi,h}}{\alpha_{\psi,j}} * \frac{k_{x,1}}{k^*_{x,2+}} \frac{\sum_{h' \neq j} h(\tilde{\boldsymbol{Y}}_j | \tilde{\boldsymbol{Y}}_{h'}, \sigma^2_{h'}, \beta_{0,h'}, \lambda_{h'})}{h(\tilde{\boldsymbol{Y}}_j | \sigma^2_j, \beta_{0,j}, \lambda_j)},$$

The second set of split-merge updates consists of the pair of 'smart-split' and 'dumb-merge' moves and the pair of 'dumb-split' and 'smart-split' moves, inspired from Wang and Russell (2015), but tailored for the nested clustering structure of the EDP to propose global updates to the x-clusters. In this split moves, one x-cluster is selected and split into two x-clusters, still contained within the same y-cluster. In the merge moves, two x-clusters, within the same y-cluster are merged. The 'smart' moves propose clustering allocations that are more likely and are paired with the corresponding 'dumb' moves, with random cluster allocations, to increase the probability of the reverse move and acceptance of the smart moves. In the first pair of moves, a smart-split or dumb-merge is proposed with probability 1/2, and in the second pair of moves, a dumb-split or smart-merge is proposed with probability 1/2 (unless, there are only singleton x-clusters or only one x-cluster within each y-cluster). Again, we define $k_{x,2+}$ as the number of x-clusters within a y-cluster with more than one x-cluster, i.e. the number x-clusters than may be merged, and additionally define

$$k_{x,1+} = \sum_{j=1}^{k} \sum_{l=1}^{k} \mathbf{1}(N_{l|j} > 1),$$

as the number of x-clusters with more than one data point, i.e. the number x-clusters than may be split.

The **global updates** to the x-clusters are described in the following steps:

- **Smart-Split/Dumb-Merge:** with probability 1/2, one of the following two moves is proposed.

  1. **Smart-Split:** x-cluster l within y-cluster j is selected among the $k_{x,1+}$ x-clusters containing more than one data point with probability proportional to $1/h(\boldsymbol{X}_{l|j})$. The proposed allocation $z^*_{1:N}$ is constructed sequentially by reallocating the data points currently allocated to x-cluster l within y-cluster j, in order of observation, to x-cluster l or a new x-cluster $k_j + 1$ with sequential probabilities

  $$z^*_n | z^*_{1:n-1} = \begin{cases} (j, l) & \text{w.p.} \\ & \propto h(x_n | \boldsymbol{X}^*_{l|j,n-1}) \\ (j, k_j + 1) & \text{w.p.} \\ & \propto h(x_n | \boldsymbol{X}^*_{k_j+1|j,n-1}) \end{cases}$$

  for n such that $z_n = (j, l)$, where $\boldsymbol{X}^*_{l|j,n-1}$ denotes the set of $x_{n'}$ such that $z^*_{n'} = (j, l)$ for $n' < n$, and similarly, $\boldsymbol{X}^*_{k_j+1|j,n-1}$ denotes the set of $x_{n'}$ such that $z^*_{n'} = (j, k_j + 1)$ for $n' < n$. Note that through sequential allocation, there is a positive probability that all points may be allocated to one cluster, and in that case the move is accepted with probability one, i.e. we remain at the current allocation. The probability of proposing the smart-split $z^*_{1:N}$ from $z_{1:N}$ is

$$
\begin{aligned}
&q_{\mathrm{SS}}(z^*_{1:N}|z_{1:N})\\
&=\frac{h(\boldsymbol{X}_{l|j})^{-1}}{\sum_{(j',l'):N_{l'|j'}>1} h(\boldsymbol{X}_{l'|j'})^{-1}}\\
&\qquad\prod_{n:z_n=(j,l)} \frac{2h(x_n|\boldsymbol{X}^*_{z^*_{x,n}|j,n-1})}{h(x_n|\boldsymbol{X}^*_{l|j,n-1})+h(x_n|\boldsymbol{X}^*_{k_j+1|j,n-1})},
\end{aligned}
$$

where the factor of 2 is is needed as the proposed state $z^*_{1:N}$ is equivalent when the labels l and k_j+1 are interchanged. The acceptance probability is $\min(1,p)$, where

$$
\begin{aligned}
p =&\frac{\alpha_j\Gamma(N^*_{l|j})\Gamma(N^*_{k_j+1|j})h(\boldsymbol{X}^*_{l|j})h(\boldsymbol{X}^*_{k_j+1|j})}{\Gamma(N_{l|j})h(\boldsymbol{X}_{l|j})}\\
&\frac{1}{k^*_{x,2+}k_j}\frac{\sum_{(j',l'):N_{l'|j'}>1} h(\boldsymbol{X}_{l'|j'})^{-1}}{h(\boldsymbol{X}_{l|j})^{-1}}*\\
&\prod_{n:z_n=(j,l)} \frac{h(x_n|\boldsymbol{X}^*_{l|j,n-1})+h(x_n|\boldsymbol{X}^*_{k_j+1|j,n-1})}{h(x_n|\boldsymbol{X}^*_{z^*_{x,n}|j,n-1})},
\end{aligned}
$$

where $k^*_{x,2+}$ is equal to $k_{x,2+}+2$ if x-cluster l was the only cluster within y-cluster j, i.e. $k_j=1$, and $k^*_{x,2+}$ is equal to $k_{x,2+}+1$ otherwise.

2. **Dumb-Merge:** x-cluster l within y-cluster j is selected uniformly among the $k_{x,2+}$ x-clusters contained within a y-cluster with more than one x-cluster with probability $1/k_{x,2+}$ and a second x-cluster $l'\neq l$ within y-cluster j is selected uniformly among the k_j-1 remaining x-clusters within y-cluster j with probability $1/(k_j-1)$. The probability of proposing the dumb-merge $z^*_{1:N}$ from $z_{1:N}$ is

$$
q_{\mathrm{DM}}(z^*_{1:N}|z_{1:N}) = \frac{2}{k_{x,2+}(k_j-1)},
$$

where the factor of 2 is needed as the proposed state $z^*_{1:N}$ can also be reached by first selecting x-cluster l' within y-cluster j and then selecting x-cluster l within y-cluster j. The acceptance probability is $\min(1,p)$, where

$$
\begin{aligned}
p =&\frac{\Gamma(N^*_{l|j})h(\boldsymbol{X}^*_{l|j})}{\alpha_j\Gamma(N_{l|j})\Gamma(N_{l'|j})h(\boldsymbol{X}_{l|j})h(\boldsymbol{X}_{l'|j})}\\
&k_{x,2+}(k_j-1)\frac{h(\boldsymbol{X}^*_{l|j})^{-1}}{\sum_{(j',l'):N^*_{l'|j'}>0} h(\boldsymbol{X}^*_{l'|j'})^{-1}}*\\
&\prod_{n:z^*_n=(j,l)} \frac{h(x_n|\boldsymbol{X}_{z_{x,n}|j,n-1})}{h(x_n|\boldsymbol{X}_{l|j,n-1})+h(x_n|\boldsymbol{X}_{l'|j,n-1})}.
\end{aligned}
$$

- **Dumb-Split/Smart-Merge:** with probability 1/2, one of the following two moves is proposed.

1. **Dumb-Split:** x-cluster l within y-cluster j is uniformly selected among the $k_{x,1+}$ x-clusters containing more than one data point with probability $1/k_{x,1+}$. The data points in x-cluster l within y-cluster j are then randomly reallocated to x-cluster l or a new x-cluster k_j+1 with probability 1/2. Note that again there is a positive probability that all points may be allocated to one cluster, and in that case the move is accepted with probability one, i.e. we remain at the current allocation. The probability of proposing the dumb-split $z^*_{1:N}$ from $z_{1:N}$ is

$$
q_{\mathrm{DS}}(z^*_{1:N}|z_{1:N}) = \frac{1}{k_{x,1+}}\frac{2}{2^{N_{l|j}}} = \frac{1}{k_{x,1+}}\frac{1}{2^{N_{l|j}-1}},
$$

where the factor of 2 is is needed as the proposed state $z^*_{1:N}$ is equivalent when the labels l and k_j+1 are interchanged. The acceptance probability is $\min(1,p)$, where

$$
\begin{aligned}
p =&\frac{\alpha_j\Gamma(N^*_{l|j})\Gamma(N^*_{k_j+1|j})h(\boldsymbol{X}^*_{l|j})h(\boldsymbol{X}^*_{k_j+1|j})}{\Gamma(N_{l|j})h(\boldsymbol{X}_{l|j})}\\
&k_{x,1+}2^{N_{l|j}-1}\frac{1}{k^*_{x,2+}}*\left(\frac{h(\boldsymbol{X}_{l|j})}{\sum_{h\neq l} h(\boldsymbol{X}^*_{(l,h)|j})}\right.\\
&\left.+\frac{h(\boldsymbol{X}_{l|j})}{\sum_{h\neq k_j+1} h(\boldsymbol{X}^*_{(k_j+1,h)|j})}\right).
\end{aligned}
$$

