Electronic Journal of Statistics
Vol. 6 (2012) 1170–1199
ISSN: 1935-7524
DOI: 10.1214/12-EJS705

Simultaneous variable selection and component selection for regression density estimation with mixtures of heteroscedastic experts[*]

Minh-Ngoc Tran[†] and David J. Nott

Australian School of Business, University of New South Wales
Sydney 2052, Australia
Department of Statistics and Applied Probability, National University of Singapore
Singapore 117546
e-mail: minh-ngoc.tran@unsw.edu.au; standj@nus.edu.sg

Robert Kohn[†]

Australian School of Business, University of New South Wales
Sydney 2052, Australia e-mail: r.kohn@unsw.edu.au

Abstract: This paper is concerned with the problem of flexibly estimating the conditional density of a response variable given covariates. In our approach the density is modeled as a mixture of heteroscedastic normals with the means, variances and mixing probabilities all varying smoothly as functions of the covariates. We use the variational Bayes approach and propose a novel fast algorithm for simultaneous covariate selection, component selection and parameter estimation. Our method is able to deal with the local maxima problem inherent in mixture model fitting, and is applicable to high-dimensional settings where the number of covariates can be larger than the sample size. In the special case of the classical regression model, the proposed algorithm is similar to currently used greedy algorithms while having many attractive properties and working efficiently in high-dimensional problems. The methodology is demonstrated through simulated and real examples.

AMS 2000 subject classifications: Primary 62G07; secondary 62G08.
Keywords and phrases: Bayesian model selection, heteroscedasticity, mixture of normals, variational approximation.

Received January 2012.

Contents

[*] The first author dedicates the paper to Professor Nguyen Duy Tien on the occasion of his seventieth birthday.

[†] The research of Minh-Ngoc Tran and Robert Kohn was partially supported by Australian Research Council grant DP0667069. The authors would like to thank Siew Li Tan for her R code to produce Figures 1 and 3.

1. Introduction

In this paper we are concerned with the problem of flexible regression density estimation. The term regression density estimation refers to the problem of flexibly estimating the conditional density function of a response variable y at all points $\boldsymbol{x}$ in the covariate space, while making relatively few assumptions about its functional form [28]. This is an important problem in applications where the response distribution is highly multimodal and would not be appropriately modeled by a simple parametric density such as a normal.

A well-established methodology for regression density estimation uses mixtures of heteroscedastic experts models [10, 28, 19]. This approach extends mixture of experts models [12, 14] by allowing components to be heteroscedastic and allowing mixing probabilities to depend on the covariates. In this paper we consider mixtures of heteroscedastic normals. More specifically, the conditional density of a response y given a covariate vector $\boldsymbol{x}$ is modeled as

$$p(y|\boldsymbol{x}) = \sum_{j=1}^{k} \pi_j(\boldsymbol{x}) N(y|\mu_j(\boldsymbol{x}), \sigma_j^2(\boldsymbol{x})), \tag{1.1}$$

where $\pi_j(\boldsymbol{x})$, $\mu_j(\boldsymbol{x})$ and $\sigma_j^2(\boldsymbol{x})$ are (functions of) linear combinations of $\boldsymbol{x}$, $\pi_j(\boldsymbol{x}) \geq 0$, $\pi_1(\boldsymbol{x}) + \cdots + \pi_k(\boldsymbol{x}) = 1$ and k is the number of components. We will refer to this model as the RDE-MHN(k) (regression density estimation with mixtures of k heteroscedastic normals) model. Hereafter, the terms *mean model, variance model* and *gating model* refer to the models for the means μ_j, variances σ_j^2 and mixing probabilities π_j, respectively.

[30] and [31] carry out Bayesian analysis on mixtures of experts models with flexible terms for the covariates, although they do not consider heteroscedasticity. [10] consider model (1.1) for regression density estimation in which only the means μ_j are allowed to depend on $\boldsymbol{x}$. [28] and [19] extend to the heteroscedastic case in which the mixing probabilities π_j, component means μ_j and variances σ_j^2 all varying with covariates. The heteroscedastic extension to mixture of experts models is important in applications where the conditional distribution of y is very complex and there is a need to model $p(y|\boldsymbol{x})$ flexibly without making too rigid assumptions on its functional form. As discussed in [19] and [28], the performance of mixtures of homoscedastic models (i.e. model (1.1) with constant σ_j^2), when used to model heteroscedastic data, deteriorates as the number of covariates increases, and cannot be improved by simply increasing the number of components. Ignoring heteroscedasticity may lead to serious problems in inference, such as misleading assessments of significance, poor predictive performance and inefficient estimation of the mean parameters. The reader is referred to [5] and [25] for a more detailed discussion on heteroscedastic modeling.

[28] use Bayesian inference and Markov chain Monte Carlo (MCMC) methods to estimate the RDE-MHN model. Using MCMC, however, may be computationally demanding in high-dimensional situations with a large number of covariates, and in time series data modeling where sequential updating is required. [19] develop a fast alternative computationally attractive estimation method using variational approximation. Their variational approximation method is computationally attractive in situations where it is necessary to re-fit complex models many times such as for sequential updating in time series data analysis. Using variational approximation for fitting mixtures of *homoscedastic* experts models is considered by a number of authors [29, 23, 2]. See also [7, 17] and [32] for applications of variational approximation to fitting Gaussian mixture models. Section 3 briefly reviews the variational approximation method.

The first issue in RDE-MHN modeling is selecting the number of components k. [28] and [19] consider this problem by fitting separate RDE-MHN models within a proposed range of potential k and selecting the one with largest cross-validation log predictive density score (see the definition in Section 6). This approach has several drawbacks. First, cross-validation is not natural for ordered data such as time series or longitudinal data (see the stock return example in Section 6.3). Second, computing the cross-validation log predictive density score may be very time consuming if the sample is divided into many parts. Third, the log predictive density score method requires prior information on the maximum number of components which may be hard to obtain.

The second important issue in RDE-MHN modeling is variable selection. Variable selection is a fundamental problem in general regression analysis in which a large number of potential covariates is often introduced at the initial stage of modeling and it is necessary to select from them a smaller subset to fit the data in order to avoid overfitting, reduce the cost of data collection and increase model interpretability. See, for example, [9, 8, 20] and references therein. Variable selection is not discussed in [19]. Incorporating variable selection is essential in complex models like ours, especially in high dimension where the full model fitting of [19] is almost impossible. Variable selection helps not only to improve performance by producing parsimonious models, but also reduces the dimension of the parameter space and makes the computation faster.

Another important issue in mixture model fitting in general is the local maxima problem in which the fitting procedure converges to one of the local maxima [27, 18, 22]. In our experience, this problem occurs very often and gives a suboptimal solution which may cause serious consequences in subsequent inferences.

Our article proposes a novel algorithm that is able to deal with the aforementioned problems. We employ a Bayesian approach and use the variational approximation method for fitting. We first modify the variational approximation fitting approach in [19] to deal with the local maxima problem by using the split-and-merge idea [see, for example, 22, 32] which repeatedly splits and/or merges poorly fitted components on the basis of maximizing the variational lower bound considered as a good approximation of the log marginal likelihood. This approach also automatically determines the number of components. We then propose a strategy for ranking covariates for inclusion into the mean, variance and gating models in a computationally thrifty way. This is a non-trivial extension of the ranking algorithm introduced in [20] who consider variable selection in the heteroscedastic linear regression model, i.e. RDE-MHN(1). These together result in a novel fast method for simultaneous variable selection, component selection and parameter estimation in the RDE-MHN modeling. Our method can be used in high-dimensional situations where the number of covariates can be much larger than the sample size.

The MCMC method of [28] and their novel variable selection prior provide an excellent approach for RDE-MHN modeling in low-dimensional settings where computation time is not a primary concern. A major advantage of our method over their MCMC method is that we provide a fast alternative; see Sections 5 and 6. It is obvious that the MCMC method is not applicable in high-dimensional problems with thousands of covariates. Another advantage is that our method does variable selection and component selection simultaneously and automatically rather than fitting separate RDE-MHN models within a range of potential k and then selecting an appropriate one based on some model selection criterion which in turn requires a considerable amount of extra computation time.

Many of the popular models in the literature are special cases of the RDE-MHN model. Model (1.1) with $k = 1$ is the heteroscedastic linear regression model considered in [25], Chapter 14 and [20]

$$y_i = \boldsymbol{x}_i'\boldsymbol{\beta} + \sigma_i\epsilon_i, \quad \sigma_i^2 = \exp(\boldsymbol{x}_i'\boldsymbol{\alpha}), \quad \epsilon_i \sim N(0,1),$$

where $(y_i, \boldsymbol{x}_i),\ i = 1, \ldots, n$ are observations. With $k = 1$ and σ^2 constant, (1.1) reduces further to the classical linear regression model

$$y_i = \boldsymbol{x}_i'\boldsymbol{\beta} + \sigma\epsilon_i, \quad \epsilon_i \sim N(0,1),$$

which is extensively studied in the literature. A nice feature of our methodology is that it is able to reach a simple model if such a model is warranted. With

$k = 1$ and constant variance, our ranking algorithm for variable selection is similar to widely used matching pursuit and greedy algorithms for model search [16, 33, 8].

The RDE-MHN modeling approach provides flexibility in exploring data in which the conditional density of interest has a complex structure. Section 6 applies our approach to analyzing the diabetes data [8]. This data set consists of observations on 442 patients about their 10 baseline variables (predictors $\boldsymbol{x}$) and a quantitative measure of disease progression (response y). Previous analysis in the literature assumed a RDE-MHN(1) model [8, 20]. Our goal is to see if relaxing the assumption of $k = 1$ components can give us more flexibility in exploring the structure of $p(y|\boldsymbol{x})$, and lead to a model with better predictive performance. We find that the RDE-MHN(3) model is selected by our algorithm which has better predictive performance than previously selected models. Figure 1 shows the clear three-component structure in the conditional distribution explored by our approach (see Section 6 for the details).

The rest of the paper is organized as follows. Section 2 describes the RDE-MHN model in detail. Section 3 combines the variational approximation method and the split-and-merge algorithm for fitting this model. Section 4 presents our fast greedy algorithm for variable selection. Sections 5 and 6 present simulation studies and real data examples illustrating our method. Section 7 concludes. Technical derivations are placed in the Appendices.

2. Mixture of heteroscedastic normals model

Let $(y_i, \boldsymbol{x}_i)$, $i = 1, \dots, n$, be n observations with y_i univariate responses and $\boldsymbol{x}_i = (x_{i1}, \dots, x_{is})'$ corresponding covariate vectors. We will write y and $\boldsymbol{x} = (x_1, \dots, x_s)'$ for a generic response and covariate vector. We are concerned with the problem of estimating the conditional distribution of y given $\boldsymbol{x}$ using mixture of experts models and with the problem of selecting important covariates as well as selecting the number of experts. The distribution of y_i given $\boldsymbol{x}_i$ is modeled by a mixture of heteroscedastic normals model as follows

$$y_i|\delta_i = j, \boldsymbol{x}_i, \boldsymbol{\beta}, \boldsymbol{\alpha} \sim N(\boldsymbol{v}_i'\boldsymbol{\beta}_j, \exp(\boldsymbol{w}_i'\boldsymbol{\alpha}_j)), \quad i = 1, \dots, n,$$

where δ_i is a latent variable indicating the component y_i comes from, $\delta_i \in \{1, 2, \dots, k\}$, $\boldsymbol{v}_i = (v_{i1}, \dots, v_{ip})'$ and $\boldsymbol{w}_i = (w_{i1}, \dots, w_{iq})'$ are vectors of covariates in the mean and variance models (which are sub-vectors of $\boldsymbol{x}_i$), $\boldsymbol{\beta}_j = (\beta_{j1}, \dots, \beta_{jp})'$ and $\boldsymbol{\alpha}_j = (\alpha_{j1}, \dots, \alpha_{jq})'$ are vectors of unknown parameters in the mean and variance models of the jth component, respectively. Write $\boldsymbol{\beta} = (\boldsymbol{\beta}_1', \dots, \boldsymbol{\beta}_k')'$, $\boldsymbol{\alpha} = (\boldsymbol{\alpha}_1', \dots, \boldsymbol{\alpha}_k')'$. The distribution of the latent variables δ_i is modeled by a multinomial logit regression model

$$P(\delta_i = j|\boldsymbol{\gamma}, \boldsymbol{x}_i) = \frac{\exp(\boldsymbol{z}_i'\boldsymbol{\gamma}_j)}{\sum_{l=1}^k \exp(\boldsymbol{z}_i'\boldsymbol{\gamma}_l)}, \quad j = 1, \dots, k; \ i = 1, \dots, n,$$

where $\boldsymbol{\gamma}_j = (\gamma_{j1}, \dots, \gamma_{jr})'$ is a vector of unknown parameters in the gating model of the jth component, $j = 2, \dots, k$. Write $\boldsymbol{\gamma} = (\boldsymbol{\gamma}_2', \dots, \boldsymbol{\gamma}_k')'$ and $\boldsymbol{\delta} =$

$(\delta_1, \ldots, \delta_n)'$. Note that we set $\boldsymbol{\gamma}_1 \equiv \mathbf{0}$ for identifiability. Again, $\boldsymbol{z}_i = (z_{i1}, \ldots, z_{ir})'$ is a sub-vector of $\boldsymbol{x}_i$, and contains the covariates used to model the mixing probabilities. We assume here and in Section 3 that we know in advance which covariates from $\boldsymbol{x}$ are included in the mean, variance and gating models. The variable selection issue will be discussed in Section 4. The above model will be referred to as the regression density estimation with mixtures of k heteroscedastic normals model, denoted by RDE-MHN(k).

