Journal of the Korean Data & Information Science Society
2017, **28**(3), 685–695

http://dx.doi.org/10.7465/jkdi.2017.28.3.685
한국데이터정보과학회지

Semiparametric mixture of experts with unspecified gate network[†]

Dahai Jung[1] · Byungtae Seo[2]

[12]Department of Statistics, Sungkyunkwan University

Received 22 March 2017, revised 17 May 2017, accepted 18 May 2017

Abstract

The traditional mixture of experts (ME) modeled the gate network using a certain parametric function. However, if the assumed parametric function does not properly reflect the true nature, the prediction strength of ME would become weak. For example, the parametric ME often uses logistic or multinomial logistic models for the network model. However, this could be very misleading if the true nature of the data is quite different from those models. Although, in this case, we may develop more flexible parametric models by extending the model at hand, we will never be free from such misspecification problems. In order to alleviate such weakness of the parametric ME, we propose to use the semi-parametric mixture of experts (SME) in which the gate network is estimated in a non-parametrical way. Based on this, we compared the performance of the SME with those of ME and neural networks via several simulation experiments and real data examples.

Keywords: EM algorithm, mixture of experts, neural network, semiparametric models.

1. Introduction

For recent decades, mixture models have gained its popularity as a tool to explore hidden structure in the data and give flexibility to many existing models. For example, Lee (2004) used mixture models in clustering curves in microarray data, and Oh (2014) and Hwang *et al.* (2015) used mixtures of some discrete distributions to provide more flexible distributions than some well known distributions. Among those, the mixture regression first introduced by Goldfeld and Quandt (1973) has been an important alternative to usual regression models as it can capture unseen data structure. Mixture regression models assume that the relationship between a response variable and explanatory variables is not confirmative, instead several relationships are mixed along with some mixing proportions. This mixture regression model

[†] This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Techonology (NRF-2016R1A2B4007373).

[1] Master student, Department of Statistics, Sungkyunkwan University, Seoul 03063, Korea.

[2] Corresponding author: Associate Professor, Department of Statistics, Sungkyunkwan University, Seoul 03063, Korea. E-mail: seobt@skku.edu

enables us to find unseen data structure so that one can draw some useful information from the data.

In machine learning literature, mixture regression models have been used and called *mixture of experts* (ME). The basic structure of the ME is the same as the usual mixture regression but it allows extra flexibility by assuming that the mixing proportion also depends on some of explanatory variables. The ME is helpful especially for prediction purpose because modeling the mixing proportion can improve its prediction accuracy. Jacob *et al.* (1991) used this model as a tool for machine learning, and Jordan and Jacobs (1994) extended this approach to the hierarchical mixture of experts architecture which is closely related to the decision tree and multivariate spline algorithms. For more comprehensive survey on the ME, see Masoudnia and Ebrahimpour (2014).

In ME, the mixing proportion is called *gating network* as it controls which expert one needs to use. The gating network function is typically modeled using some parametric models such as logistic or mutinomial logit models. However, this parametric model often fails to yield good prediction strength because the performance becomes poor if the assumed parametric model is not suitable. For this, Young and Hunter (2010) proposed a semiparametric mixture regression model in which the mixing proportion is estimated nonparametrically. This idea was further refined and developed in Huang and Yao (2012). In this paper, we exemplified how one can use the semi-parametric mixture of experts (SME) especially for prediction purpose. In addition, we study the performance of SME compared to the traditional ME and neural network models via simulation experiments and real data analysis.

This paper is organized as follows. In Section 2, we give a brief review of mixture of experts models. We then present the semi-parametric mixture of experts and the estimation procedure in Section 3. Simulation studies along with some real data analysis are provided in Section 4. Finally, conclusions are given in Section 5.

2. Mixture of experts

2.1. Mixture of Experts

The ME model was first introduced by Jacobs *et al.* (1991) as an alternative neural network model. The fundamental principle for mixture of experts is *divide and conquer*. This principle states that large problems can be easily solved by partitioning them into smaller problems. Each small problem can be solved by each expert, and the gate network then combines the outputs from experts.

The ME consists of gating network and a set of experts. Given input $\mathbf{x}$, the gating function determines the probability of each expert assignment. Hence, in the conventional ME, a large problem is stochastically split into smaller problems. The logistic function and softmax function are popular choices for the gating network model, and the expert can be any linear or non-linear function.