2. **Smart-Merge:** x-cluster l within y-cluster j is selected uniformly among the $k_{x,2+}$ x-clusters contained within a y-cluster with more than one x-cluster with probability $1/k_{x,2+}$. A second x-cluster $l'\neq l$ within y-cluster j is selected among the k_j-1 remaining x-clusters within y-cluster j with probability proportional to $h(\boldsymbol{X}_{(l,l')|j})$, where $\boldsymbol{X}_{(l,l')|j}$ denotes the set of x_n under the merger of x-clusters l and l'. The probability of proposing the smart-merge $z^*_{1:N}$ from $z_{1:N}$ is

$$
\begin{aligned}
q_{\mathrm{SM}}(z^*_{1:N}|z_{1:N}) =&\frac{1}{k_{x,2+}}\left(\frac{h(\boldsymbol{X}_{(l,l')|j})}{\sum_{h\neq l} h(\boldsymbol{X}_{(l,h)|j})}\right.\\
&\left.+\frac{h(\boldsymbol{X}_{(l,l')|j})}{\sum_{h\neq l'} h(\boldsymbol{X}_{(l',h)|j})}\right),
\end{aligned}
$$

which is the sum of the probability of first selecting l and then l' and vice versa, as the proposed state $z^*_{1:N}$ is equivalent under these proposals. The acceptance probability is $\min(1,p)$, where

$$p = \frac{\Gamma(N^*_{l|j})h(\boldsymbol{X}^*_{l|j})}{\alpha_j\Gamma(N_{l|j})\Gamma(N_{l'|j})h(\boldsymbol{X}_{l|j})h(\boldsymbol{X}_{l'|j})} \frac{1}{k^*_{x,1+}}\frac{1}{2^{N^*_{l|j}-1}}k_{x,2+} * \left(\frac{h(\boldsymbol{X}^*_{l|j})}{\sum_{h\neq l} h(\boldsymbol{X}_{(l,h)|j})} + \frac{h(\boldsymbol{X}^*_{l|j})}{\sum_{h\neq l'} h(\boldsymbol{X}_{(l',h)|j})}\right)^{-1},$$

where $k^*_{x,1+}$ is equal to $k_{x,1+}+1$ if two singleton clusters are merged; $k_{x,1+}$ if one of merged clusters is a singleton; and $k_{x,1+}-1$ if neither cluster is a singleton.

Cluster parameters. The parameters for each cluster are conditionally independent across $j=1,\dots,k$ with full conditional

$$\pi(\sigma_j^2, \beta_{0,j}, \lambda_j|\tilde{\mathbf{Y}}_j) \propto h(\tilde{\mathbf{Y}}_j|\sigma_j^2, \beta_{0,j}, \lambda_j)\pi(\sigma_j^2)\pi(\beta_{0,j})\pi(\lambda_j),$$

which is not available in closed form. We use Hamiltonian Monte Carlo (Duane et al., 1987) to sample from the full conditional.

Mass parameters. The concentration parameters α_θ and $\alpha_{\psi,1:k}$ are updated using the auxiliary variable technique of Escobar and West (1995). For α_θ, sample an auxiliary variable $\xi \sim \text{Beta}(\alpha_\theta+1, N)$; set $\widehat{v}_\theta = v_\theta - \log(\xi)$ and

$$\widehat{u}_\theta = \begin{cases} u_\theta + k - 1 & \text{w.p. } \frac{N\widehat{v}_\theta}{N\widehat{v}_\theta + u_\theta + k - 1} \\ u_\theta + k & \text{w.p. } \frac{u_\theta + k - 1}{N\widehat{v}_\theta + u_\theta + k - 1} \end{cases};$$

and sample $\alpha \sim \text{Gam}(\widehat{u}_\theta, \widehat{v}_\theta)$. Similarly, for $\alpha_{\psi,j}$, for $j=1,\dots,k$, sample an auxiliary variable $\xi_j \sim \text{Beta}(\alpha_{\psi,j}+1, N_j)$; set $\widehat{v}_{\psi,j} = v_\psi - \log(\xi_j)$ and

$$\widehat{u}_{\psi,j} = \begin{cases} u_\psi + k_j - 1 & \text{w.p. } \frac{N_j\widehat{v}_{\psi,j}}{N_j\widehat{v}_{\psi,j} + u_\psi + k_j - 1} \\ u_\psi + k_j & \text{w.p. } \frac{u_\psi + k_j - 1}{N_j\widehat{v}_{\psi,j} + u_\psi + k_j - 1} \end{cases};$$

and sample $\alpha_{\psi,j} \sim \text{Gam}(\widehat{u}_{\psi,j}, \widehat{v}_{\psi,j})$.

Latent outputs. The latent outputs are independent across cluster $j=1,\dots,k$, with full conditional

$$\pi(\tilde{\mathbf{Y}}_j|\mathbf{Y}_j, \sigma_j^2, \beta_{0,j}, \lambda_j) \propto h(\tilde{\mathbf{Y}}_j|\sigma_j^2, \beta_{0,j}, \lambda_j) \prod_{n:z_n=j} p(y_n|\tilde{y}_n).$$

In the Gaussian case, $p(y_n|\tilde{y}_n) = \mathbf{1}(y_n = \tilde{y}_n)$, and this step is not needed. For the other probit-type models, the full conditional of the latent outputs in cluster j is a truncated multivariate Gaussian, which is sampled through a Gibbs algorithm combined with cumulative distribution function inversion techniques (Kotecha and Djuric, 1999).

4 PREDICTIONS

Letting $\zeta = (z_{1:N}, \sigma^2_{1:k}, \beta_{0,1:k}, \lambda_{1:k}, \alpha_\theta, \alpha_{\psi,1:k}, \tilde{y}_{1:N})$ denote the model parameters and latent variables, the MCMC algorithm provides samples $\zeta^{(m)}$, for $m=1,\dots,M$, from the posterior. In the Gaussian example, the posterior density at y_* given a new x_* is given by

$$\begin{aligned} &f(y_*|x_*, y_{1:N}, x_{1:N}) \\ &= \int f(y_*|x_*, y_{1:N}, x_{1:N}, \zeta)\frac{\pi(\zeta|y_{1:N}, x_{1:N})f(x_*|x_{1:N},\zeta)}{f(x_*|x_{1:N})}d\zeta \\ &\approx \frac{1}{f(x_*|x_{1:N})}\sum_{m=1}^M f(y_*|x_*, y_{1:N}, x_{1:N}, \zeta^{(m)})f(x_*|x_{1:N}, \zeta^{(m)}) \\ &\approx C^{-1}\left(\sum_{m=1}^M p^{(m)}_{k^{(m)}+1}(x_*)h(y_*) + \sum_{j=1}^{k^{(m)}} p^{(m)}_j(x_*)h(y_*|\mathbf{Y}^{(m)}_j, \beta^{(m)}_{0,j}, \lambda^{(m)}_j, \sigma_j^{2\,(m)})\right), \end{aligned}$$

with

$$f(x_*|x_{1:N}) \approx C := \sum_{m=1}^M p^{(m)}_{k^{(m)}+1}(x_*) + \sum_{j=1}^{k^{(m)}} p^{(m)}_j(x_*).$$

In this case, we have a weighted average of the GP predictive densities across clusters and the marginal likelihood $h(y_*)$ for a new cluster. Note that the marginal likelihood $h(y_*)$ for a new cluster is unavailable in closed form as it requires integration over the parameters $(\beta_0, \lambda, \sigma^2)$. However, we can compute a simple Monte Carlo estimate of this quantity by sampling from the prior,

$$h(y_*) \approx \frac{1}{S}\sum_{s=1}^S \text{N}(y_* \mid \beta_0^s, \sigma^{2\,s} + K_{\lambda^s}(x_*, x_*)),$$

with $(\sigma^{2\,s}, \beta_0^s, \lambda^s)$ i.i.d. samples from the prior.

For other types of outputs through probit models, we can similarly use the MCMC output to compute predictive quantities of interest at a test input x_*. For example, considering the ordered probit with ordered categories $l=0,\dots,L$ and fixed cutoffs $0=\varepsilon_0<\varepsilon_1<\dots<\varepsilon_{L-1}$, we first note that we can compute the expectation and density of the latent continuous $\tilde{y}_*$ given the test input x_*, as in the Gaussian example.

The posterior probability that $y_* = l$ given the test input x_* is

$$\begin{aligned}
&\mathbb{P}(y_* = l|x_*, y_{1:N}, x_{1:N})\\
=&\int \mathbb{P}(y_* = l|x_*, y_{1:N}, x_{1:N}, \zeta)\\
&\quad \frac{\pi(\zeta|y_{1:N}, x_{1:N}) f(x_*|x_{1:N}, \zeta)}{f(x_*|x_{1:N})} d\zeta\\
\approx& C^{-1}\left(\sum_{m=1}^{M} p^{(m)}_{k^{(m)}+1}(x_*)\mathbb{P}(y_* = l|x_*)\right.\\
&\left. + \sum_{j=1}^{k^{(m)}} p_j^{(m)}(x_*)\mathbb{P}(y_* = l|x_*, \tilde{\mathbf{Y}}_j^{(m)}, \sigma_j^{2\,(m)}, \beta_{0,j}^{(m)}, \lambda_j^{(m)})\right).
\end{aligned}$$