This article employs a Bayesian approach and uses normal distributions for priors on the parameters $\boldsymbol{\beta}$, $\boldsymbol{\alpha}$ and $\boldsymbol{\gamma}$

$$p(\boldsymbol{\beta}) = \prod_{j=1}^{k} N(\boldsymbol{\mu}^0_{\beta_j}, \Sigma^0_{\beta_j}), \quad p(\boldsymbol{\alpha}) = \prod_{j=1}^{k} N(\boldsymbol{\mu}^0_{\alpha_j}, \Sigma^0_{\alpha_j}), \quad p(\boldsymbol{\gamma}) = N(\boldsymbol{\mu}^0_{\gamma}, \Sigma^0_{\gamma}),$$

where $\boldsymbol{\mu}^0_{\beta_j}, \Sigma^0_{\beta_j}, \boldsymbol{\mu}^0_{\alpha_j}, \Sigma^0_{\alpha_j}, \boldsymbol{\mu}^0_{\gamma}$ and Σ^0_{γ} are hyperparameters. We assume that $\boldsymbol{\beta}$, $\boldsymbol{\alpha}$ and $\boldsymbol{\gamma}$ are independent a priori, i.e.

$$p(\boldsymbol{\beta}, \boldsymbol{\alpha}, \boldsymbol{\gamma}) = p(\boldsymbol{\beta})p(\boldsymbol{\alpha})p(\boldsymbol{\gamma}).$$

The main issues relating to the implementation of the RDE-MHN modeling are: (i) estimating the parameters $\boldsymbol{\beta}$, $\boldsymbol{\alpha}$ and $\boldsymbol{\gamma}$ in a fast and reliable way, (ii) selecting the important covariates and, (iii) selecting the number of components k. Another important issue when fitting mixture models is the local maxima problem. This problem happens when the fitting algorithm (by the maximum likelihood method, for example) converges to a local rather than the global maximum [27, 18, 22]. In later sections we present a method for dealing with these problems.

3. Fitting the RDE-MHN model

3.1. The variational approximation fitting approach

Our method for fitting and doing model selection in the RDE-MHN model is based on the variational approximation fitting approach of [19]. It is reproduced here and in the Appendices to make the paper self-contained. The reader who is not familiar with variational approximation is referred to, for example, [1] or [21].

The RDE-MHN(k) model can be written as

$$\begin{aligned}
y_i|\delta_i = j, \boldsymbol{\beta}, \boldsymbol{\alpha} &\sim N(\boldsymbol{v}_i'\boldsymbol{\beta}_j, \exp(\boldsymbol{w}_i'\boldsymbol{\alpha}_j)),\ i = 1, \ldots, n\\
\boldsymbol{\beta}_j &\sim N(\boldsymbol{\mu}^0_{\beta_j}, \Sigma^0_{\beta_j}),\ j = 1, \ldots, k\\
\boldsymbol{\alpha}_j &\sim N(\boldsymbol{\mu}^0_{\alpha_j}, \Sigma^0_{\alpha_j}),\ j = 1, \ldots, k\\
P(\delta_i = j|\boldsymbol{\gamma}) &= p_{ij}(\boldsymbol{\gamma}) = \frac{\exp(\boldsymbol{z}_i'\boldsymbol{\gamma}_j)}{\sum_{l=1}^k \exp(\boldsymbol{z}_i'\boldsymbol{\gamma}_l)}, \quad j = 1, \ldots, k;\ i = 1, \ldots, n\\
\boldsymbol{\gamma} &\sim N(\boldsymbol{\mu}^0_{\gamma}, \Sigma^0_{\gamma}).
\end{aligned}$$

Write $\boldsymbol{\theta} = (\boldsymbol{\beta}, \boldsymbol{\alpha}, \boldsymbol{\gamma}, \boldsymbol{\delta})$. Our Bayesian inferences are based on the posterior distribution $p(\boldsymbol{\theta}|\boldsymbol{y}) \propto p(\boldsymbol{\theta})p(\boldsymbol{y}|\boldsymbol{\theta})$ which is difficult to handle. We proceed by approximating this posterior by a more tractable distribution $q(\boldsymbol{\theta})$. The *variational approximation posterior* $q(\boldsymbol{\theta})$ is selected by minimizing the Kullback-Leibler (KL) divergence

$$\int \log \frac{q(\boldsymbol{\theta})}{p(\boldsymbol{\theta}|\boldsymbol{y})} q(\boldsymbol{\theta}) d\boldsymbol{\theta}$$

among some restricted class of functions. From the identity

$$\log\, p(\boldsymbol{y}) \quad = \quad \int \log \frac{p(\boldsymbol{\theta})p(\boldsymbol{y}|\boldsymbol{\theta})}{q(\boldsymbol{\theta})} q(\boldsymbol{\theta}) d\boldsymbol{\theta} + \int \log \frac{q(\boldsymbol{\theta})}{p(\boldsymbol{\theta}|\boldsymbol{y})} q(\boldsymbol{\theta}) d\boldsymbol{\theta}, \qquad (3.1)$$

we see that minimizing the KL divergence is equivalent to maximizing

$$L(q) \quad = \quad \int \log \frac{p(\boldsymbol{\theta})p(\boldsymbol{y}|\boldsymbol{\theta})}{q(\boldsymbol{\theta})} q(\boldsymbol{\theta}) d\boldsymbol{\theta}. \qquad (3.2)$$

Because of the non-negativity of the KL divergence term in (3.1), (3.2) is a lower bound on the log marginal likelihood log $p(\boldsymbol{y})$. The lower bound (3.2), when maximized with respect to q, is often used as an approximation to the log marginal likelihood log $p(\boldsymbol{y})$. This approximation is useful, since log $p(\boldsymbol{y})$ is a key quantity in Bayesian model selection. The accuracy of variational approximation is experimentally studied in [19, 20]. Some results on the asymptotic normality of variational approximation estimators are recently obtained in [11].

[19] develop a variational approximation approach for fitting the RDE-MHN model with the variational approximation posterior $q(\boldsymbol{\theta})$ assuming the following product form

$$q(\boldsymbol{\theta}) = q(\boldsymbol{\beta})q(\boldsymbol{\alpha})q(\boldsymbol{\delta})q(\boldsymbol{\gamma}),$$

where

$$q(\boldsymbol{\beta}) = \prod_{j=1}^{k} q(\boldsymbol{\beta}_j), \quad q(\boldsymbol{\alpha}) = \prod_{j=1}^{k} q(\boldsymbol{\alpha}_j), \quad q(\boldsymbol{\delta}) = \prod_{i=1}^{n} q(\boldsymbol{\delta}_i),$$

and $q(\boldsymbol{\beta}_j)$ is normal $N(\boldsymbol{\mu}^q_{\beta_j}, \Sigma^q_{\beta_j})$, $q(\boldsymbol{\alpha}_j)$ is normal $N(\boldsymbol{\mu}^q_{\alpha_j}, \Sigma^q_{\alpha_j})$, $q(\delta_i = j) = q_{ij}$ where $\sum_j q_{ij} = 1,\ i = 1, \ldots, n$. The posterior $q(\boldsymbol{\gamma})$ is assumed to be the Dirac delta distribution $\delta_{\boldsymbol{\mu}^q_\gamma}(\cdot)$ concentrated at a point $\boldsymbol{\mu}^q_\gamma$, i.e. $\delta_{\boldsymbol{\mu}^q_\gamma}(A) = 1$ if and only if $\boldsymbol{\mu}^q_\gamma \in A$ for every Borel set A in $I\!R^{(k-1)r}$. Using the Dirac delta distribution for the variational approximation posterior of $\boldsymbol{\gamma}$ means that we are interested in its posterior mode, and this facilitates the computation. Note that, for simplicity, with a little abuse of notation we have not distinguished between distributions and densities in the above. The advantage of the variational approximation posterior above is that the lower bound (3.2) has a closed form (see Appendix A), which allows fast and easy optimization. Algorithm 1 in Appendix A summarizes a procedure for maximizing this lower bound.

3.2. The split-and-merge algorithm

Two difficult issues associated with the above mixture model concern the existence of local maxima and the selection of the number of components [27, 18]. We now address these problems by adapting the split-and-merge algorithm discussed in [22]. See also [24] and [32]. The idea of the split-and-merge algorithm is to repeatedly merge two components and/or split a component until some criterion is satisfied. This algorithm has been proven useful in overcoming the local maxima problem and automatically determining the number of components [22, 32]. We now adapt this idea to our RDE-MHN model.

We first initialize the number of components k using the method of [4] for selecting the number of clusters. With an initial k, after Algorithm 1 has converged, we consider merging two components or splitting a component until the lower bound is not improved any further. Let $\boldsymbol{\theta}^*$ and L^* denote the parameter estimate and the maximized lower bound after Algorithm 1 converges.

Merge criterion Two components are considered most plausible for merging if they are close to each other in some sense. Here we use the Kullback-Leibler (KL) divergence to measure similarity. The KL distance between two distributions P and Q is defined as

$$\mathrm{KL}(P,Q) := \frac{1}{2}(\mathrm{KL}(P\|Q)+\mathrm{KL}(Q\|P)),$$

where $\mathrm{KL}(P\|Q) = \int \log\frac{P(x)}{Q(x)}dP(x)$ is the KL divergence of Q to P. If $P \sim N(\mu_1,\sigma_1)$, $Q \sim N(\mu_2,\sigma_2)$, then

$$\mathrm{KL}(P,Q) = \frac{1}{4}\left(\frac{(\mu_1-\mu_2)^2+\sigma_1^2}{\sigma_2^2}+\frac{(\mu_1-\mu_2)^2+\sigma_2^2}{\sigma_1^2}-2\right).$$

In our context, the KL distance (averaged over n observed points) between two components j_1 and j_2 is given by

$$\begin{aligned}\mathrm{KL}(j_1,j_2) \quad = \quad & \frac{1}{4n}\sum_{i=1}^{n}\Bigg(\frac{(\boldsymbol{v}_i'\boldsymbol{\mu}^q_{\beta_{j_1}}-\boldsymbol{v}_i'\boldsymbol{\mu}^q_{\beta_{j_2}})^2+\exp(\boldsymbol{w}_i'\boldsymbol{\mu}^q_{\alpha_{j_1}})}{\exp(\boldsymbol{w}_i'\boldsymbol{\mu}^q_{\alpha_{j_2}})}\\ & \qquad +\frac{(\boldsymbol{v}_i'\boldsymbol{\mu}^q_{\beta_{j_1}}-\boldsymbol{v}_i'\boldsymbol{\mu}^q_{\beta_{j_2}})^2+\exp(\boldsymbol{w}_i'\boldsymbol{\mu}^q_{\alpha_{j_2}})}{\exp(\boldsymbol{w}_i'\boldsymbol{\mu}^q_{\alpha_{j_1}})}-2\Bigg).\end{aligned}$$

The smaller the KL distance of two components, the more plausible they are as candidates for merging. Let $\mathcal{C} = \{(j_1,j_2),\ j_1 = 1,\dots,k;\ j_2 = 1,\dots,k;\ j_1 \neq j_2\}$ be the set of index pairs, $\xi_1 = \mathrm{argmin}\{\mathrm{KL}(j_1,j_2),\ (j_1,j_2)\in\mathcal{C}\}$ be the index pair of two components with the smallest KL distance, $\xi_i = \mathrm{argmin}\{\mathrm{KL}(j_1,j_2), (j_1,j_2)\in\mathcal{C}\setminus\{\xi_1,\dots,\xi_{i-1}\}\},\ i = 2,\dots$ Write $\mathcal{C}_{\mathrm{merge}} = \{\xi_1,\xi_2,\dots\}$. Note that $\mathcal{C}_{\mathrm{merge}} = \mathcal{C}_{\mathrm{merge}}(\boldsymbol{\theta}^*)$ depends on $\boldsymbol{\theta}^*$. The idea is to try merging the most plausible pairs of components until the lower bound is improved or the number of merging operations exceeds a pre-specified number $C^{\max}_{\mathrm{merge}}$.

Merge operation To estimate the parameters of the new merging model, it is important to make use of the previous estimate to initialize the iterative scheme. Recall that the parameters for optimization consist of $\boldsymbol{\mu}^q_{\beta_j}$, $\Sigma^q_{\beta_j}$, $\boldsymbol{\mu}^q_{\alpha_j}$, $\Sigma^q_{\alpha_j}$, $\boldsymbol{\mu}^q_{\gamma}$ and q_{ij} for $i = 1, \ldots, n$, $j = 1, \ldots, k$. Suppose that two components j_1 and j_2 are to be merged into a new component j'. The initial values for the new merging model are assigned as follows. For the initial values of the new component j', we set

$$\boldsymbol{\mu}^q_{\beta_{j'}} = \frac{\bar{q}_{\cdot j_1}\boldsymbol{\mu}^q_{\beta_{j_1}} + \bar{q}_{\cdot j_2}\boldsymbol{\mu}^q_{\beta_{j_2}}}{\bar{q}_{\cdot j_1} + \bar{q}_{\cdot j_2}}, \quad \Sigma^q_{\beta_{j'}} = \frac{\bar{q}_{\cdot j_1}\Sigma^q_{\beta_{j_1}} + \bar{q}_{\cdot j_2}\Sigma^q_{\beta_{j_2}}}{\bar{q}_{\cdot j_1} + \bar{q}_{\cdot j_2}},$$
$$\boldsymbol{\mu}^q_{\alpha_{j'}} = \frac{\bar{q}_{\cdot j_1}\boldsymbol{\mu}^q_{\alpha_{j_1}} + \bar{q}_{\cdot j_2}\boldsymbol{\mu}^q_{\alpha_{j_2}}}{\bar{q}_{\cdot j_1} + \bar{q}_{\cdot j_2}}, \quad \Sigma^q_{\alpha_{j'}} = \frac{\bar{q}_{\cdot j_1}\Sigma^q_{\alpha_{j_1}} + \bar{q}_{\cdot j_2}\Sigma^q_{\alpha_{j_2}}}{\bar{q}_{\cdot j_1} + \bar{q}_{\cdot j_2}},$$

with $\bar{q}_{\cdot j} = \frac{1}{n}\sum_{i=1}^n q_{ij}$ and $q_{ij'} = q_{ij_1} + q_{ij_2}$. The initial values for the parameters in the other components are fixed at the current estimate. Now the iterative scheme in Algorithm 1 can be readily performed to estimate the parameters of the new model. Note that the number of components now is reduced by 1.

Split criterion A component is considered unreliable and a plausible split candidate if it has a small likelihood, i.e. it is poorly fitted and should be split. The reliability of component j is defined as

$$R(j) = \frac{1}{n}\sum_{i=1}^{n}\log\hat{p}_j(\boldsymbol{x}_i) = \frac{1}{n}\sum_{i=1}^{n}\left(-\frac{1}{2}\log(2\pi) - \frac{1}{2}\boldsymbol{w}_i'\boldsymbol{\mu}^q_{\alpha_j} - \frac{(y_i - \boldsymbol{v}_i'\boldsymbol{\mu}^q_{\beta_j})^2}{2\exp(\boldsymbol{w}_i'\boldsymbol{\mu}^q_{\alpha_j})}\right),$$

where $\hat{p}_j(\boldsymbol{x})$ is the density function of the jth component. The smaller the $R(j)$, the less reliable is component j and the more plausible it is as a candidate for a split. Let $\eta_1 = \text{argmin}\{R(j),\ j = 1, \ldots, k\}$, $\eta_i = \text{argmin}\{R(j),\ j = 1, \ldots, k;\ j \neq \eta_1, \ldots, \eta_{i-1}\}$, $i \geq 2$. Write $\mathcal{C}_{\text{split}} = \mathcal{C}_{\text{split}}(\boldsymbol{\theta}^*) = \{\eta_1, \eta_2, \ldots\}$. As in the merge step, we split the most plausible components until the lower bound is improved or the number of split operations exceeds a pre-specified number $C^{\max}_{\text{split}}$.

Split operation Denote the component to be split by j' and the new components by j_1 and j_2. We set $q_{ij_1} = q_{ij_2} = q_{ij'}/2$, $\boldsymbol{\mu}^q_{\alpha_{j_1}} = \boldsymbol{\mu}^q_{\alpha_{j_2}} = \boldsymbol{\mu}^q_{\alpha_{j'}}$, $\Sigma^q_{\alpha_{j_1}} = \Sigma^q_{\alpha_{j_2}} = \Sigma^q_{\alpha_{j'}}$, and keep the other q_{ij}, $\boldsymbol{\mu}^q_{\alpha_j}$ and $\Sigma^q_{\alpha_j}$ unchanged. Then we are able to perform the updates in Algorithm 1 for all parameters. The number of components is now increased by 1.

We now summarize our final algorithm for estimating the RDE-MHN model, which is able to deal with the local maxima problem, while automatically determining the number of components.

Algorithm 2: Split-and-merge variational approximation

1. Perform Algorithm 1. After convergence, denote the estimated parameters by $\boldsymbol{\theta}^*$ and the maximized lower bound by L^*.
2. Compute the sets $\mathcal{C}_{\text{merge}}(\boldsymbol{\theta}^*)$ and $\mathcal{C}_{\text{split}}(\boldsymbol{\theta}^*)$.

```
3. For $i_{\text{merge}} = 1 : C_{\text{merge}}^{\max}$ do
    • Merge two components with the index pair $\xi_{i_{\text{merge}}}$. Let $L^*_{\text{merge}}, \boldsymbol{\theta}^*_{\text{merge}}$ be the new lower bound and parameter estimate.
    • If $L^*_{\text{merge}} > L^*$ halt the for loop.
4. For $i_{\text{split}} = 1 : C_{\text{split}}^{\max}$ do
    • Split the components $\eta_{i_{\text{split}}}$. Let $L^*_{\text{split}}, \boldsymbol{\theta}^*_{\text{split}}$ be the new lower bound and parameter estimate.
    • If $L^*_{\text{split}} > L^*$ halt the for loop.
5. If $L^*_{\text{merge}} > \max\{L^*_{\text{split}}, L^*\}$ then $L^* := L^*_{\text{merge}}$, $\boldsymbol{\theta}^* := \boldsymbol{\theta}^*_{\text{merge}}$ and go back to Step 2.
   If $L^*_{\text{split}} > \max\{L^*_{\text{merge}}, L^*\}$ then $L^* := L^*_{\text{split}}$, $\boldsymbol{\theta}^* := \boldsymbol{\theta}^*_{\text{split}}$ and go back to Step 2.
```

4. Model selection

This section considers the problem of selecting significant covariates out of s given potential covariates $x_1, \ldots, x_s$ for inclusion in the mean, variance and gating models. Write $C = \{1, \ldots, s\}$ for the index set of the potential covariates. Before presenting our strategy for ranking variables for inclusion, we discuss the model prior. Let $\pi_i^m, \ \pi_i^v, \ \pi_i^g$ be the prior probabilities for inclusion of covariate x_i in the mean, variance and gating models, respectively, and write $\boldsymbol{\pi}^m = (\pi_1^m, \ldots, \pi_s^m)'$, $\boldsymbol{\pi}^v = (\pi_1^v, \ldots, \pi_s^v)'$, $\boldsymbol{\pi}^g = (\pi_1^g, \ldots, \pi_s^g)'$. Suppose that we have a current model $\mathcal{M}$ with $C_m, \ C_v, \ C_g$ the index sets of covariates in its mean, variance and gating models, respectively. We assume

$$\begin{aligned} p(C_m|\boldsymbol{\pi}^m) &= \prod_{i \in C_m} \pi_i^m \prod_{i \notin C_m} (1 - \pi_i^m), \\ p(C_v|\boldsymbol{\pi}^v) &= \prod_{i \in C_v} \pi_i^v \prod_{i \notin C_v} (1 - \pi_i^v), \\ p(C_g|\boldsymbol{\pi}^g) &= \prod_{i \in C_g} \pi_i^g \prod_{i \notin C_g} (1 - \pi_i^g), \end{aligned}$$

and that

$$p(\mathcal{M}) = p(C_m, C_v, C_g|\boldsymbol{\pi}^m, \boldsymbol{\pi}^v, \boldsymbol{\pi}^g) = p(C_m|\boldsymbol{\pi}^m)p(C_v|\boldsymbol{\pi}^v)p(C_g|\boldsymbol{\pi}^g). \tag{4.1}$$

If no such detailed prior information is available on the inclusion probability for each predictor (which is the case we consider in this paper), one may assume that $\pi_1^m = \cdots = \pi_s^m = \pi_m$, $\pi_1^v = \cdots = \pi_s^v = \pi_v$ and $\pi_1^g = \cdots = \pi_s^g = \pi_g$ (we note a slight abuse of notation here), then

$$\begin{aligned} p(C_m|\pi_m) &= \pi_m^{|C_m|}(1 - \pi_m)^{s-|C_m|}, \\ p(C_v|\pi_v) &= \pi_v^{|C_v|}(1 - \pi_v)^{s-|C_v|}, \\ p(C_g|\pi_g) &= \pi_g^{|C_g|}(1 - \pi_g)^{s-|C_g|}, \end{aligned} \tag{4.2}$$

where for a set A, $|A|$ denotes its cardinality. The hyperparameters $\pi_m,\ \pi_v,\ \pi_g \in [0,1]$ are user-specified, and small values encourage parsimonious models. By setting $\pi_m = \pi_v = \pi_g = 1/2$, one can set the uniform prior on the inclusions of covariates. Another option is to put uniform distributions on the π's. Then

$$p(C_m) = \int_0^1 p(C_m|\pi_m)d\pi_m \propto \binom{s}{|C_m|}^{-1}, \tag{4.3}$$

and similarly $p(C_v) \propto \binom{s}{|C_v|}^{-1}$, $p(C_g) \propto \binom{s}{|C_g|}^{-1}$. This prior agrees with the one used in the extended BIC proposed by [6]. It has the advantage of requiring no hyperparameter while still encouraging parsimony. We recommend using this as the default prior.

We now consider adding a single variable in either the mean, variance or gating model, and then a one step update to the current variational lower bound in the proposed model as a computationally thrifty way of ranking the predictors for their possible inclusion. Write $X_{C_m}, X_{C_v}, X_{C_g}$ for the corresponding design matrices and in particular $\boldsymbol{x}'_{iC_m}$ for the ith row of X_{C_m}; $\boldsymbol{\beta}_j^{C_m}, \boldsymbol{\alpha}_j^{C_v}, \boldsymbol{\gamma}_j^{C_g}$ for the current coefficient vectors in the mean, variance and gating models of the jth component; $\beta_j^l, \alpha_j^l, \gamma_j^l$ for the new coefficients with respect to a new covariate x_l in the jth component.

Our method for ranking covariates for inclusion in the mean and variance models is an extension of the ranking algorithm proposed in [20] who considered the RDE-MHN(1) model only. The idea of ranking covariates for inclusion in the gating model is a non-trivial contribution of the present paper.

4.1. Ranking covariates in the mean model

We first consider the effect of adding a new covariate x_l with $l \notin C_m$ to the mean model of the k components. It is also possible to consider adding different covariates to different components, but this complicates the model somewhat. We consider a variational approximation to the posterior of the form

$$q(\boldsymbol{\theta}) = q(\boldsymbol{\delta})q(\boldsymbol{\alpha})q(\boldsymbol{\gamma})\prod_{j=1}^k q(\boldsymbol{\beta}_j^{C_m})q(\beta_j^l),$$

with $q(\boldsymbol{\beta}_j^{C_m}) \sim N(\boldsymbol{\mu}^q_{\beta_j^{C_m}}, \Sigma^q_{\beta_j^{C_m}})$ and $q(\beta_j^l) \sim N(\mu^q_{\beta_j^l}, (\sigma^q_{\beta_j^l})^2)$. From (A.1) in Appendix A, it is easy to see that the lower bound of the new model with x_l in the mean model (of all components) can be written as

$$\begin{aligned} L_{\text{new}} \quad = \quad & L_{\text{old}} + \frac{1}{2}\sum_{j=1}^k \left\{ 1 + \log\frac{(\sigma^q_{\beta_j^l})^2}{(\sigma^0_{\beta_j})^2} - \frac{(\sigma^q_{\beta_j^l})^2}{(\sigma^0_{\beta_j})^2} - \frac{(\mu^q_{\beta_j^l})^2}{(\sigma^0_{\beta_j})^2} - \right. \\ & \left. - \sum_{i=1}^n q_{ij}\frac{x_{il}^2(\sigma^q_{\beta_j^l})^2 + x_{il}^2(\mu^q_{\beta_j^l})^2 - 2x_{il}\mu^q_{\beta_j^l}(y_i - \boldsymbol{x}'_{iC_m}\boldsymbol{\mu}^q_{\beta_j^{C_m}})}{\exp(\boldsymbol{x}'_{iC_v}\boldsymbol{\mu}^q_{\alpha_j^{C_v}} - \frac{1}{2}\boldsymbol{x}'_{iC_v}\Sigma^q_{\alpha_j^{C_v}}\boldsymbol{x}_{iC_v})} \right\}, \end{aligned} \tag{4.4}$$

where L_{old} is the lower bound of the current model (i.e., the model without covariate x_l in its mean model) and $\boldsymbol{\mu}^q_{\alpha_j^{C_v}}$, $\Sigma^q_{\alpha_j^{C_v}}$ are parameters in the variational posterior of the variance model. We maximize the lower bound (4.4) with respect to new parameters $\mu^q_{\beta^l_j}$, $(\sigma^q_{\beta^l_j})^2$ when the others are fixed at the current fit. Writing $\hat{\mu}^q_{\beta^l_j}$, $(\hat{\sigma}^q_{\beta^l_j})^2$ for the optimizers, we have that

$$(\hat{\sigma}^q_{\beta^l_j})^2 = \left(\frac{1}{(\sigma^0_{\beta_j})^2} + \sum_{i=1}^n q_{ij} \frac{x_{il}^2}{\exp(\boldsymbol{x}'_{iC_v}\boldsymbol{\mu}^q_{\alpha_j^{C_v}} - \frac{1}{2}\boldsymbol{x}'_{iC_v}\Sigma^q_{\alpha_j^{C_v}}\boldsymbol{x}_{iC_v})} \right)^{-1}$$

and

$$\hat{\mu}^q_{\beta^l_j} = (\hat{\sigma}^q_{\beta^l_j})^2 \sum_{i=1}^n q_{ij} \frac{x_{il}(y_i - \boldsymbol{x}'_{iC_m}\boldsymbol{\mu}^q_{\beta_j^{C_m}})}{\exp(\boldsymbol{x}'_{iC_v}\boldsymbol{\mu}^q_{\alpha_j^{C_v}} - \frac{1}{2}\boldsymbol{x}'_{iC_v}\Sigma^q_{\alpha_j^{C_v}}\boldsymbol{x}_{iC_v})}, \quad j = 1, \ldots, k.$$