For cluster j of sample m, the probability that $y_* = l$ is

$$\begin{aligned}
&\mathbb{P}(y_* = l|x_*, \tilde{\mathbf{Y}}_j^{(m)}, \sigma_j^{2\,(m)}, \beta_{0,j}^{(m)}, \lambda_j^{(m)})\\
=&\mathbb{P}(\varepsilon_{l-1} < \tilde{y}_* \leq \varepsilon_l|x_*, \tilde{\mathbf{Y}}_j^{(m)}, \sigma_j^{2\,(m)}, \beta_{0,j}^{(m)}, \lambda_j^{(m)})\\
=&\Phi\left(\frac{\varepsilon_l - \widehat{m}_j^{(m)}(x_*)}{\sqrt{\widehat{K}_j^{(m)}(x_*, x_*) + \sigma_j^{2\,(m)}}}\right) -\\
&\Phi\left(\frac{\varepsilon_{l-1} - \widehat{m}_j^{(m)}(x_*)}{\sqrt{\widehat{K}_j^{(m)}(x_*, x_*) + \sigma_j^{2\,(m)}}}\right),
\end{aligned}$$

with $\varepsilon_{-1} = -\infty$, $\varepsilon_L = \infty$ and $\widehat{m}_j^{(m)}(x_*)$ and $\widehat{K}_j^{(m)}(x_*, x_*)$ denoting the GP predictive mean and kernel functions in cluster j of sample m. For a new cluster, the marginal probability $\mathbb{P}(y_* = l|x_*)$ is unavailable in closed form as it requires integration over the parameters $(\beta_0, \lambda, \sigma^2)$. We can again employ a Monte Carlo approach to estimate this quantity,

$$\begin{aligned}
\mathbb{P}(y_* = l|x_*) \approx& \frac{1}{S}\sum_{s=1}^{S}\Phi\left(\frac{\varepsilon_l - \beta_0^s}{\sqrt{K_{\lambda^s}(x_*, x_*)) + \sigma^{2\,s}}}\right) -\\
&\Phi\left(\frac{\varepsilon_{l-1} - \beta_0^s}{\sqrt{K_{\lambda^s}(x_*, x_*)) + \sigma^{2\,s}}}\right),
\end{aligned}$$

with $(\sigma^{2\,s}, \beta_0^s, \lambda^s)$ i.i.d. samples from the prior.

An advantage of jointly modelling the outputs and inputs includes the possibility to compute the predictive distribution of y_* based only on a subset of inputs, say only based on a single input x_{*d}. In this case, the weights would only involve the local predictive marginal likelihood of x_{*d} for each cluster $h(x_{*d}|\mathbf{X}^{(m)}_{l|j,d})$, $j = 1, \dots, k^{(m)}$ and $l = 1, \dots, k_j^{(m)}$, and for a new cluster $h(x_{*d})$. However, the local expectation would need to be integrated with respect to predictive marginal likelihood of $x_{*-d} = (x_{*1}, \dots, x_{*d-1}, x_{*d+1}, \dots, x_{*D})$ in each nested clustering. For example, in the Gaussian case,

$$\begin{aligned}
&\mathbb{E}[y_*|x_{*d}, y_{1:N}, x_{1:N}] \approx\\
&C_d^{-1}\left(\sum_{m=1}^{M} p^{(m)}_{k^{(m)}+1}(x_{*d})\mu_\beta + \sum_{j=1}^{k^{(m)}} p^{(m)}_{j,1}(x_{*d})\mathbb{E}_{x_{*-d}}[\widehat{m}_j^{(m)}(x_*)]\right.\\
&\left. + \sum_{j=1}^{k^{(m)}}\sum_{l=1}^{k_j^{(m)}} p^{(m)}_{j,l}(x_{*d})\mathbb{E}_{x_{*-d}}[\widehat{m}_j^{(m)}(x_*)|\mathbf{X}^{(m)}_{l|j,-d}]\right),
\end{aligned}$$

where expectations are taken with respect to $h(x_{*-d})$ and $h(x_{*-d}|\mathbf{X}^{(m)}_{l|j,-d})$, i.e.

$$\begin{aligned}
&\mathbb{E}_{x_{*-d}}[\widehat{m}_j^{(m)}(x_*)] =\\
&\qquad \int \widehat{m}_j^{(m)}(x_*)\prod_{d' \neq d} h(x_{*d'})dx_{*-d},\\
&\mathbb{E}_{x_{*-d}}[\widehat{m}_j^{(m)}(x_*)|\mathbf{X}^{(m)}_{l|j,-d}]\\
&\qquad = \int \widehat{m}_j^{(m)}(x_*)\prod_{d' \neq d} h(x_{*d'}|\mathbf{X}^{(m)}_{l|j,d'})dx_{*-d},
\end{aligned}$$

with

$$\begin{aligned}
p^{(m)}_{k^{(m)}+1}(x_{*d}) =& \frac{\alpha_\theta^{(m)}}{\alpha_\theta^{(m)} + N}h(x_{*d});\\
p^{(m)}_{j,1}(x_{*d}) =& \frac{N_j^{(m)}}{\alpha_\theta^{(m)} + N}\frac{\alpha^{(m)}_{\psi,j}}{\alpha^{(m)}_{\psi,j} + N_j^{(m)}}h(x_{*d})\\
p^{(m)}_{j,l}(x_{*d}) =& \frac{N_j^{(m)}}{\alpha_\theta^{(m)} + N}\frac{N^{(m)}_{l|j}}{\alpha^{(m)}_{\psi,j} + N_j^{(m)}}h(x_{*d}|\mathbf{X}^{(m)}_{l|j,d}),
\end{aligned}$$

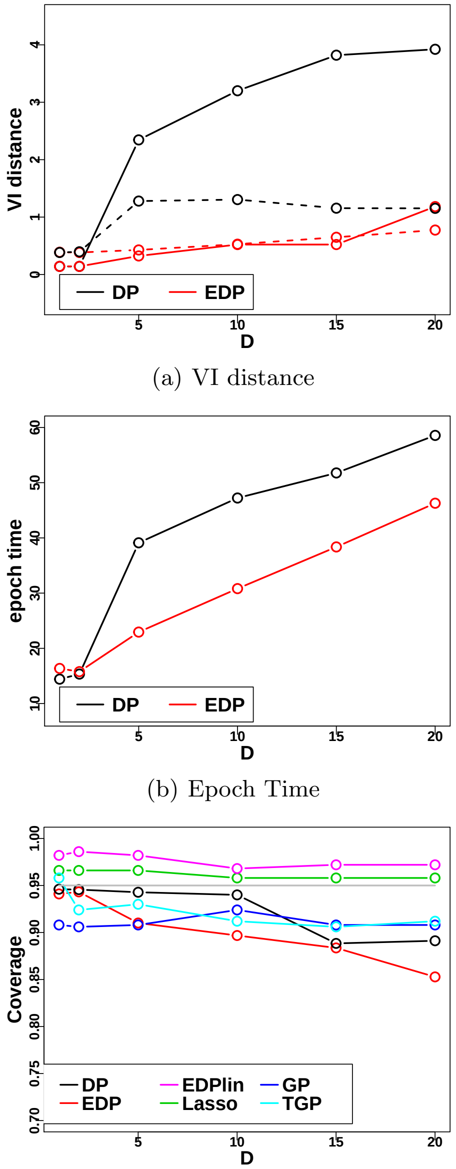
with $C_d = \sum_{m=1}^{M} p^{(m)}_{k^{(m)}+1}(x_{*d}) + \sum_{j=1}^{k^{(m)}} p^{(m)}_{j,1}(x_{*d}) + \sum_{j=1}^{k^{(m)}}\sum_{l=1}^{k_j^{(m)}} p^{(m)}_{j,l}(x_{*d})$.

5 EXAMPLES

In the following subsections, we provide further details and insights on the results of the corresponding examples presented in the manuscript.

5.1 Simulated Mixture of Damped Cosine Functions

In the main text we consider a simulated example in which points are generated from a highly non-linear regression obtained as a mixture of two damped cosines (Santner et al., 2003). This is coupled with a distribution over the inputs, which are independently sampled from a multivariate normal. In the following sections we present additional supplementary details



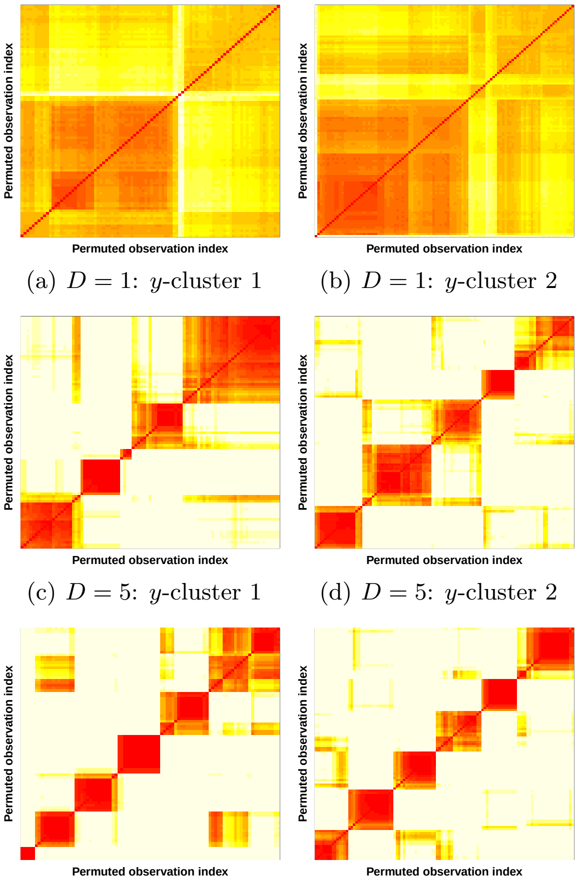
(a) VI distance

(b) Epoch Time

(c) Coverage of CI

Figure 1: Simulated ARD example. Comparison of the EDP MoE with the DP MoE, Lasso, GP, and TGP in terms of the VI distance between the true and estimated clustering (with dashed lines for the size of the credible ball), average epoch time (in seconds), and the coverage of the 95% credible intervals (CI).

on this example, where only the first input is a predictor in the damped cosine functions and an automatic relevance determination (ARD) kernel is employed to recover the sparse structure. Following this we present an alternative isotropic kernel example, in which the damped cosine functions used the input mean as the predictor. This additional example demonstrates the scalability of our method to larger dimension D.