Substituting these back to the lower bound (4.4) and writing $L_l^M(C_m, C_v, C_g)$ for the optimized lower bound gives

$$L_l^M(C_m, C_v, C_g) = L_{\text{old}} + \frac{1}{2}\sum_{j=1}^k \left(\log \frac{(\hat{\sigma}^q_{\beta^l_j})^2}{(\sigma^0_{\beta_j})^2} + \frac{(\hat{\mu}^q_{\beta^l_j})^2}{(\hat{\sigma}^q_{\beta^l_j})^2} \right). \tag{4.5}$$

The superscript M means the lower bound is associated with the mean model. The most plausible variable for inclusion in the mean model is the one that maximizes the above lower bound (over $l \in C \setminus C_m$). Note that we only use (4.5) for ranking covariates for inclusion, the actual inclusion is based on the improvement of the lower bound fitted using the full variational algorithm described in the previous section.

In the full variational approximation fit, the "naive" estimates $\hat{\mu}^q_{\beta^l_j}$, $(\hat{\sigma}^q_{\beta^l_j})^2$ can be used to initialize the new parameters with respect to new covariate x_l in the mean model, while all the other parameters can be initialized by the current estimates. This so-called *warm start* (i.e., the output of a previous fit is used for initial values in a subsequent fit) is very important in variational approximation, especially in the complex context of mixture models, and makes the variational approximation procedure more stable and faster [20]. We observe that, with the warm start, Algorithm 1 converges very quickly, often after just a few iterations.

4.2. Ranking covariates in the variance model

We now consider adding a new covariate x_l with $l \notin C_v$ to the variance model. As in the mean model, we consider a variational approximation to the posterior of the form

$$q(\boldsymbol{\theta}) = q(\boldsymbol{\delta})q(\boldsymbol{\beta})q(\boldsymbol{\gamma})\prod_{j=1}^k q(\boldsymbol{\alpha}_j^{C_v})q(\alpha_j^l),$$

where $q(\boldsymbol{\alpha}_j^{C_v}) \sim N(\boldsymbol{\mu}^q_{\alpha_j^{C_v}}, \Sigma^q_{\alpha_j^{C_v}})$ and $q(\alpha_j^l) \sim N(\mu^q_{\alpha_j^l}, (\sigma^q_{\alpha_j^l})^2)$. If we do not assume a particular parametric form for $q(\alpha_j^l)$, the optimal choice is [see, for example, 21]

$$\begin{aligned} q_{\text{opt}}(\alpha_j^l) &\propto \exp\big(E[\log p(\alpha_j^l) + \log p(\boldsymbol{y}|\boldsymbol{\theta})]\big) \\ &\propto \exp\Bigg(-\frac{1}{2}\log(\sigma^0_{\alpha_j})^2 - \frac{(\alpha_j^l)^2}{(\sigma^0_{\alpha_j})^2} - \frac{1}{2}\sum_{i=1}^n q_{ij}\boldsymbol{x}'_{iC_v}\boldsymbol{\mu}^q_{\alpha_j^{C_v}} - \frac{1}{2}\sum_{i=1}^n q_{ij}x_{il}\alpha_j^l \\ &\qquad - \frac{1}{2}\sum_{i=1}^n q_{ij}\frac{(y_i - \boldsymbol{x}'_{iC_m}\boldsymbol{\mu}^q_{\beta_j^{C_m}})^2 + \boldsymbol{x}'_{iC_m}\Sigma^q_{\beta_j^{C_m}}\boldsymbol{x}_{iC_m}}{\exp(\boldsymbol{x}'_{iC_v}\boldsymbol{\mu}^q_{\alpha_j^{C_v}} + x_{il}\alpha_j^l - \frac{1}{2}\boldsymbol{x}'_{iC_v}\Sigma^q_{\alpha_j^{C_v}}\boldsymbol{x}_{iC_v})}\Bigg), \end{aligned}$$

where the expectation is with respect to all parameters except α_j^l, $j = 1, \ldots, k$. Therefore, to obtain good estimates for $\mu^q_{\alpha_j^l}$ and $(\sigma^q_{\alpha_j^l})^2$, we make a normal approximation to $q_{\text{opt}}(\alpha_j^l)$. Then the mean and variance of the normal approximation are the mode of $q_{\text{opt}}(\alpha_j^l)$ and the negative inverse Hessian at the mode of log $q_{\text{opt}}(\alpha_j^l)$, respectively. We have

$$\begin{aligned} \hat{\mu}^q_{\alpha_j^l} &= \frac{1}{2}\sum_{i=1}^n q_{ij}x_{il}(v_i - 1)\Bigg/\left(\frac{1}{(\sigma^0_{\alpha_j})^2} + \frac{1}{2}\sum_{i=1}^n q_{ij}x_{il}^2 v_{ij}\right), \\ (\hat{\sigma}^q_{\alpha_j^l})^2 &= \left(\frac{1}{(\sigma^0_{\alpha_j})^2} + \frac{1}{2}\sum_{i=1}^n \frac{q_{ij}x_{il}^2 v_i}{\exp(x_{il}\hat{\mu}^q_{\alpha_j^l})}\right)^{-1}, \end{aligned}$$

where

$$v_{ij} = \frac{(y_i - \boldsymbol{x}'_{iC_m}\boldsymbol{\mu}^q_{\beta_j^{C_m}})^2 + \boldsymbol{x}'_{iC_m}\Sigma^q_{\beta_j^{C_m}}\boldsymbol{x}_{iC_m}}{\exp(\boldsymbol{x}'_{iC_v}\boldsymbol{\mu}^q_{\alpha_j^{C_v}} - \frac{1}{2}\boldsymbol{x}'_{iC_v}\Sigma^q_{\alpha_j^{C_v}}\boldsymbol{x}_{iC_v})}, \quad j = 1, \ldots, k, \ i = 1, \ldots, n.$$

To obtain a more accurate estimate of the mode, in our implementation, we use Newton's method initialized with $\hat{\mu}^q_{\alpha_j^l}$. Note that Newton's method is very convenient here because the second derivative is available in closed form. We found that $\hat{\mu}^q_{\alpha_j^l}$ is a very good approximation and the Newton iteration often stops after a few iterations.

From (A.1), the new lower bound can be written as

$$\begin{aligned} &L_l^V(C_m, C_v, C_g) \\ &\quad = \text{const} + \frac{1}{2}\sum_{j=1}^k \Bigg\{\log\frac{(\hat{\sigma}^q_{\alpha_j^l})^2}{(\sigma^0_{\alpha_j})^2} - \frac{(\hat{\sigma}^q_{\alpha_j^l})^2}{(\sigma^0_{\alpha_j})^2} - \frac{(\hat{\mu}^q_{\alpha_j^l})^2}{(\sigma^0_{\alpha_j})^2} - \sum_{i=1}^n q_{ij}x_{il}\hat{\mu}^q_{\alpha_j^l} \\ &\qquad - \sum_{i=1}^n q_{ij}\frac{(y_i - \boldsymbol{x}'_{iC_m}\boldsymbol{\mu}^q_{\beta_j^{C_m}})^2 + \boldsymbol{x}'_{iC_m}\Sigma^q_{\beta_j^{C_m}}\boldsymbol{x}_{iC_m}}{\exp(\boldsymbol{x}'_{iC_v}\boldsymbol{\mu}^q_{\alpha_j^{C_v}} - \frac{1}{2}\boldsymbol{x}'_{iC_v}\Sigma^q_{\alpha_j^{C_v}}\boldsymbol{x}_{iC_v} + x_{il}\hat{\mu}^q_{\alpha_j^l} - \frac{1}{2}x_{il}^2(\hat{\sigma}^q_{\alpha_j^l})^2)}\Bigg\}, \end{aligned} \tag{4.6}$$

where the constant term does not depend on the new parameters with respect to x_l, i.e., $\hat{\mu}^q_{\alpha^l_j}$ and $(\hat{\sigma}^q_{\alpha^l_j})^2$. The superscript V means the lower bound is associated with the variance model. As in the mean model, we only use (4.6) for ranking covariates for inclusion and the actual inclusion is based on the full variational approximation fit. Also, the estimates $\hat{\mu}^q_{\alpha^l_j}$ and $(\hat{\sigma}^q_{\alpha^l_j})^2$ can be used to create a warm start in the full variational approximation fit.

4.3. Ranking covariates in the gating model

Using lower bounds for ranking covariates for inclusion in the gating model is more complex. We proceed by using a measure of association between the response and the covariates for ranking for inclusion. The measure we use is the *distance correlation* introduced recently by [26]. The distance correlation is a measure of general, not just linear, dependence between two random vectors of arbitrary dimensions and has the property that it is zero if and only if the two random vectors are independent. This new definition of correlation measure is very interesting, useful and widely applicable. The reader is referred to [26] for the definition and properties of this association measure.

Let $x_{\hat{l}}$ be the covariate that has highest (sample) distance correlation with y among x_l with $l \in C \setminus C_g$. If the covariate space of the gating model is completely separate from that of the mean and variance models, then this high correlation of $x_{\hat{l}}$ suggests that $x_{\hat{l}}$ is the most plausible covariate to be added to the gating model, and it will be selected if its inclusion to the gating model improves the lower bound. The situation is more complex if the covariate spaces of the three models overlap or are the same, which is the case we consider in this paper. In this situation care must be taken in considering the inclusion of $x_{\hat{l}}$. We proceed as follows. If $x_{\hat{l}}$ has not been included in the mean or variance model, i.e. $\hat{l} \notin C_m \cup C_v$, then $x_{\hat{l}}$ will be added to the gating model if its inclusion improves the lower bound. Suppose that $x_{\hat{l}}$ has already been included in the mean or variance model. It will be added to the gating model as well if its inclusion improves the lower bound. Otherwise, the high correlation of $x_{\hat{l}}$ is likely to be caused via other models rather than the gating model, therefore we consider for inclusion the covariate $x_{l'}$ which has second highest distance correlation with y among the covariates in $C \setminus C_g$. This consideration for inclusion is repeated until a covariate is selected or no $x_{l'}$ exists.

We give below pseudo-code for the variable selection strategy. Let $\tau_l = \text{dCor}(y, x_l)$ be the sample distance correlation, i.e. the distance correlation calculated from the data, between the response y and covariate x_l [see 26]. Write $\boldsymbol{\tau} = (\tau_1, \dots, \tau_p)'$.

```
Set $\tau_l = -\infty$ if $l \in C_g$, $l = 1, \dots, s$
stop=FALSE
**While** not stop **do**
    $\hat{l} = \text{argmax}\{\tau_l,\ l = 1, \dots,\ s\}$
    If $\tau_{\hat{l}} = -\infty$ then stop=TRUE
    else
      • If $\hat{l} \notin C_m \cup C_v$: set stop=TRUE, add $x_{\hat{l}}$ to the gating model if its
        inclusion improves the lower bound.
      • If $\hat{l} \in C_m \cup C_v$: if inclusion of $x_{\hat{l}}$ improves the lower bound then add $x_{\hat{l}}$
        to the gating model and set stop = TRUE, otherwise set $\tau_{\hat{l}} = -\infty$.
    end else
  end while
```

Note that it is still possible to obtain a warm start in the variational approximation fit. To initialize the variational approximation fit in Algorithm 1, we start with Step 6 to update $\boldsymbol{\mu}_\gamma^q$, while all other parameters are initialized by the current estimates.

4.4. The full algorithm for variable and component selection

We now summarize our ranking algorithm for variable selection combined with the split-and-merge variational approximation for component selection. We denote the algorithm by RSMVA. Recall that we denote by $C_m,\ C_v,\ C_g$ the index sets of current covariates in the mean, variance and gating models, respectively. We write $L(C_m, C_v, C_g)$ for the lower bound optimized by the full variational approximation fit procedure described in Section 3. Write C_m^{+l} for the set $C_m \cup \{l\}$ and similarly for C_v and C_g. Denote by $p(C_m, C_v, C_g)$ the prior of the model with index sets C_m, C_v, C_g. Note that for simplicity of discussion in Sections 4.1-4.3, we did not mention the model prior.

Algorithm 3: RSMVA algorithm for variable selection and component selection

```
1. Initialize $C_m,\ C_v,\ C_g$ and set $L_{\text{opt}} := L(C_m, C_v, C_g)$.
2. Repeat the following steps until stop
   (a) Store $C_m^{\text{old}} := C_m$, $C_v^{\text{old}} := C_v$ and $C_g^{\text{old}} := C_g$
   (b) Let $\hat{l} = \arg\max_{l \in C \setminus C_m} \{L_l^M(C_m, C_v, C_g)\}$. If

          $$L(C_m^{+\hat{l}}, C_v, C_g) + p(C_m^{+\hat{l}}, C_v, C_g) > L_{\text{opt}} + p(C_m, C_v, C_g)$$

       then set $C_m := C_m^{+\hat{l}}$, $L_{\text{opt}} = L(C_m^{+\hat{l}}, C_v, C_g)$.
   (c) Let $\hat{l} = \arg\max_{l \in C \setminus C_v} \{L_l^V(C_m, C_v, C_g)\}$. If

          $$L(C_m, C_v^{+\hat{l}}, C_g) + p(C_m, C_v^{+\hat{l}}, C_g) > L_{\text{opt}} + p(C_m, C_v, C_g)$$

       then set $C_v := C_v^{+\hat{l}}$, $L_{\text{opt}} = L(C_m, V_{+\hat{l}}, C_g)$.
   (d) Set $\tau_l = \text{dCor}(y, x_l)$ if $l \notin C_g$ else $\tau_l = -\infty$, $l = 1, \ldots, s$, and set
       stop=FALSE.
       While not stop do
```

```
        $\hat{l} = \mathrm{argmax}\{\tau_l,\ l = 1, \ldots,\ s\}$
        **If** $\tau_{\hat{l}} = -\infty$ **then** stop=TRUE
        **else**
          • If $\hat{l} \notin C_m \cup C_v$: stop=TRUE. If
                $L(C_m, C_v, C_g^{+\hat{l}}) + p(C_m, C_v, C_g^{+\hat{l}}) > L_{\text{opt}} + p(C_m, C_v, C_g)$
            then set $C_g := C_g^{+\hat{l}}$, $L_{\text{opt}} := L(C_m, C_v, C_g^{+\hat{l}})$.
          • If $\hat{l} \in C_m \cup C_v$: If
                $L(C_m, C_v, C_g^{+\hat{l}}) + p(C_m, C_v, C_g^{+\hat{l}}) > L_{\text{opt}} + p(C_m, C_v, C_g)$
            then set $C_g := C_g^{+\hat{l}}$, $L_{\text{opt}} := L(C_m, C_v, C_g^{+\hat{l}})$ and stop=TRUE,
            otherwise set $\tau_{\hat{l}} = -\infty$.
        **end else**
      **end while**
  (e) If ($C_m = C_m^{\text{old}}$ and $C_v = C_v^{\text{old}}$ and $C_g = C_g^{\text{old}}$) then stop, else return
      to (a).
```

This RSMVA algorithm was implemented in `R` and the code is available upon contacting the authors.

Remarks The RSMVA algorithm does not consider models of all sizes; it stops when important covariates have been included, so that the computations just involve low-dimensional matrices. This makes our method work in high-dimensional problems in which, amongst a large number of potential covariates, only a few are significant. Note that in such situations, the full model fitting of [19] and the MCMC approach of [28] are very time demanding if not impossible. In the simulation studies below, we consider an example with 1000 potential predictors.

When $k = 1$, the RSMVA reduces to the ranking algorithm for variable selection in heteroscedastic linear regression proposed in [20]. Then, as noted by [20], the ranking algorithm is similar to frequentist matching pursuit and greedy algorithms [16, 33, 8], while having many good properties such as not requiring any extra tuning parameter and not penalizing non-zero coefficients for doing variable selection. The reader is referred to [20], Section 3.5, for more details.

5. Simulation study

This section considers a simulation study of our method. The data is simulated from the following regression density model

$$y_i = \sum_{j=1}^{3} \pi_j(\boldsymbol{x}_i) N(\boldsymbol{x}_i' \boldsymbol{\beta}_j, \exp(\boldsymbol{x}_i' \boldsymbol{\alpha}_j)), \quad \pi_j(\boldsymbol{x}_i) = \frac{\exp(\boldsymbol{x}_i' \boldsymbol{\gamma}_j)}{\sum_{l=1}^{3} \exp(\boldsymbol{x}_i' \boldsymbol{\gamma}_l)}, \quad i = 1, \ldots, n,$$

Table 1
True parameter vectors $\boldsymbol{\beta}_j,\ \boldsymbol{\alpha}_j,\ \boldsymbol{\gamma}_j$

$\boldsymbol{\beta}_1$	$\boldsymbol{\beta}_2$	$\boldsymbol{\beta}_3$	$\boldsymbol{\alpha}_1$	$\boldsymbol{\alpha}_2$	$\boldsymbol{\alpha}_3$	$\boldsymbol{\gamma}_1$	$\boldsymbol{\gamma}_2$	$\boldsymbol{\gamma}_3$
5	2	-5	-2	-1	-1	0	1.5	1
-2	-4	3	2	-3	2	0	1	-3.5
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
4	2	-4	-1	3	-3	0	-4	1.5

Table 2
Easy example: Correctly fitted rates (%) for variable selection in the mean, variance and gating models, for component selection and for the overall model

s	n	Mean	Variance	Gating	Component	Overall
5	500	100	90	80	96	76
	1000	100	100	92	100	90
100	500	100	60	76	90	60
	1000	100	90	86	100	84

where $\boldsymbol{\beta}_j,\ \boldsymbol{\alpha}_j,\ \boldsymbol{\gamma}_j$ are in Table 1. The predictors $\boldsymbol{x}_i = (1, \tilde{\boldsymbol{x}}_i)'$ are generated by first generating $\tilde{\boldsymbol{x}}_i$ from the normal distribution $N(0, \Sigma)$ (with Σ specified below) and then transforming each component into the unit interval by the cumulative distribution function $\Phi(\cdot)$ of the standard normal. The reason for making the transformation is to control the magnitude of the noise levels $\exp(\boldsymbol{x}_i'\boldsymbol{\alpha}_j)$ and mixing coefficients $\pi_j(\boldsymbol{x}_i)$. It is natural to always include intercepts in the three models. The performance is measured by correctly-fitted rates for variable selection in the three models, for the selection of the number of components and for overall model selection, over 50 replications. That is, the correctly-fitted rate for a model is the proportion of replications that the true model is correctly identified.

An easy example We first consider an easy problem in which the covariates have small correlations by setting $\Sigma_{ij} = 0.5^{|i-j|}$. A low-dimensional case with $s = 5$ and a higher-dimensional case with $s = 100$ are investigated. For the latter, the first five entries of $\boldsymbol{\beta}_j,\ \boldsymbol{\alpha}_j,\ \boldsymbol{\gamma}_j$ are the same as in Table 1, while the rest are all zeros. Note that the full model fitting of [19] and the MCMC approach of [28] are almost impossible with $s = 100$ and a full model fitting would give a very poor fit because all of the irrelevant covariates are included.

Table 2 summarizes the simulation results for two cases, $n = 500$ and $n = 1000$. The results suggest that the RSMVA algorithm is able to correctly identify the zero-coefficients in the mean, variance and gating models, the true number of components as well as the true overall model. On average, the CPU time taken for each replication is 4.5 minutes for $n = 500$ and 10.1 minutes for $n = 1000$ in the low-dimensional case, and 5.9 and 18.7 minutes for $n = 500$ and $n = 1000$ respectively in the higher-dimensional case. The code is written in the `R` language and run on an Intel Core i7-2600 3.40GHz desktop.

We also consider an example in which the data is generated from a simple homoscedastic linear regression model, i.e. from a RDE-MHN(1) model with

Table 3

Classical regression: Correctly fitted rates (%) for variable selection in the mean and variance models, for component selection and for the overall model.

n	σ	Mean	Variance	Component	Overall
500	1	86	80	94	80
	2	70	66	90	64
1000	1	100	96	100	96
	2	100	90	100	90

Table 4

Harder example: Correctly fitted rates (%) for variable selection in the mean, variance and gating models, for component selection and for the overall model.

s	n	Mean	Variance	Gating	Component	Overall
5	500	92	50	42	96	42
	1000	100	80	72	100	68
100	500	100	42	56	86	40
	1000	100	82	82	100	80

Table 5

Small-n large-s example: Correctly fitted rates (%) for variable selection in the mean, variance and gating models, for component selection and for the overall model

	Mean	Variance	Gating	Component	Overall
Easy case	96	66	76	90	64
Harder case	96	42	68	88	42

only an intercept in the variance model. The data is simulated from

$$y_i \sim N(\boldsymbol{x}_i'\boldsymbol{\beta}, \sigma^2), \quad i = 1, \dots, n,$$

with $\boldsymbol{\beta}$ as $\boldsymbol{\beta}_1$ above. The correctly-fitted rates are summarized in Table 3 for four cases, $n = 500,\ 1000$ and $\sigma = 1,\ 2$. The CPU time is approximately 3.1 and 6.9 minutes for $n = 500$ and $n = 1000$ respectively for each replication. These results suggest that the RSMVA algorithm is able to reach a simple homoscedastic linear regression model if such a model is appropriate.

A harder example We now consider a harder example in which the covariates have high correlations by setting $\Sigma_{ij} = 0.9^{|i-j|}$. The simulation results are summarized in Table 4. The correctly-fitted rates for the variance and gating models are now smaller than those in Table 2. This is because fitting the variance and gating models is much harder than fitting the mean. However, as we observed, the algorithm misidentifies only one or two covariates with small mean squared errors in the coefficient estimation (results not shown).

A small-n large-s example We finally consider an example in which the number of potential covariates s is much larger than the number of observations n. We set $s = 1000$ and $n = 500$ and consider two cases as before: the easy case with $\Sigma_{ij} = 0.5^{|i-j|}$ and the harder case with $\Sigma_{ij} = 0.9^{|i-j|}$. The data generating process is similar to those in the previous examples. Table 5 summarizes the correctly-fitted rates and shows that the performance is very similar to the case with $s = 100$ and $n = 500$. This suggests that the large number of irrelevant covariates does not have much influence, they are easily dropped out by the ranking algorithm. The CPU time taken is about 51.7 minutes for each replication.

6. Applications

We first describe the log predictive density score considered by [28] and [19] to measure the performance of a model. Suppose we split the data $\boldsymbol{y}$ into two parts: future or validation set $\boldsymbol{y}_F$ and training set $\boldsymbol{y}_{\setminus F}$. The log predictive density score is defined by

$$\text{LPDS} = \log p(\boldsymbol{y}_F|\boldsymbol{y}_{\setminus F}), \tag{6.1}$$

where $p(\boldsymbol{y}_F|\boldsymbol{y}_{\setminus F}) = \int p(\boldsymbol{y}_F|\boldsymbol{\theta})p(\boldsymbol{\theta}|\boldsymbol{y}_{\setminus F})d\boldsymbol{\theta}$ can be approximated by Monte Carlo samples from the posterior $p(\boldsymbol{\theta}|\boldsymbol{y}_{\setminus F})$ which in turn can be replaced with the variational posterior $q(\boldsymbol{\theta})$. A simpler method to estimate $p(\boldsymbol{y}_F|\boldsymbol{y}_{\setminus F})$ is the plug-in method in which $p(\boldsymbol{y}_F|\boldsymbol{y}_{\setminus F})$ is estimated by $p(\boldsymbol{y}_F|\hat{\boldsymbol{\theta}}(\boldsymbol{y}_{\setminus F}))$ with $\hat{\boldsymbol{\theta}}(\boldsymbol{y}_{\setminus F})$ a point estimate, based on data $\boldsymbol{y}_{\setminus F}$, of the model parameters. We use the plug-in method in the following examples.

If the observations in $\boldsymbol{y}$ are exchangeable and if we can randomly split $\boldsymbol{y}$ into roughly B equal parts $F_1, \dots, F_B$, then the B-fold cross-validation log predictive density score is defined as

$$\text{LPDS} = \frac{1}{B}\sum_{i=1}^{B} \log p(\boldsymbol{y}_{F_i}|\boldsymbol{y}_{\setminus F_i}). \tag{6.2}$$

Note that for time series data $\boldsymbol{y}_{1:T}$, the cross-validation idea in definition (6.2) is not natural; however the log predictive density score in (6.1) can be still well-defined as follows. If $\boldsymbol{y}_{1:t}$ is the training set and $\boldsymbol{y}_{t+1:T}$ is the validation set, predictive performance is measured by

$$\text{LPDS} = \log p(\boldsymbol{y}_{t+1:T}|\boldsymbol{y}_{1:t}) = \sum_{i=1}^{T-t} \log p(y_{t+i}|\boldsymbol{y}_{1:t+i-1}), \tag{6.3}$$

with $p(y_{t+i}|\boldsymbol{y}_{1:t+i-1}) = \int p(y_{t+i}|\boldsymbol{\theta}, \boldsymbol{y}_{1:t+i-1})p(\boldsymbol{\theta}|\boldsymbol{y}_{1:t+i-1})d\boldsymbol{\theta}$.