(a) $D = 1$: y-cluster 1 (b) $D = 1$: y-cluster 2

(c) $D = 5$: y-cluster 1 (d) $D = 5$: y-cluster 2

(e) $D = 10$: y-cluster 1 (f) $D = 10$: y-cluster 2

Figure 2: Simulated ARD example. Heat map of the posterior similarity matrix for the x-clustering within the two estimated y-clusters for the enriched MoE. Rows correspond to increasing $D = 1, 5, 10$, whilst columns correspond to y-cluster. To improve visualisation, observations are permuted based on hierarchical clustering.

Automatic relevance determination kernel

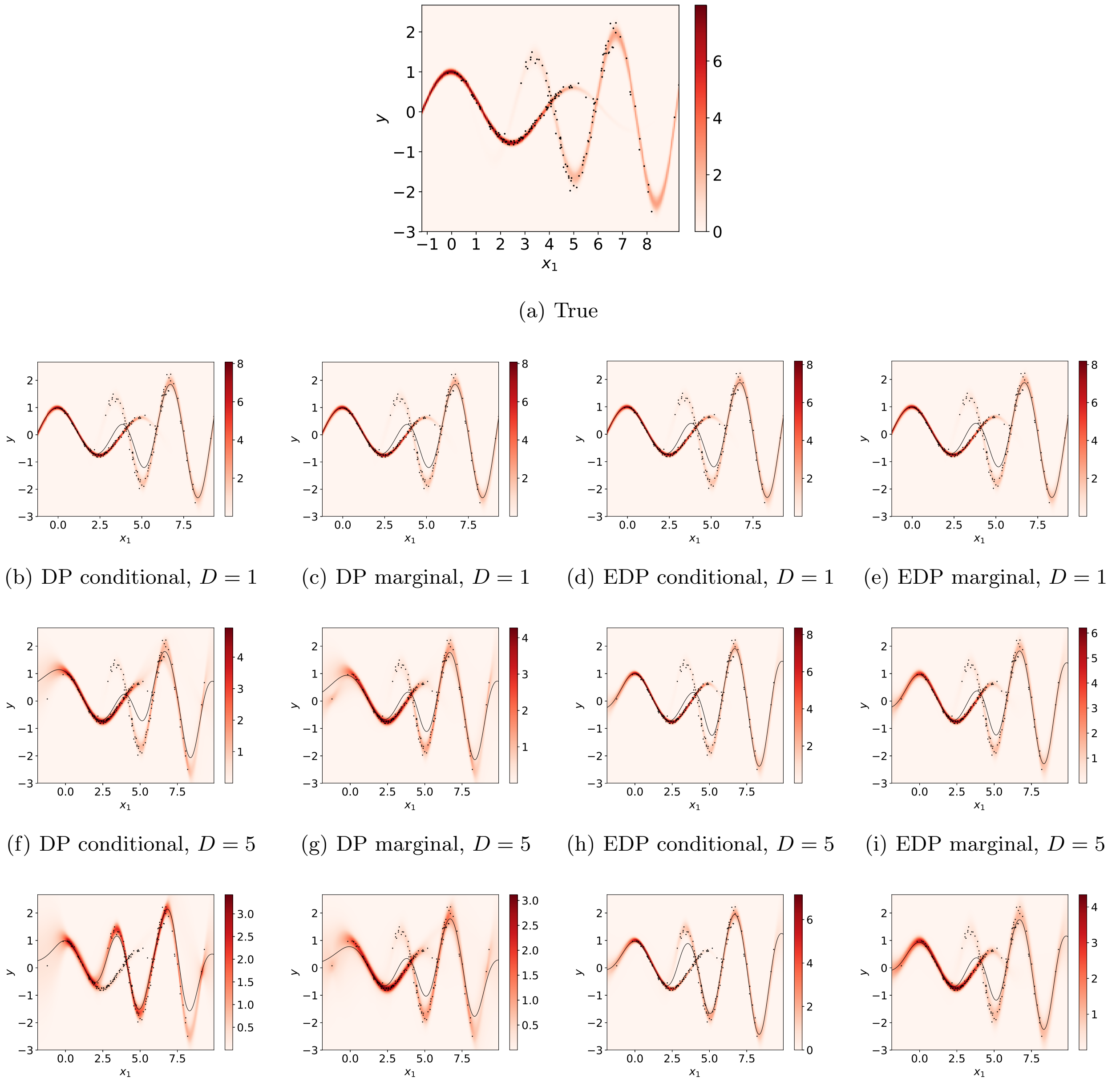
In the first example of the article, a data set of 200 points is generated from a mixture of two damped cosine functions by:

$$\begin{aligned} y_n|x_n \overset{ind}{\sim} & \; p\left(x_{n,1}\right) \mathrm{N}\left(\exp\left\{\beta_{1,0}x_{n,1}\right\}\cos\left(\beta_{1,1}\pi x_{n,1}\right), \sigma_1^2\right) \\ & + \left(1-p\left(x_{n,1}\right)\right)\mathrm{N}\left(\exp\left\{\beta_{2,0}x_{n,1}\right\}\cos\left(\beta_{2,1}\pi x_{n,1}\right), \sigma_2^2\right), \end{aligned} \tag{2}$$

with mixture weights, $p\left(x_{n,1}\right)$, equal to

$$\frac{\tau_1 \exp\left\{-\frac{\tau_1}{2}\left(x_{n,1}-\mu_1\right)^2\right\}}{\tau_1 \exp\left\{-\frac{\tau_1}{2}\left(x_{n,1}-\mu_1\right)^2\right\} + \tau_2 \exp\left\{-\frac{\tau_2}{2}\left(x_{n,1}-\mu_2\right)^2\right\}}.$$

The damped cosines are parametrised by $\beta_1 = (0.1, 0.6)'$, $\beta_2 = (-0.1, 0.4)'$ with $\sigma_1 = 0.15$, $\sigma_2 = 0.05$.



(a) True

(b) DP conditional, $D = 1$

(c) DP marginal, $D = 1$

(d) EDP conditional, $D = 1$

(e) EDP marginal, $D = 1$

(f) DP conditional, $D = 5$

(g) DP marginal, $D = 5$

(h) EDP conditional, $D = 5$

(i) EDP marginal, $D = 5$

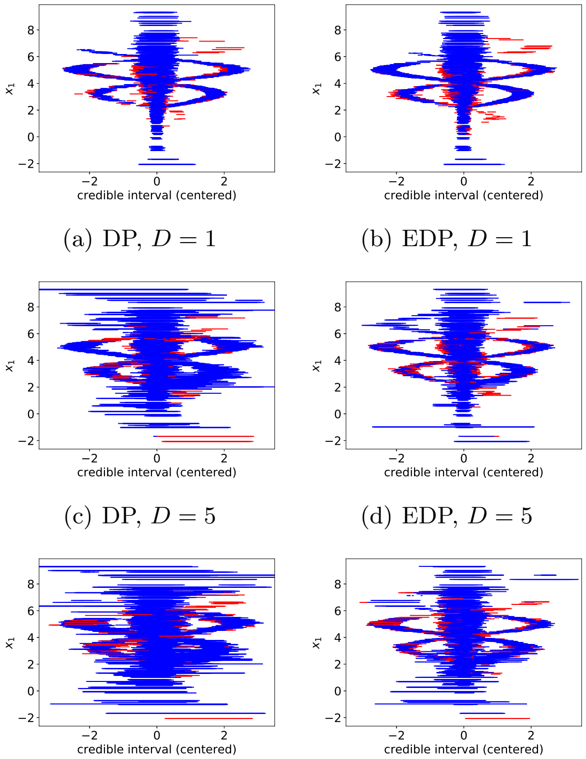
(j) DP conditional, $D = 10$

(k) DP marginal, $D = 10$

(l) EDP conditional, $D = 10$

(m) EDP marginal, $D = 10$

Figure 3: Simulated ARD example. Top row: the true data generating predictive density plot. Bottom three rows: the predictive mean (black line) and predictive density (red) plots for the DP (first two columns) and EDP (last two columns) for a grid of $x_{*,1}$ values, with additional inputs conditioned on their sample means (first and third columns) or marginalised (second and last column), with increasing $D = 1, 5, 10$ (rows).



(a) DP, $D = 1$

(b) EDP, $D = 1$

(c) DP, $D = 5$

(d) EDP, $D = 5$

(e) DP, $D = 10$

(f) EDP, $D = 10$

Figure 4: Simulated ARD example. Coverage for the DP and EDP mixture of experts with increasing $D = 1, 5, 10$. Each horizontal line depicts the 95% credible interval based on the unions of highest posterior density. This line is blue if the sampled truth lies inside the interval, and red otherwise. The percentage of samples lying inside the interval is the empirical coverage.