A disadvantage of using (6.3) is that we lose some information contained in the validation sets in the learning processes. Furthermore, using (6.2) or (6.3) as a model selection criterion may be time consuming in some cases.

6.1. Diabetes data

We apply our method to analyze a benchmark data set on the progression of diabetes [8, 20]. Ten baseline variables, age, sex, body mass index, average blood pressure and six blood serum measurements, were obtained for each of $n = 442$ diabetes patients, as well as the response of interest y, a quantitative measure of disease progression one year after baseline. Previous analysis in the literature assumed a RDE-MHN(1) model [8, 20]. Our goal is to see if relaxing the assumption of $k = 1$ components gives us more flexibility in exploring the structure in the conditional distribution of y, and leads to a model with better predictive performance.

The RSMVA algorithm selects a RDE-MHN(3) model with homoscedastic components, only intercepts in the mean and variance models, and covariates 3 and 9 in the gating model. We call this model A. If we fix $k = 1$, the RSMVA algorithm (this is the VAR algorithm of Nott et al. 2011b) selects an intercept and covariates 2, 3, 7 and 9 to the mean model, and only an intercept to the variance model. We call this model B. The 10-fold cross-validation log predictive density score of model A is -236.7 and of model B is -241.3. This suggests that model A has better predictive performance than model B.

Figure 1 shows the fitted RDE-MHN model with 3 homoscedastic components. We have separated the observed responses into clusters according to which component each response is most likely to lie in. The planes show the fitted means which are 149.7, 72.4 and 259.7. The right column shows the fitted mixing probabilities. Figure 2 shows the plots of standardized residuals versus fitted values for the two models RDE-MHN(3) and RDE-MHN(1). The RDE-MHN(3) seems to give a more satisfying residual plot, because the absolute residuals of the RDE-MHN(1) model increase when the fitted values increase. These pictures tell us visually that the distribution of y is better modeled by a mixture of 3 components. The CPU time taken to run the RSMVA algorithm to get the final model in this example is 3.9 minutes.

6.2. Rainfall runoff emulation model

This application is concerned with model emulation of a deterministic rainfall runoff model. The goal of model emulation is to replace a computationally expensive deterministic model with a computationally cheap statistical model/emulator which may allow similar results to be achieved in a manageable computational time. The emulation model we consider is the Australian Water Balance Model (AWBM) of [3]. The reader is referred to [19] for a more detailed description. We are concerned with estimating the distribution of the AWBM streamflow at a time of peak rainfall (response y) as a function of three AWBM parameters/covariates: the maximum storage capacity `S` (x_1), the baseflow recession factor `K` (x_2) and the base flow index `BFI` (x_3). We have an available dataset obtained for 500 different values of parameters `S`,`K` and `BFI`. Following [19], we also add independent normal random noise with a standard deviation 0.01 to the response y to avoid degeneracies in the variance model in regions of the space where the response tends to be identically zero.

[19] emulated the AWBM streamflow response as a function of `S` and `K` (`BFI` was found insensitive to the response). They fitted five RDE-MHN models to the data. The first four models (named A, B, C and D) have $k = 2,\ 3,\ 4$ and 5 components respectively with both covariates (apart from an intercept) in

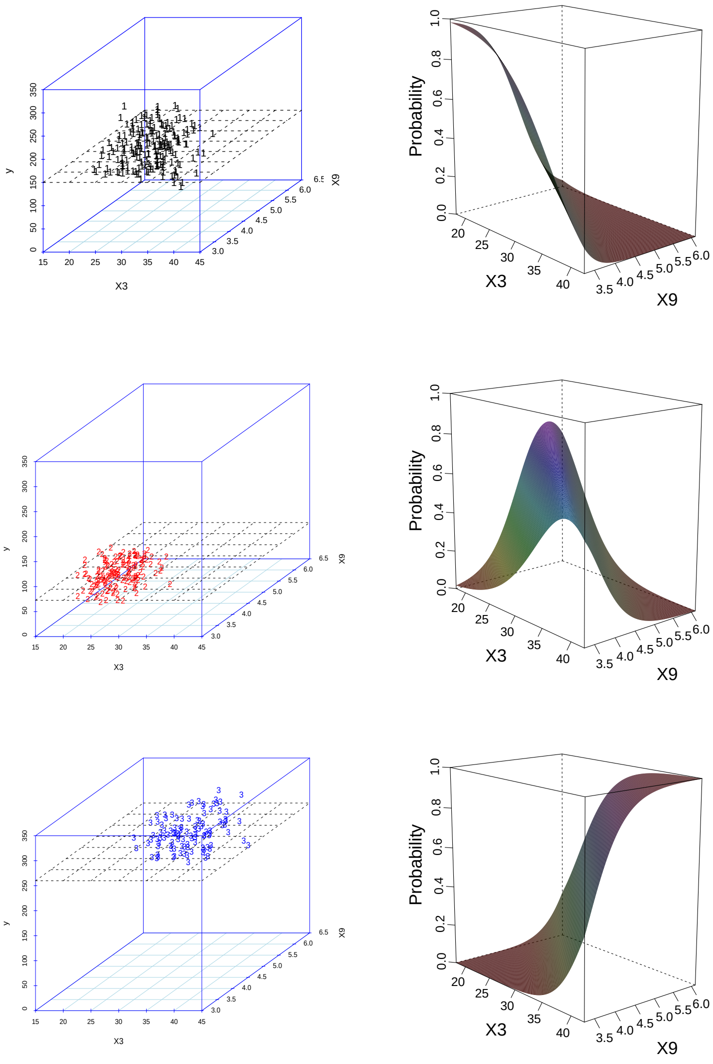

FIG 1. *Diabetes data: The left column shows the fitted component means as well as the clusters for 442 patients. The right shows the fitted mixing probabilities.*

the mean, variance and gating models. The fifth, model E, has $k = 4$ components with only an intercept in the variance model. Based on the 10-fold cross-validation log predictive density scores, [19] choose model C (among the five considered). The 10-fold cross-validation log predictive density score of model C is -57.4. This value is slightly different from the value reported in [19], due

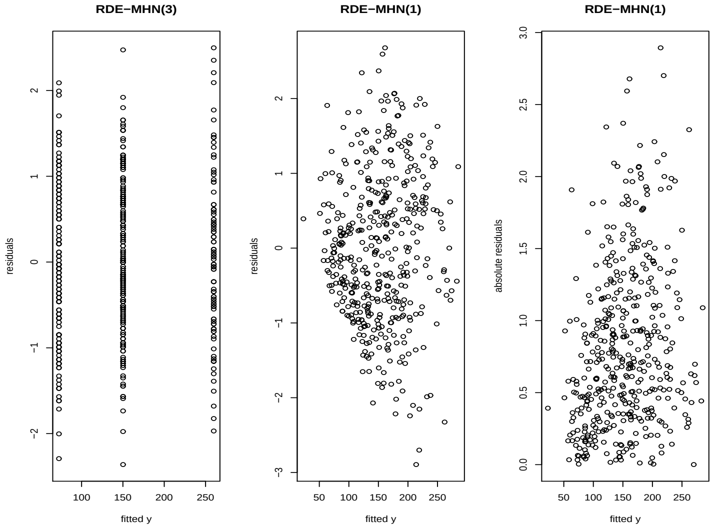

FIG 2. *Diabetes data: Plots of standardized residuals vs fitted responses. For the RDE-MHN(3) model, we first separate responses into clusters and use the fitted means for prediction.*

to the randomness in adding random noise to the responses, the difference in the use of priors and the randomness of the partition in the cross-validation.

If we fix the covariates for inclusion in the mean, variance and gating models as considered in the four models A-D above and let the split-and-merge variational approximation algorithm (Algorithm 2) determine the number of components, then $k = 4$ is selected, which is consistent with the finding of [19]. The CPU time taken is 1.6 minutes. In order to find the best model C, [19] compute the cross-validation log predictive density scores for the four models. As reported in their Table 2, the CPU times taken is 16.3 minutes for variational approximation and 5.8 hours for MCMC. Our method is roughly 10 and 216 times faster than the variational approximation and MCMC methods based on log predictive density scores. It should however be noted that these differences may be partly due to the different CPU's used to carry out the computations in the two papers.

We now consider the problem of variable selection and component selection simultaneously. The RSMVA algorithm then selects a RDE-MHN(4) model with both covariates in the mean and variance models and only covariate `S` in the gating model. The CPU time taken is 4.38 minutes. We call this model F, whose 10-fold cross-validation log predictive density score is −52.9. This suggests that model F has better predictive performance than model C. This also illustrates that variable selection helps improve on log predictive density scores. Figure 3 summarizes the fitted RDE-MHN(4) model. These figures tell us visually the structure of the data.

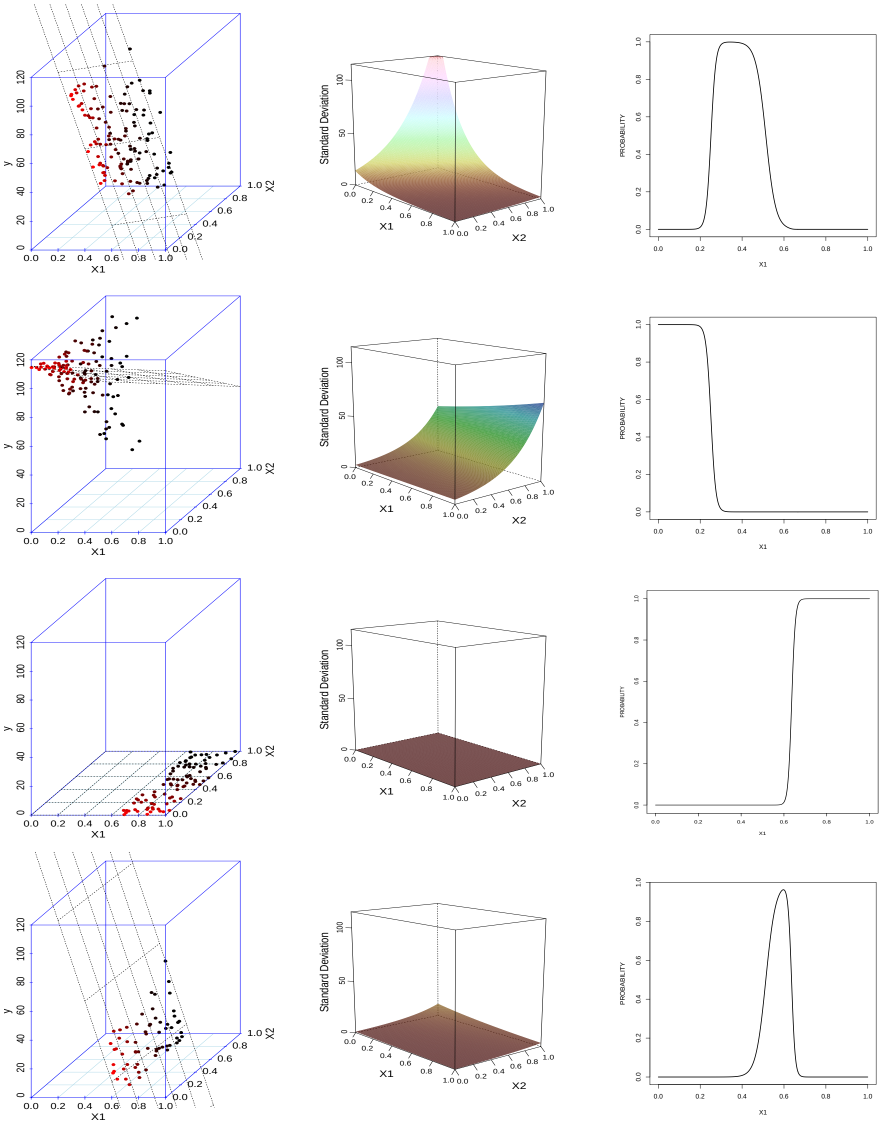

Fig 3. *Rainfall runoff model: The fitted component means (first column), standard deviations (second column) and mixing probabilities (last column).*

6.3. Standard & Poor's 500 index

We reanalyze the data set of returns to the Standard & Poor's 500 stock market index considered in [10], [28] and [19] who must rely on future observations to do model selection. This data consists of 4646 daily returns from January 1, 1990 to May 29, 2008. The response y_t is $\log(p_t/p_{t-1})$, with p_t the closing index on day t. The goal is to flexibly estimate the distribution of y_t given the data $y_{1:t-1}$ up to time $t-1$. Modeling $p(y_t|y_{1:t-1})$ is challenging because it is well known in the economics literature that the distribution of y_t is non-standard and has heavy tails. The distribution of y_t would therefore not be appropriately modeled by a simple parametric density such as a normal.