The mixture model is parametrised by $\tau_1 = \tau_2 = 0.8$, $\mu_1 = 5$ and $\mu_2 = 3$. The inputs are independently sampled from a multivariate normal $x_n \sim \mathrm{N}(\mu, \Sigma)$, centred at $\mu = (4, \ldots, 4)$, with standard deviation of 2 along each dimension, that is $\Sigma_{h,h} = 4$. The covariance matrix Σ assumes with the additional inputs positively correlated among each other, with $\Sigma_{h,l} = 3.5$ for $h \neq l$, $h > 1$ and $l > 1$, but independent of the first input, with $\Sigma_{1,l} = 0$ for $l > 1$.

For both the DP and EDP mixtures of GP experts, we employ the same prior choices, based on identified reasonable ranges for the parameters. For the ARD squared exponential kernels of the GPs, we utilise a Gamma(3, 1) prior on the first input dimension length-scale, Gamma(10, 1/2) prior on the other input dimension length-scales and a Gamma(2, 1.5) prior on the magnitude. The constant means β_0 of the GPs have a $\mathrm{N}(0, 0.5^2)$ prior. The variance σ_y^2 has a log-$\mathrm{N}(\log(0.01), 0.5^2)$. For the DP, the mass parameter has hyper-parameters $(u_a = 1, v_a = 1)$, and for the EDP, the mass parameters have hyper-parameters $(u_\theta = 1, v_\theta = 1)$ and $(u_\psi = 1, v_\psi = 1)$. A Gaussian input model is used with hyperparameters of the conjugate normal-inverse gamma set to $u_{0,d} = \bar{x}_d$, $c_d = 1/4$, $b_{x,d} = 1$, and $a_{x,d} = 2$.

Table 1: Simulated ARD example. The number of clusters in the VI estimated x-clustering within the two estimated y-clusters for the enriched model, as D increases.

y-cluster	D				
	1	5	10	15	20
1	1	5	6	7	7
2	1	6	7	8	7

Posterior inference for both models is performed with 5000 total iterations and a burn-in of 1000. Average epoch times (in seconds) after burn-in are reported in Figure 1(b). When fewer experts are identified through the nested clustering of the EDP (e.g. $D > 2$ in our example), average epoch time is reduced for the EDP compared with the DP. Each run was performed independently and in parallel using the high performance computing resources provided by *commented for blind review.*

The VI distance between the true and estimated (y-level) clustering is depicted in Figure 1(a), with dashed lines representing the size of the 95% VI credible balls. For the DP, the distance increases greatly with D, and the true clustering is far from the credible ball. The behaviour of the y-level clustering of the EDP is more robust to increasing D, Figure 1(b), while the x-level clustering requires an increasing number of clusters. Figure 2 depicts the heat map of the posterior similarity matrix for the x-clustering within the two estimated y-clusters, and Table 1 reports the number of x-clusters in the VI estimated x-clustering within the two estimated y-clusters.

We plot the estimates for the predictive response density and mean against the first input over a dense grid. These are presented in Figure 3, for different choices of D. In the first and third columns the additional inputs are fixed to their sample means (approximately 4) for the DP and EDP models, respectively. Further, in the second and fourth columns the additional inputs are marginalised.

Finally, coverage plots are presented in Figure 4. Centered around the true values (sampled from the data generating distribution of equation (2)), these plots show the 95% highest posterior density credible intervals for randomly sampled inputs (in some cases this may be a union of intervals). When the sample of the truth lies within our credible interval the line is blue, otherwise it is red. The increasing uncertainty of the DP for increasing D is clearly visible from Figure 4, while the EDP retains smaller credible intervals, with similar coverage. Figure 1(c) summarises the coverage across the competing models. All GP-based methods show a decrease in coverage with increasing D. In order to cope with the additional noisy inputs, length-scale priors with heavier tails may be required to effectively identify the relevant inputs.

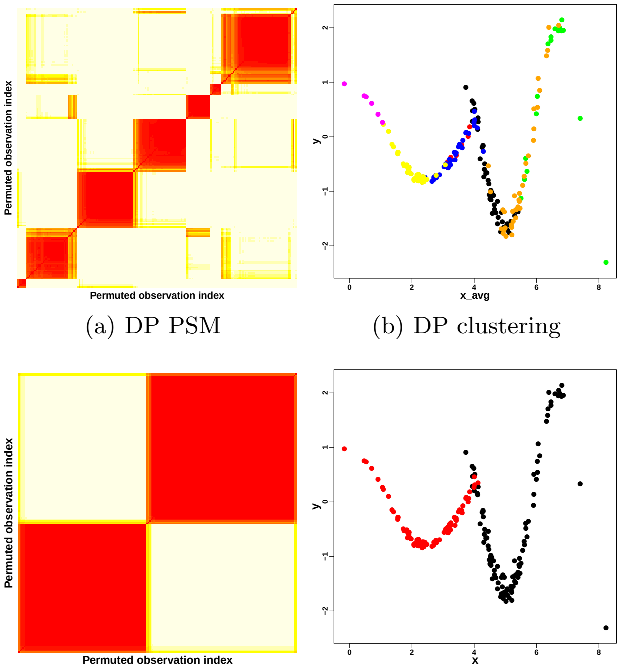
5.1.1 Isotropic kernel

In this second example, $N = 200$ points are again generated from a highly non-linear regression obtained as a mixture of two damped cosines (2) (Santner et al., 2003). However, in this case the mixture weights and damped cosine functions depend on the inputs through their average across dimensions. These inputs are independently sampled from a positively-correlated multivariate normal, and for all GP experts an isotropic squared exponential kernel is used. This kernel favours scalability of the GP experts with respect to D.

The damped cosines are now parametrised by $\beta_1 = (0.1, 0.6)'$, $\beta_2 = (-0.1, 0.4)'$ with $\sigma_1 = 0.15$, $\sigma_2 = 0.05$. The mixture model is parametrised by $\tau_1 = \tau_2 = 1.2$, $\mu_1 = 2$ and $\mu_2 = 6$. The inputs are independently sampled from a multivariate normal $x_n \sim \mathrm{N}(\mu, \Sigma)$, centred at $\mu = (4, \ldots, 4)$, with standard deviation of 2 along each dimension, that is $\Sigma_{h,h} = 4$. The covariance matrix Σ assumes with the additional inputs positively correlated among each other, with $\Sigma_{h,l} = 3.5$ for $h \neq l$, $h > 1$ and $l > 1$, but independent of the first input, with $\Sigma_{1,l} = 0$ for $l > 1$.

For both the DP and EDP mixtures of GP experts we employ the same prior choices, based on identified reasonable ranges for the parameters which change empirically with D. For the isotropic squared exponential kernels of the GPs, we utilise a $\mathrm{Gamma}(1.25^2/1.5\sqrt{p}, 5/6)$ prior on the length-scales and a Gamma prior on the magnitude with expectation $3/(\log(p) + 1)$ and variance 0.5. The constant means β_0 of the GPs have a $\mathrm{N}(0, 0.5^2)$ prior. The variance σ_y^2 has a $\log\text{-}\mathrm{N}(\log(0.01), 1)$. For the DP, the mass parameter has hyper-parameters $(u_a = 1, v_a = 1)$, and for the EDP, the mass parameters have hyper-parameters $(u_\theta = 1, v_\theta = 1)$ and $(u_\psi = 1, v_\psi = 1)$. A Gaussian input model is used with hyperparameters of the conjugate normal-inverse gamma set to $u_{0,d} = \bar{x}_d$, $c_d = 1/4$, $b_{x,d} = 1$, and $a_{x,d} = 2$.

Figure 5 depicts the heat map of the posterior similarity matrix and VI clustering estimate for the DP and for the y-level clustering of the EDP with $D = 5$; the over-partitioning of the DP is evident even for $D = 5$. Figure 6(a) emphasizes the improvement of the EDP over the DP in recovering the true clustering for increasing D. It also highlights the inability of model with linear experts to recover the true nonlinear structure for large D, due to the small sample sizes associated to each local linear approximation. The VI distance between the true and estimated (y-level) clustering is depicted in Figure 6(b), with dashed lines representing the size of the 95% VI credible balls. For the DP, the distance increases greatly with D, and the true clustering is far from the credible ball. The behaviour of the y-level clustering of the EDP is more robust to increasing D, while the x-level clustering requires an increasing number of clusters.

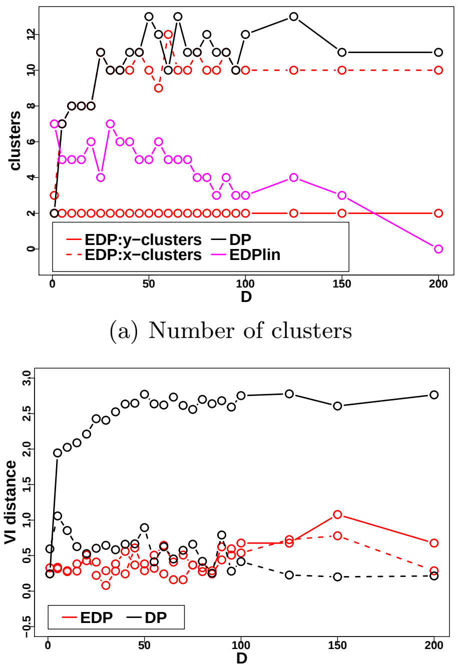


(a) DP PSM (b) DP clustering

(c) EDP PSM (d) EDP clustering

Figure 5: Simulated isotropic example with $D = 5$ dimensional inputs. *Left*: Heat map of the posterior similarity matrix (PSM), with observations indices permuted based on hierarchical clustering to improve visualisation. *Right:* VI clustering estimate with data points $(y_n, \bar{x}_{n,\cdot})$ coloured by cluster membership. Rows correspond to the DP and EDP MoE, respectively. For the EDP, plots correspond to the y-level clustering.