We can set up the density estimation problem of $p(y_t|y_{1:t-1})$ in terms of a regression density estimation problem where the covariates are functions of lagged response values. We refer the reader to [28] for a list and definition of potential covariates. [19] estimate the density function of y_t by four RDE-MHN models in which the mean model contains only an intercept, the variance and gating models contain an intercept and covariates `RLastWeek, RLastMonth` and `MaxMin95` and $k = 1$, 2, 3 and 4 experts. Using a validation data set consisting of 199 future observations, they report that the RDE-MHN(2) model is adequate. The CPU time taken by their approach (including times for initial fit and for computing the log predictive density scores) is 5.4 hours. Our split-and-merge variational approximation algorithm (Algorithm 2) gives the same result, with a CPU time of 9.5 minutes. Note that our method for selecting the number of experts k does not rely on future observations.

We now let the RSMVA algorithm itself select important variables as well as the number of components. As usual in the literature on stock market return data where a mean relation is not expected, we restrict the mean model to include only an intercept. Our algorithm then selects a RDE-MHN model similar to the model found adequate above, except that covariate `RLastWeek` drops out in the variance model. The CPU time is 2.1 hours. This model has a log predictive density score of -472.2, which has slightly less predictive performance than the model suggested by [19] with a log predictive density score of -471.1.

7. Conclusions

Our paper describes a split-and-merge variational approximation strategy for fitting the RDE-MHN model. The approach automatically determines the number of components and is able to overcome the local maxima problem in fitting mixture models. We also present a fast greedy algorithm for variable selection. The full algorithm, RSMVA, provides a computationally thrifty path following strategy for doing simultaneous variable selection, parameter estimation and component selection. The RSMVA is able to reach a simple model if such a model is warranted, and in the special case of $k = 1$ components, reduces to well-studied algorithms in the literature. The proposed methodology applies to high-dimensional problems.

The RSMVA algorithm can be regarded as a forward greedy algorithm because it considers adding at each step another covariate to the current model. A drawback, as in many other greedy forward algorithms, is that if a predictor has been wrongly selected then it cannot be removed anymore. Adding a backward elimination process would help correct mistakes made in earlier forward selection steps. This research direction is currently in progress.

Appendix section

Appendix A

It can be shown [19] that the lower bound L on $\log p(\boldsymbol{y})$ is

$$\begin{aligned}
L \quad = \quad & \log p(\boldsymbol{\mu}^q_\gamma) - \frac{n}{2}\log(2\pi) + \frac{(p+q)k}{2} - \frac{1}{2}\sum_{j=1}^k \log|\Sigma^0_{\beta_j}| - \frac{1}{2}\sum_{j=1}^k \log|\Sigma^0_{\alpha_j}| \\
& -\frac{1}{2}\sum_{j=1}^k \mathrm{tr}({\Sigma^0_{\beta_j}}^{-1}\Sigma^q_{\beta_j}) - \frac{1}{2}\sum_{j=1}^k(\boldsymbol{\mu}^q_{\beta_j} - \boldsymbol{\mu}^0_{\beta_j})'{\Sigma^0_{\beta_j}}^{-1}(\boldsymbol{\mu}^q_{\beta_j} - \boldsymbol{\mu}^0_{\beta_j}) \\
& -\frac{1}{2}\sum_{j=1}^k \mathrm{tr}({\Sigma^0_{\alpha_j}}^{-1}\Sigma^q_{\alpha_j}) - \frac{1}{2}\sum_{j=1}^k(\boldsymbol{\mu}^q_{\alpha_j} - \boldsymbol{\mu}^0_{\alpha_j})'{\Sigma^0_{\alpha_j}}^{-1}(\boldsymbol{\mu}^q_{\alpha_j} - \boldsymbol{\mu}^0_{\alpha_j}) \\
& +\sum_{i=1}^n\sum_{j=1}^k q_{ij}\log\frac{p_{ij}(\boldsymbol{\mu}^q_\gamma)}{q_{ij}} + \frac{1}{2}\sum_{j=1}^k\log|\Sigma^q_{\beta_j}| + \frac{1}{2}\sum_{j=1}^k\log|\Sigma^q_{\alpha_j}| \\
& -\frac{1}{2}\sum_{i=1}^n\sum_{j=1}^k q_{ij}\left\{\boldsymbol{w}_i'\boldsymbol{\mu}^q_{\alpha_j} + \frac{(y_i - \boldsymbol{v}_i'\boldsymbol{\mu}^q_{\beta_j})^2 + \boldsymbol{v}_i'\Sigma^q_{\beta_j}\boldsymbol{v}_i}{\exp(\boldsymbol{w}_i'\boldsymbol{\mu}^q_{\alpha_j} - \frac{1}{2}\boldsymbol{w}_i'\Sigma^q_{\alpha_j}\boldsymbol{w}_i)}\right\}. \qquad \text{(A.1)}
\end{aligned}$$

This needs to be optimized with respect to $\boldsymbol{\mu}^q_{\beta_j}$, $\Sigma^q_{\beta_j}$, $\boldsymbol{\mu}^q_{\alpha_j}$, $\Sigma^q_{\alpha_j}$ for $j = 1, \ldots, k$, $\boldsymbol{\mu}^q_\gamma$ and q_{ij} for $i = 1, \ldots, n$, $j = 1, \ldots, k$.

Maximization with respect to $\boldsymbol{\mu}^q_{\beta_j}$, with other terms held fixed, leads to

$$\boldsymbol{\mu}^q_{\beta_j} = \left(V'D_jV + {\Sigma^0_{\beta_j}}^{-1}\right)^{-1}\left({\Sigma^0_{\beta_j}}^{-1}\boldsymbol{\mu}^0_{\beta_j} + V'D_j\boldsymbol{y}\right), \qquad \text{(A.2)}$$

where $V = (\boldsymbol{v}_1', \ldots, \boldsymbol{v}_n')'$ is the design matrix for the mean model, and D_j is the diagonal matrix with ith entry $q_{ij}/\exp(\boldsymbol{w}_i'\boldsymbol{\mu}^q_{\alpha_j} - \frac{1}{2}\boldsymbol{w}_i'\Sigma^q_{\alpha_j}\boldsymbol{w}_i)$, $j = 1, \ldots, k$. Maximization with respect to $\Sigma^q_{\beta_j}$ leads to

$$\Sigma^q_{\beta_j} = \left(V'D_jV + {\Sigma^0_{\beta_j}}^{-1}\right)^{-1}. \qquad \text{(A.3)}$$

If no parametric form for the variational posterior $q(\boldsymbol{\alpha}_j)$ is assumed then the optimal choice for $q(\boldsymbol{\alpha}_j)$ is [see, for example, 21]

$$q(\boldsymbol{\alpha}_j) \propto \exp\Big(E(\log[p(\boldsymbol{\theta})p(\boldsymbol{y}|\boldsymbol{\theta})])\Big),$$

where the expectation is with respect to all parameters except $\boldsymbol{\alpha}_j$. It can be shown that

$$\begin{aligned}
q(\boldsymbol{\alpha}_j) \propto \exp\Bigg(& -\frac{1}{2}\sum_{i=1}^n q_{ij}\left[\boldsymbol{w}_i'\boldsymbol{\alpha}_j + \frac{(y_i - \boldsymbol{v}_i'\boldsymbol{\mu}^q_{\beta_j})^2 + \boldsymbol{v}_i'\Sigma^q_{\beta_j}\boldsymbol{v}_i}{\exp(\boldsymbol{w}_i'\boldsymbol{\alpha}_j)}\right] \\
& -\frac{1}{2}(\boldsymbol{\alpha}_j - \boldsymbol{\mu}^0_{\alpha_j})'{\Sigma^0_{\alpha_j}}^{-1}(\boldsymbol{\alpha}_j - \boldsymbol{\mu}^0_{\alpha_j})\Bigg), \qquad \text{(A.4)}
\end{aligned}$$

which takes the form of the posterior (apart from a normalization constant) in a Bayesian generalized linear model (GLM) with gamma response and log link, coefficient of variation $\sqrt{2/q_{ij}}$, responses $\lambda_{ij} = (y_i - \boldsymbol{v}_i'\boldsymbol{\mu}^q_{\beta_j})^2 + \boldsymbol{v}_i'\Sigma^q_{\beta_j}\boldsymbol{v}_i$, $i = 1, \ldots, n$, and the log of the mean response being $\boldsymbol{w}_i'\boldsymbol{\alpha}_j$. The prior in this Bayesian GLM is $N(\boldsymbol{\mu}^0_{\alpha_j}, \Sigma^0_{\alpha_j})$. If we use a quadratic approximation to log $q(\boldsymbol{\alpha}_j)$, then this results in a normal approximation to $q(\boldsymbol{\alpha}_j)$ with the mean and variance the posterior mode and the negative inverse Hessian of the log posterior at the mode. The computations required are standard ones in fitting a Bayesian GLM. Write $\Lambda(\boldsymbol{\alpha}_j)$ (as a function of $\boldsymbol{\alpha}_j$) for the diagonal matrix with entries $\frac{1}{2}\lambda_{ij}\exp(-\boldsymbol{w}_i'\boldsymbol{\alpha}_j)$, $i = 1, \ldots, n$. With $\boldsymbol{\mu}^q_{\alpha_j}$ the posterior mode, we obtain for $\Sigma^q_{\alpha_j}$ the expression

$$\Sigma^q_{\alpha_j} = \left(W'\Lambda(\boldsymbol{\mu}^q_{\alpha_j})W + {\Sigma^0_{\alpha_j}}^{-1}\right)^{-1}, \tag{A.5}$$

with $W = (\boldsymbol{w}_1', \ldots, \boldsymbol{w}_n')'$ the design matrix for the variance model. Maximization with respect to q_{ij} is easy. Letting

$$T_{ij} = \exp\left(-\frac{1}{2}\boldsymbol{w}_i'\boldsymbol{\mu}^q_{\alpha_j} - \frac{1}{2}\frac{(y_i - \boldsymbol{v}_i'\boldsymbol{\mu}^q_{\beta_j})^2 + \boldsymbol{v}_i'\Sigma^q_{\beta_j}\boldsymbol{v}_i}{\exp(\boldsymbol{w}_i'\boldsymbol{\mu}^q_{\alpha_j} - \frac{1}{2}\boldsymbol{w}_i'\Sigma^q_{\alpha_j}\boldsymbol{w}_i)}\right),$$

we have that

$$q_{ij} = \frac{p_{ij}T_{ij}}{\sum_{l=1}^k p_{il}T_{il}}. \tag{A.6}$$

Finally, maximization of the lower bound L with respect to $\boldsymbol{\mu}^q_\gamma$ is equivalent to maximization of log $p(\boldsymbol{\mu}^q_\gamma) + \sum_{i,j} q_{ij} \log p_{ij}(\boldsymbol{\mu}^q_\gamma)$. This is the log posterior in a Bayesian multinomial regression with a normal prior on the regression parameter $\boldsymbol{\mu}^q_\gamma$ and with ith response $(q_{i1}, \ldots, q_{ik})'$. Although the response vectors are not in the typical multinomial form, the usual iterative optimization algorithms can be used to find the mode. The Newton method for fitting this Bayesian multinomial regression model is presented in Appendix B.

We now summarize the above optimization process. After initializing the parameters, the following iterative scheme is performed.

Algorithm 1: Maximization of the variational lower bound

1. Update $\boldsymbol{\mu}^q_{\beta_j}$, $j = 1, \ldots, k$ as in (A.2).
2. Update $\Sigma^q_{\beta_j}$, $j = 1, \ldots, k$ as in (A.3).
3. Update $\boldsymbol{\mu}^q_{\alpha_j}$ as the posterior mode in the Bayesian gamma GLM (A.4).
4. Update $\Sigma^q_{\alpha_j}$, $j = 1, \ldots, k$ as in (A.5).
5. Update q_{ij} as in (A.6), $i = 1, \ldots, n$, $j = 1, \ldots, k$.
6. Update $\boldsymbol{\mu}^q_\gamma$ as the posterior mode in a Bayesian multinomial regression with normal prior $N(\boldsymbol{\mu}^0_\gamma, \Sigma^0_\gamma)$ and ith response $(q_{i1}, \ldots, q_{ik})'$.
7. Repeat steps 1-6 until the increase in the variational lower bound (A.1) is less than some user-specified tolerance.

To initialize the algorithm, [19] first perform a k-means clustering algorithm to cluster n vectors $(y_i, \boldsymbol{v}_i)_{i=1,\ldots,n}$ into k clusters, then assign 1 to q_{ij} if the ith observation lies in cluster j and 0 otherwise. For each cluster j, an ordinary least squares fit of y_i to predictors $\boldsymbol{v}_i$ is performed to get an estimate $\hat{\boldsymbol{\beta}}_j$ of $\boldsymbol{\beta}_j$ and residuals $r_i = (y_j - \boldsymbol{v}_i'\hat{\boldsymbol{\beta}}_j)^2$, then an initial estimate for $\boldsymbol{\mu}^q_{\alpha_j}$ and $\Sigma^q_{\alpha_j}$ is obtained by fitting $\log r_i$ to the predictors $\boldsymbol{w}_i$. The iterative scheme in Algorithm 1 is now ready to be performed for all parameters. In our experience, the final estimate of the parameters at convergence is typically insensitive to the initial value of q_{ij}. However, a good initial value makes the algorithm converge quickly.