We compute the estimates for the predictive response density over a dense grid of y_* values against the mean of the test inputs across the input dimension. The difference between these estimated and the true densities is summarised by the L1 error in Figure 7 for increas-



(a) Number of clusters

(b) VI distance

Figure 6: Simulated isotropic example. Comparison of the EDP with the DP and the EDP model with linear experts, in terms of the number of clusters in the VI estimate and the VI distance between the true and estimated clustering (with dashed lines for the size of the credible ball).

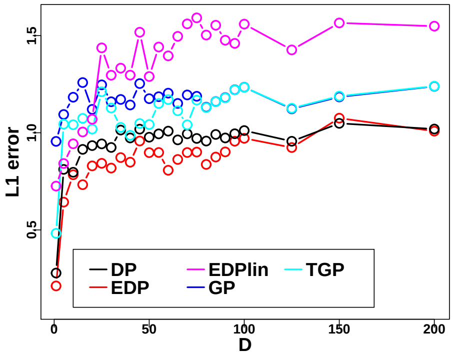


Figure 7: Simulated isotropic example. Comparison of the EDP with the DP, EDP model with linear experts, Lasso, GP, and TGP, in terms of the approximate L_1 distance between the estimated and true conditional densities.

ing D, and compared with the EDP model with linear experts, GP, and TGP. Figure 8 plots the predictive densities for different choices of D, for a more detailed comparison. The second and third rows correspond to the DP and EDP models respectively. An improvement with the EDP is visible, particularly in producing smooth estimates that avoid the sharp piecewise behavior due to the over-partitioning of the DP.

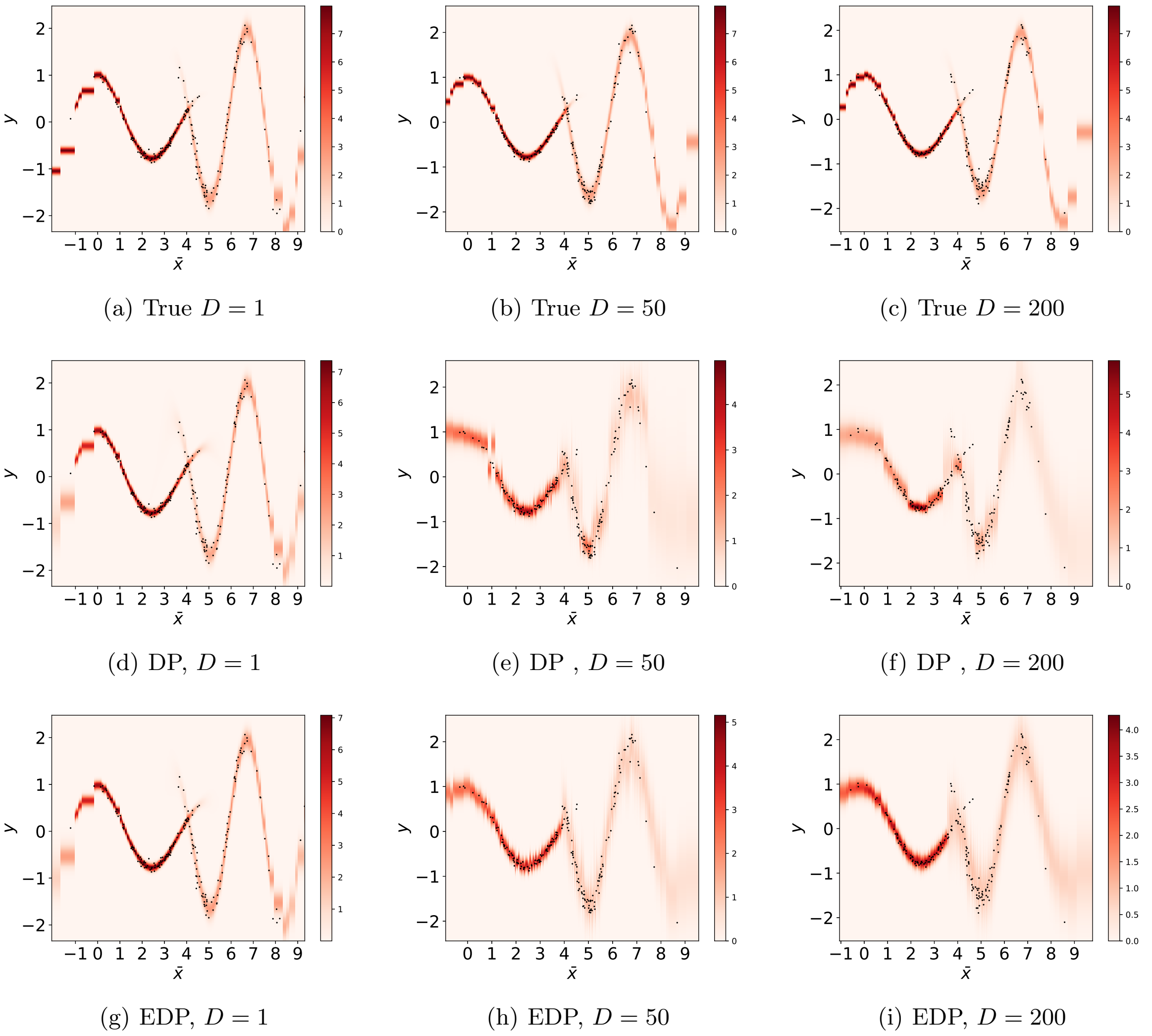
5.2 Alzheimer's Challenge

Training data for the challenge is extracted from the Alzheimer's Disease Neuro-Initiative (ADNI) database[1]. This data set consists of six inputs: age (in fraction of years), gender, the baseline mini mental-state exam (MMSE) score, the number of years an individual has spent in education, APOE genotype (recoded to reflect the number of copies of the type 4 allele), and the clinical diagnosis assessed at the baseline. The output is the MMSE score taken at a 24 month follow-up visit, and the task is to predict the cognitive decline in a patient over this period.

We again employ the same prior choices for both mixtures of GP experts models, based on identified reasonable ranges for the parameters. We consider an ARD squared exponential kernel for the GP with Gamma$(a_{l,d}, b_{l,d})$ priors on the length-scales with $a_l = (3, 2, 3, 5, 3, 2)$, and $b_l = (3/20, 5, 1, 1, 5, 4)$, in order of the inputs listed above. Additionally, we specify a Gamma(a_m, b_m) prior on the magnitude with $a_m = 2$ and $b_m = 1$. These parameters were selected to reflect our prior knowledge on the relationship between follow-up MMSE and the inputs and based on the range of the inputs. The GP is assumed to have a prior constant mean with a N$(20, 7.5^2)$ prior. The variance σ_y^2 has a Gamma(a_y, b_y) prior with $a_y = 1.5$ and $b_y = 0.5$.

For the DP, the mass parameter has hyper-parameters $(u_a = 1, v_a = 1)$, and for the EDP, the mass parameters have hyper-parameters $(u_\theta = 1, v_\theta = 1)$ and

[1] The ADNI was launched in 2003 by the National Institute on Aging (NIA), the National Institute of Biomedical Imaging and Bioengineering (NIBIB), the Food and Drug Administration (FDA), private pharmaceutical companies and non-profit organizations, as a \$ 60 million, 5-year public- private partnership. The primary goal of ADNI has been to test whether serial magnetic resonance imaging (MRI), positron emission tomography (PET), other biological markers, and clinical and neuropsychological assessment can be combined to measure the progression of mild cognitive impairment (MCI) and early Alzheimer's disease (AD). Determination of sensitive and specific markers of very early AD progression is intended to aid researchers and clinicians to develop new treatments and monitor their effectiveness, as well as lessen the time and cost of clinical trials. The Principal Investigator of this initiative is Michael W. Weiner, MD, VA Medical Center and University of California-San Francisco. ADNI is the result of efforts of many co-investigators from a broad range of academic institutions and private corporations, and subjects have been recruited from over 50 sites across the U.S. and Canada. The initial goal of ADNI was to recruit 800 adults, ages 55 to 90, to participate in the research, approximately 200 cognitively normal older individuals to be followed for 3 years, 400 people with MCI to be followed for 3 years and 200 people with early AD to be followed for 2 years. For up-to-date information, see `www.adni-info.org`.



(a) True $D = 1$

(b) True $D = 50$

(c) True $D = 200$

(d) DP, $D = 1$

(e) DP , $D = 50$

(f) DP , $D = 200$

(g) EDP, $D = 1$

(h) EDP, $D = 50$

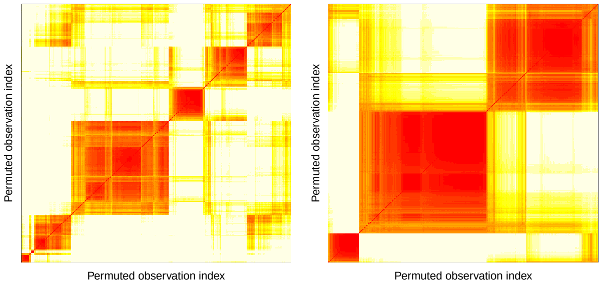
(i) EDP, $D = 200$

Figure 8: Simulated isotropic example. First row: the true data generating predictive density (red) evaluated at a grid of y_* values against the mean of test inputs across the input dimension. Bottom rows: predictive density for the DP (second row) and EDP (third row) with increasing $D = 1, 50, 200$ (rows). Here each D corresponds to separate training and test sets and the x-axis shows the mean across covariates. In each plot the training samples are shown as scatter points (black).

($u_\psi = 1, v_\psi = 1$). The parametric local model for x_n is the product of a normal density for age, a categorical density for gender, and four binomial densities for baseline MMSE, education, APOE4, and diagnosis. The input hyperparameters are $u_0 = 72$, $c = 2$, $b_x = 10$, and $a_x = 2$ for age; $\gamma_2 = (1, 1)$ for gender; $\gamma_3 = (5, 1)$ for MMSE; $\gamma_4 = (3, 2)$ for education; $\gamma_5 = (1, 3)$ for APOE4; $\gamma_6 = (1, 1)$ for diagnosis. Posterior inference for both models is performed with 5000 total iterations and a burn-in of 1000.

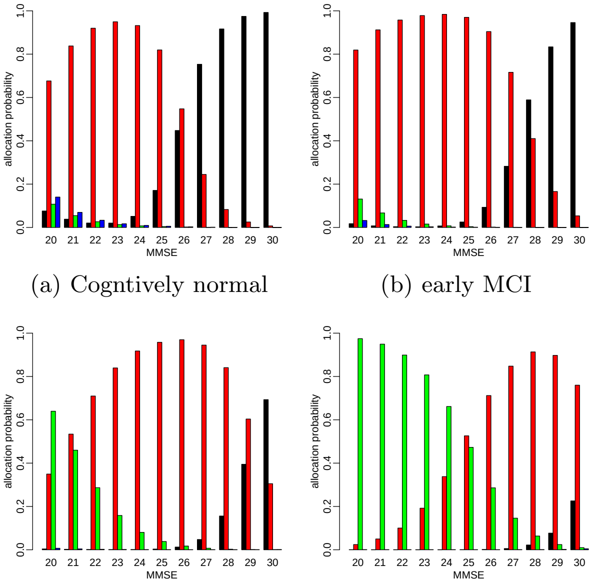
For comparison with the best performers of this subchallenge, the GuanLab and ADDT teams, we implemented the models using publicly available packages in **R**. For the GuanLab model, we used the **svm** function of the **e1071** package (Meyer et al., 2018). For the ADDT model, we used the **rlm** function of the **MASS** package (Venables and Ripley, 2002). For both mixtures of experts, posterior medians, i.e. the point estimate under the absolute error loss, are used to predict MMSE scores, which are appropriate due to the heavy left tail of the predictive densities.

Heat maps of the posterior similarity matrices are provided in Figure 9, and visualizations of the VI clusterings through side-by-side bar plots of MMSE baseline, MMSE follow-up, education, diagnosis, APOE4, gender and age are provided in Figure 11, with colours representing clusters. Interestingly, the enriched model identifies three clusters consisting mostly of cogni-



(a) DP (b) EDP

Figure 9: Alzheimer's challenge. Heat map of the posterior similarity matrix for the DP and EDP MoE. To improve visualisation, observations are permuted based on hierarchical clustering.



(a) Cogntively normal (b) early MCI

(c) late MCI (d) Alzheimer's disease

Figure 10: Alzheimer's challenge. The allocation probabilities for a new test point as a function of baseline MMSE and diagnosis of CN in (a), eMCI in (b), lMCI in (c), and AD in (d), with other inputs marginalised. Allocation probabilities are based on the estimated VI clustering and coloured by cluster membership for each of the estimated VI clusters from the enriched model in Figure 11.

tively normal (black), mild cognitive impairment (red), and AD (green) individuals, similar to the GuanLab model, with slight modifications considering the other variables, particularly, MMSE baseline and follow-up scores. For example, one late MCI individual is allocated to the AD (green) cluster in Figure 11(d) due to the observed sharp drop in MMSE from 27 at baseline to 8 at follow-up. Additionally, we observe that the relative proportion of individuals in the red and green clusters increases with higher APOE4, but does not (marginally) depend on gender and age.

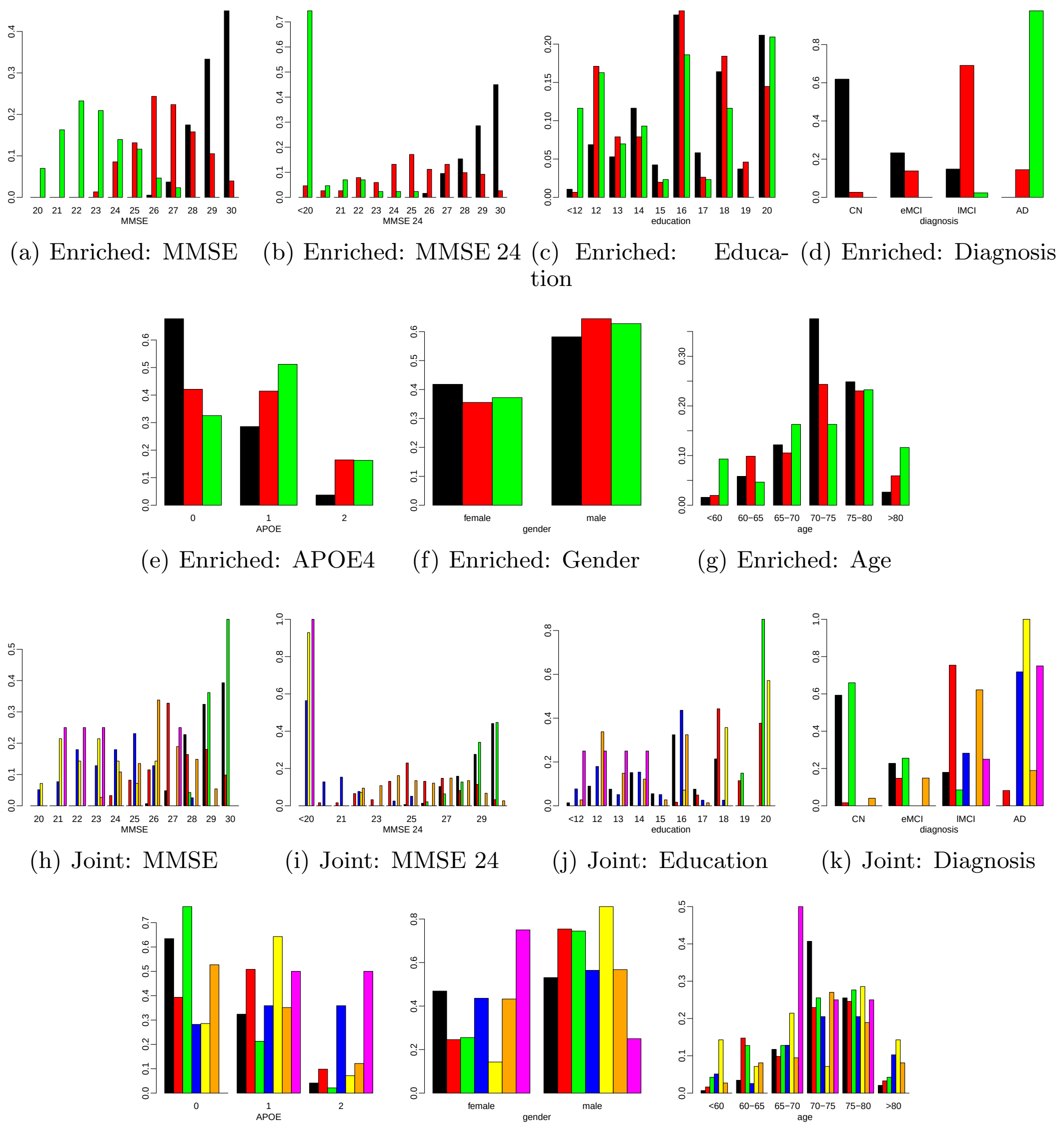
The DP, on the other hand, further subdivides clusters due to multimodality in education. Similarly, for the enriched model, the VI estimate of x-clustering within each VI estimated y-cluster, contains two x-clusters due to multimodality in education. Figure 12 depicts the heap map of the posterior similarity matrix for the x-clustering within each estimated y-cluster and also shows the VI estimate of x-clustering within each VI estimated y-cluster for education, with each estimated x-clustering containing two clusters.

We can further appreciate the difference between the deterministic clustering of the GuanLab model and the stochastic clustering of the enriched model in Figure 10, which shows the allocation probabilities of a new test point for MMSE baseline scores of 20-30 and diagnosis of CN (Figure 10(a)), eMCI (Figure 10(b)), lMCI (Figure 10(c)), AD (Figure 10(d)), with other inputs marginalised. As opposed to the GuanLab model which classifies new individuals based on diagnosis, we observe that CN individuals with baseline MMSE ≥ 27 have the highest probability of being allocated to the black cluster, while this baseline MMSE cutoff is increased to 28 and 30 for eMCI and lMCI individuals, respectively. Below these respective cutoffs, CN, eMCI, and lMCI individuals have the highest probability of being allocated to the red cluster (apart from lMCI individuals with baseline MMSE of 20 that are allocated to the green cluster with highest probability). Instead, AD individuals have the highest probability of belonging to the red cluster for baseline MMSE ≥ 25 and to the green cluster otherwise. We note that for CN individuals with low MMSE baseline (not observed), there is a small probability of allocation to a new (blue) cluster.