We now discuss our choice of the hyperparameters. For simplicity, we set $\boldsymbol{\mu}^0_{\beta_j} = \boldsymbol{0}$, $\boldsymbol{\mu}^0_{\alpha_j} = \boldsymbol{0}$, $\boldsymbol{\mu}^0_{\gamma} = \boldsymbol{0}$ and $\Sigma^0_{\beta_j} = (\sigma^0_{\beta_j})^2 I$, $\Sigma^0_{\alpha_j} = (\sigma^0_{\alpha_j})^2 I$, $\Sigma^0_{\gamma} = (\sigma^0_{\gamma})^2 I$ with I the identity matrix. The hyperparameters $(\sigma^0_{\beta_j})^2, (\sigma^0_{\alpha_j})^2$ and $(\sigma^0_{\gamma})^2$ can be estimated by empirical Bayes. The log posterior for these hyperparameters is (apart from an independent constant)

$$\log p((\sigma^0_{\beta_j})^2, (\sigma^0_{\alpha_j})^2, (\sigma^0_{\gamma})^2) + \log p(\boldsymbol{y}|(\sigma^0_{\beta_j})^2, (\sigma^0_{\alpha_j})^2, (\sigma^0_{\gamma})^2) \tag{A.7}$$

where the second term can be approximated by the lower bound (A.1). If we assume independent inverse gamma priors, $\mathrm{IG}(a,b)$, for these parameters, then maximizing (A.7) leads to

$$\begin{aligned}
(\sigma^0_{\beta_j})^2 &= \frac{b + \frac{1}{2}\boldsymbol{\mu}^q_{\beta_j}{}'\boldsymbol{\mu}^q_{\beta_j} + \frac{1}{2}\mathrm{tr}(\Sigma^q_{\beta_j})}{a+1+\frac{p}{2}},\ j=1,\dots,k,\\
(\sigma^0_{\alpha_j})^2 &= \frac{b + \frac{1}{2}\boldsymbol{\mu}^q_{\alpha_j}{}'\boldsymbol{\mu}^q_{\alpha_j} + \frac{1}{2}\mathrm{tr}(\Sigma^q_{\alpha_j})}{a+1+\frac{q}{2}},\ j=1,\dots,k,\\
(\sigma^0_{\gamma})^2 &= \frac{b + \frac{1}{2}\boldsymbol{\mu}^q_{\gamma}{}'\boldsymbol{\mu}^q_{\gamma}}{a+1+\frac{(k-1)r}{2}}.
\end{aligned}$$

These updates can be added to Algorithm 1 given above.

Appendix B

We now present the Newton method for fitting a Bayesian multinomial regression model with a normal prior. Let $\boldsymbol{y}_1, \dots, \boldsymbol{y}_n \in \mathbb{R}^k$ be n multinomial-type responses with k categories and $\boldsymbol{z}_1, \dots, \boldsymbol{z}_n \in \mathbb{R}^p$ be vectors of covariates. Note that in the application to find the mode $\boldsymbol{\mu}^q_\gamma$ in step 6 of Algorithm 1, the ith pseudo-response vector is $(q_{i1}, \dots, q_{ik})'$. Write $Y = (\boldsymbol{y}_1', \dots, \boldsymbol{y}_n')'$, $Z = (\boldsymbol{z}_1', \dots, \boldsymbol{z}_n')'$ and Y^* as the matrix Y without the first column. The goal is to minimize the negative of the log of the posterior

$$-\log p(\boldsymbol{\gamma}|Y) \propto -\sum_{i=1}^{n}\sum_{j=1}^{k} y_{ij}\log p_{ij}(\boldsymbol{\gamma}) + \frac{1}{2}(\boldsymbol{\gamma} - \boldsymbol{\mu}^0_\gamma)'\Sigma_0^{-1}(\boldsymbol{\gamma} - \boldsymbol{\mu}^0_\gamma)$$

where

$$p_{ij}(\boldsymbol{\gamma}) = \frac{\exp(\boldsymbol{z}_i'\boldsymbol{\gamma}_j)}{1 + \sum_{s=2}^{k}\exp(\boldsymbol{z}_i'\boldsymbol{\gamma}_s)},\quad \boldsymbol{\gamma} = (\boldsymbol{\gamma}_2', \dots, \boldsymbol{\gamma}_k')'.$$

Write $P = P(\boldsymbol{\gamma}) = (p_{ij}(\boldsymbol{\gamma}))$ for the $n \times (k-1)$ matrix with entries $p_{ij}(\boldsymbol{\gamma})$, $i = 1, \dots, n,\ j = 2, \dots, k$. The gradient can then be written as

$$u(\boldsymbol{\gamma}) := -\frac{\partial \log p(\boldsymbol{\gamma}|Y)}{\partial \boldsymbol{\gamma}} = \mathrm{vec}(Z'(P - Y^*)) + \Sigma_0^{-1}(\boldsymbol{\gamma} - \boldsymbol{\mu}_\gamma^0)$$

with vec(·) the vectorization operator. The Hessian is of the form

$$H(\boldsymbol{\gamma}) := -\frac{\partial^2 \log p(\boldsymbol{\gamma}|Y)}{\partial \boldsymbol{\gamma} \partial \boldsymbol{\gamma}'} = \Gamma + \Sigma_0^{-1}$$

where $\Gamma = \Gamma(\boldsymbol{\gamma})$ is a $(k-1) \times (k-1)$ block matrix with the (i, j) block of the form

$$\gamma_{ij} = \begin{cases} Z'\mathrm{diag}(\boldsymbol{p}_i \otimes (\boldsymbol{p}_i - \mathbf{1}))Z, & \text{if} \quad i = j \\ -Z'\mathrm{diag}(\boldsymbol{p}_i \otimes \boldsymbol{p}_j)Z, & \text{if} \quad i \neq j \end{cases}$$

where $\boldsymbol{p}_j$ is the jth column of P, $\mathbf{1}$ is the vector of 1's and $\otimes$ is the direct product. The Newton method for estimating the mode of $p(\boldsymbol{\gamma}|Y)$ is as follows.

- Initialization: Set starting value $\boldsymbol{\gamma}^{(0)}$.
- Iteration: For $k = 1, 2, \dots$, update $\boldsymbol{\gamma}^{(k)} = \boldsymbol{\gamma}^{(k-1)} - H^{-1}(\boldsymbol{\gamma}^{(k-1)})u(\boldsymbol{\gamma}^{(k-1)})$ until some stopping rule is satisfied.

References

[1] Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. New York: Springer. MR2247587

[2] Bishop, C. M. and Svensen, M. (2003). Bayesian hierarchical mixtures of experts. In: U. Kjaerulff and C. Meek (Eds.), Proceedings Nineteenth Conference on Uncertainty in Artificial Intelligence, pp. 57-64, Morgan Kaufmann, San Francisco, CA.

[3] Boughton, W. (2004). The Australian water balance model. *Environmental Modelling and Software* 19, 943-956.

[4] Calinski, R. B. and Harabasz, J. (1974). A dendrite method for cluster analysis. *Communications in Statistics*, 3(1), 1-27. MR0375641

[5] Chan, D., Kohn, R., Nott, D. J. and Kirby, C. (2006). Adaptive nonparametric estimation of mean and variance functions. *Journal of Computational and Graphical Statistics*, 15, 915-936. MR2273484

[6] Chen, J. and Chen, Z. (2008). Extended Bayesian information criteria for model selection with large model spaces. *Biometrika*, 95, 759-771. MR2443189

[7] Corduneanu, A., and Bishop, C. M. (2001). Variational Bayesian model selection for mixture distributions. In: T. Jaakkola and T. Richardson (Eds), Artifcial Intelligence and Statistics, 27-34, Morgan Kaufmann, San Francisco, CA.

[8] Efron, B., Hastie, T., Johnstone, I., and Tibshirani, R. (2004). Least angle regression (with discussion). *The Annals of Statistics*, 32, 407–451. MR2060166

[9] George, E. I. and McCulloch, R. E. (1993). Variable selection via Gibbs sampling. *Journal of the American Statistical Association*, 88, 881-889.

[10] Geweke, J. and Keane, M. (2007). Smoothly mixing regressions. *Journal of Econometrics*, 138, 252–290. MR2380699

[11] Hall, P., Pham, T., Wand, M. P. and Wang, S. S. J. (2011). Asymptotic normality and valid inference for Gaussian variational approximation. *The Annals of Statistics*, 39(5), 2502-2532. MR2906876

[12] Jacobs, R., Jordan, M., Nowlan, S. and Hinton, G. (1991). Adaptive mixtures of local experts. *Neural Computation*, 3, 79-87.

[13] Jiang, W. and Tanner, M. A. (1999). Hierarchical mixture-of-experts for exponential family regression models: Approximation and maximum likelihood estimation. *The Annals of Statistics*, 27, 987–1011. MR1724038

[14] Jordan, M. I. and Jacobs, R. A. (1994). Hierarchical mixtures of experts and the EM algorithm. *Neural Computation*, 6, 181-214.

[15] Jordan, M. I., Ghahramani, Z., Jaakkola, T. S., Saul, L. K. (1999). An introduction to variational methods for graphical models. In M. I. Jordan (Ed.), Learning in Graphical Models. MIT Press, Cambridge.

[16] Mallat, S. G. and Zhang, Z. (1993). Matching pursuits with time-frequency dictionaries. *IEEE Transactions on signal processing*, 41, 3397-3415.

[17] McGrory, C. A. and Titterington, D. M. (2007). Variational approximations in Bayesian model selection for finite mixture distributions. *Computaional Statistics and Data Analysis*, 51, 5352-5367. MR2370876

[18] McLachlan, G. J. and Peel, D. (2000). *Finite Mixture Models*. Wiley, New York. MR1789474

[19] Nott, D. J., Tan, S. L., Villani, M. and Kohn, R. (2011). Regression density estimation with variational methods and stochastic approximation. *Journal of Computational and Graphical Statistics*, to appear. Preprint: `http://villani.files.wordpress.com/2010/02/variational-heteroscedastic-moe-july-6-20114.pdf`

[20] Nott, D. J., Tran, M.-N. and Leng, C. (2012). Variational approximation for heteroscedastic linear models and matching pursuit algorithms. *Statistics and Computing*, 22(2), 497-512.

[21] Ormerod, J. T. and Wand, M. P. (2010). Explaining variational approximation. *The American Statistician*, 64(2), 140-153. MR2757005

[22] Ueda, N., Nakano, R., Ghahramani, Z. and Hinton, G. E. (2000). SMEM algorithm for mixture models. *Neural Computation*, 12, 2109-2128.

[23] Ueda, N. and Ghahramani, Z. (2002). Bayesian model search for mixture models based on optimizing variational bounds. *Neural Networks*, 15, 1223-1241.

[24] Richardson, S. and Green, P. J. (1997). On Bayesian analysis of mixtures with an unknown number of components. *Journal of the Royal Statistical Society B*, 59(4), 731-792. MR1483213

[25] Ruppert, D., Wand, M. P. and Carroll, R. J. (2003). *Semiparametric Regression*, Cambridge University Press, Cambridge. MR1998720

[26] Szekely, G. J. and Rizzo, M. L. (2009). Brownian distance covariance. *The Annals of Applied Statistics*, 3, 1236–1265. MR2752127

[27] Titterington, D. M., Smith, A. F. M. and Makov, U. E. (1985). *Statistical Analysis of Finite Mixture Distributions*. John Wiley & Sons, New York. MR0838090

[28] Villani, M., Kohn, R. and Giordani, P. (2009). Regression density estimation using smooth adaptive Gaussian mixtures. *Journal of Econometrics*, 153, 155–173. MR2558502

[29] Waterhouse, S., MacKay, D. and Robinson, T. (1996). Bayesian methods for mixtures of experts. In: D.S. Touretzky, M.C. Mozer and M.E. Hasselmo (Eds.), Advances in Neural Information Processing Systems 8, pp. 351-357, MIT Press, Cambridge.

[30] Wood, S.A., Jiang, W., and Tanner, M.A. (2002). Bayesian mixture of splines for spatially adaptive nonparametric regression. *Biometrika*, 89, 513-528. MR1929159

[31] Wood, S.A., Kohn, R., Cottet, R., Jiang, W. and Tanner, M. (2008). Locally adaptive nonparametric binary regression. *Journal of Computational and Graphical Statistics*, 17, 352-372. MR2439964

[32] Wu, B., McGrory, C. A. and Pettitt, A. N. (2012). A new variational Bayesian algorithm with application to human mobility pattern modeling. *Statistics and Computing*, 22(1), 185-203. MR2865064

[33] Zhang, T. (2009). On the consistency of feature selection using greedy least squares regression. *Journal of Machine Learning Research*, 10, 555-568. MR2491749