Figure 13 shows how the predictive densities of MMSE follow-up scores change given different combinations of baseline MMSE, diagnosis, and APOE4. For CN individuals, the differences between APOE4 type are minor and the posterior mass is very concentrated on high follow-up MMSE scores given high baseline MMSE scores. More evident differences between APOE4 type are visible for more severe diagnosis, and in general, we observe a greater decrease in follow-up scores with more uncertainty for more severe dementia and increased APOE4. In particular, for AD patients that are carriers of APOE4, there is a visible probability of progressing to severe dementia (MMSE≤ 12), that increases with decreased baseline MMSE.

Bibliography

S. Duane, A.D. Kennedy, B.J. Pendleton, and D. Roweth. Hybrid Monte Carlo. *Physics Letters B*, 195(2):216–222, 1987.



(a) Enriched: MMSE (b) Enriched: MMSE 24 (c) Enriched: Education (d) Enriched: Diagnosis

(e) Enriched: APOE4 (f) Enriched: Gender (g) Enriched: Age

(h) Joint: MMSE (i) Joint: MMSE 24 (j) Joint: Education (k) Joint: Diagnosis

(l) Joint: APOE4 (m) Joint: Gender (n) Joint: Age

Figure 11: Alzheimer's challenge. A visualization of the VI clustering estimate through side-by-side bar plots coloured by cluster membership. The first two rows correspond to enriched model and the second two correspond to the joint model.

M.D. Escobar and M. West. Bayesian density estimation and inference using mixtures. *Journal of the American Statistical Association*, 90:577–588, 1995.

GPy. GPy: A Gaussian process framework in Python. `http://github.com/SheffieldML/GPy`, since 2012.

J.H. Kotecha and P.M. Djuric. Gibbs sampling approach for generation of truncated multivariate Gaussian random variables. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 1757–1760, 1999.

A. Kottas, P. Müller, and F. Quintana. Nonparametric Bayesian modeling for multivariate ordinal data. *Journal of Computational and Graphical Statistics*, 14:610–625, 2005.

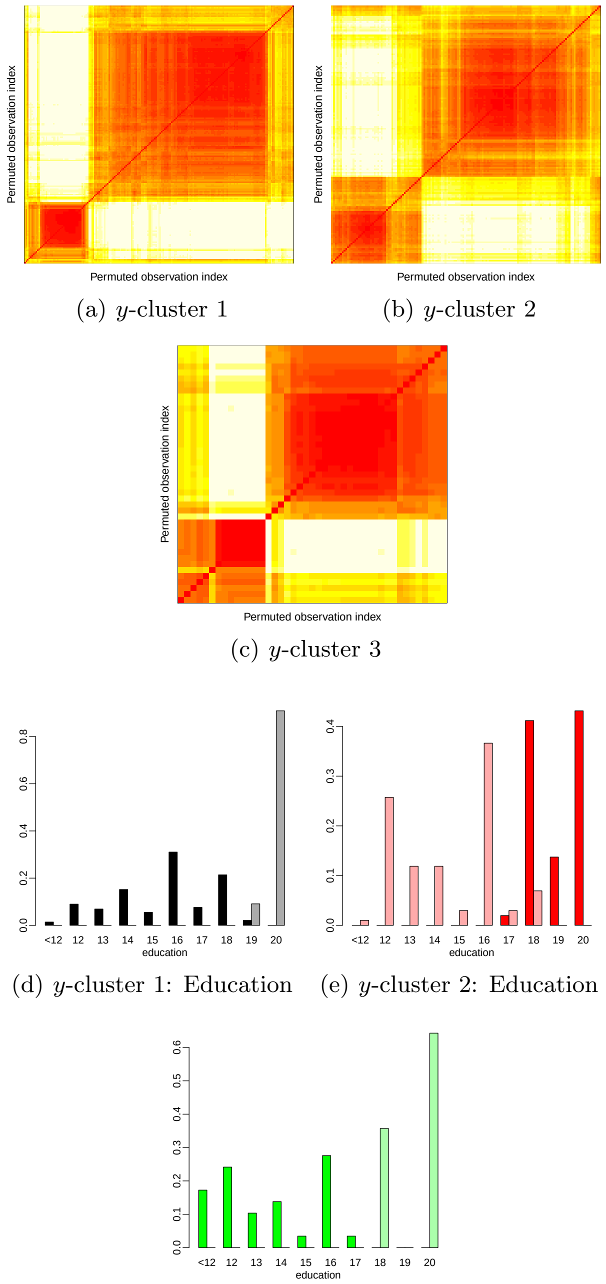
D. Meyer, E. Dimitriadou, K. Hornik, A. Weingessel, and F. Leisch. *e1071: Misc functions of the Department of Statistics, Probability Theory Group (Formerly: E1071), TU Wien*, 2018. URL `https://CRAN.R-project.org/package=e1071`.

R.M. Neal. Markov chain sampling methods for Dirichlet process mixture models. *Journal of Computational and Graphical Statistics*, 9(2):249–265, 2000.

T.J. Santner, B.J. Williams, and W.I. Notz. *The Design and Analysis of Computer Experiments*. Springer-Verlag, New York, 2003.

W. N. Venables and B. D. Ripley. *Modern Applied Statistics with S*. Springer, New York, fourth edition, 2002.

S. Wade, D.B. Dunson, S. Petrone, and L. Trippa. Im-



(a) y-cluster 1

(b) y-cluster 2

(c) y-cluster 3

(d) y-cluster 1: Education

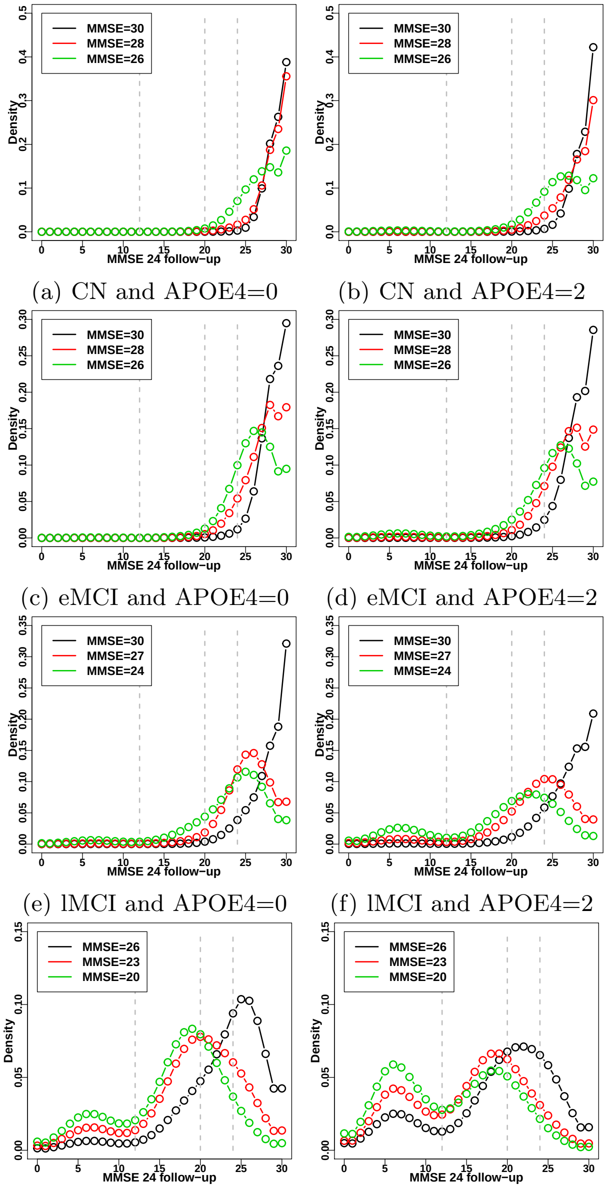
(e) y-cluster 2: Education

(f) y-cluster 3: Education

Figure 12: Alzheimer's challenge. First row: heat maps of the posterior similarity matrices for the x-clustering within each y-cluster for the enriched model. Second row: a visualization of the VI x-clustering estimate within each y-cluster through side-by-side bar plots for education. Colour corresponds to the y-cluster, while shading corresponds to the x-cluster.

proving prediction from Dirichlet process mixtures via enrichment. *Journal of Machine Learning Research*, 15 (1):1041–1071, 2014.

W. Wang and S.J. Russell. A smart-dumb/dumb-smart algorithm for efficient split-merge MCMC. In *Uncertainty in Artificial Intelligence*, pages 902–911, 2015.



(a) CN and APOE4=0

(b) CN and APOE4=2

(c) eMCI and APOE4=0

(d) eMCI and APOE4=2

(e) lMCI and APOE4=0

(f) lMCI and APOE4=2

(g) AD and APOE4=0

(h) AD and APOE4=2

Figure 13: Alzheimer's challenge. Predictive distribution of MMSE 24-month follow-up for different combinations of MMSE baseline, diagnosis, and APOE4, with other inputs marginalised for the enriched model. Columns represent APOE4 types of 0 and 2, whilst rows represent diagnosis. Dashed lines indicate established cutoffs for MMSE: ≥ 25 suggests no dementia; $20-24$ suggests mild dementia; $13-19$ suggests moderate dementia; ≤ 12 suggests severe dementia.