The ME can also be viewed as a mixture regression model in which the mixing proportion is the gating function and each component regression model is the expert. To explain this further, let $D = \{(\mathbf{x}_i, Y_i) \in \mathbb{R}^p \times \mathbb{R} : i = 1, \ldots, n\}$ be the data, where Y_i's are scalar-valued target or output variables and $\mathbf{x}_i$'s are p-variate input vectors. Suppose further that Θ^g and Θ^e are the sets of gate and expert parameters and $\Theta = \{\Theta^g, \Theta^e\}$. The conditional probability

density function of Y given $\mathbf{x}$ is then represented as

$$f(y|\mathbf{x},\Theta) = \sum_{j=1}^{J} g(\mathbf{x};\Theta_j^g) f(y|\mathbf{x};\Theta_j^e). \tag{2.1}$$

The j-th component density $f(y|\mathbf{x};\Theta_j^e)$ represents the probability structure of the target variable Y given $\mathbf{x}$ in the j-th expert. The mixing proportion or gate function $g(\mathbf{x};\Theta_j^g)$ represents the probability that the target variable Y comes from the j-th expert. There are many ways to choose a parametric form for the gate function and experts. Among those, the most popular choice for $g(\mathbf{x};\Theta_j^g)$ is the softmax function

$$g(\mathbf{x};\Theta_j^g) = g(\mathbf{x};\boldsymbol{\alpha}_j) = \frac{\exp(\mathbf{x}^T\boldsymbol{\alpha}_j)}{\sum_{j=1}^{J}\exp(\mathbf{x}^T\boldsymbol{\alpha}_j)},$$

where $\boldsymbol{\alpha}_j$ is a $p \times 1$ unknown parameter vector. The choice of each expert involves the parametrization of the mean function and the choice of the conditional density. Although there are various ways for this choice, in this paper, we will largely consider the linear expert with normal errors. That is, we assume that $f(y|\mathbf{x};\Theta_j^e)$ is the normal density with mean $\mathbf{x}^T\boldsymbol{\beta}_j$ and variance σ_j^2. In this case, (2.1) is reduced to

$$f(y|\mathbf{x};\Theta) = \sum_{j=1}^{J} \frac{\exp(\mathbf{x}^T\boldsymbol{\alpha}_j)}{\sum_{l=1}^{J}\exp(\mathbf{x}^T\boldsymbol{\alpha}_l)} \phi(y;\mathbf{x}^T\boldsymbol{\beta}_j,\sigma_j^2),$$

where $\phi(y;a,b)$ is the normal density with mean a and variance b.

Based on this model and its estimatd parameters $\hat{\alpha}_j$'s and $\hat{\beta}_j$'s, the prediction of Y given $\mathbf{x}$ can be obtained by computing the conditional expectation of Y given $\mathbf{x}$ which is just the weighted sum of predicted values of each expert. That is,

$$\hat{y} = \sum_{j=1}^{J} g(\mathbf{x};\hat{\boldsymbol{\alpha}}_j)\mathbf{x}^T\hat{\boldsymbol{\beta}}_j.$$

2.2. The EM algorithm

In ME models, the parameters of both the gate and the experts can be learned using the expectation-maximization (EM) algorithm. The EM algorithm has been developed to iteratively find the maximum likelihood estimator (MLE) when some of random variables are missing (Dempster *et al.*, 1977). Although we assume that there is no missing variable in a given dataset, the EM algorithm can still be used by introducing latent variables. For this, we introduce a random indicator vector $\boldsymbol{Z}_i = (Z_{i1},\ldots,Z_{iJ})$ defined as

$$Z_{ij} = \begin{cases} 1, & \text{if } Y_i \text{ is obtained from } j\text{-th expert} \\ 0, & \text{elsewhere .} \end{cases}$$

Then the joint probability density of $(Y_i,\boldsymbol{Z}_i)$ given $\mathbf{x}_i$ is

$$f(y_i,\mathbf{z}_i) = \prod_{j=1}^{J}\left[g(\mathbf{x}_i;\boldsymbol{\alpha}_j)\phi(y_i;\mathbf{x}_i^T\boldsymbol{\beta}_j,\sigma_j^2)\right]^{z_{ij}},$$

and the complete log-likelihood function can be written as

$$l(\Theta) = \sum_{i=1}^{n}\sum_{j=1}^{J} z_{ij}\left[\log g(\mathbf{x}_i;\boldsymbol{\alpha}_j) + \log\phi(y_i;\mathbf{x}_i^T\boldsymbol{\beta}_j,\sigma_j^2)\right]. \tag{2.2}$$

By doing this, we can see that (2.2) is split into functions of each component parameters, and this enables us to separately estimate parameters of each component and gate parameters. However, since $\boldsymbol{Z}_i$ is not observable, $l(\Theta)$ cannot be directly maximized and the EM algorithm can be used in this case. The EM algorithm consists of expectation (E) and maximization (M) steps. In the E-step, we need to compute the the conditional expectation of (2.2) given all observed variables, and this is equivalent to computing $E[Z_{ij}|\mathbf{x}_i, Y_i;\Theta^{(t)}]$ which can be computed as

$$\begin{aligned}\hat{z}_{ij} &= E[Z_{ij}|\mathbf{x}_i, Y_i = y_i;\Theta^{(t)}]\\ &= P(Z_{ij}=1|\mathbf{x}_i, Y_i = y_i;\Theta^{(t)})\\ &= \frac{g\left(\mathbf{x}_i;\boldsymbol{\alpha}_j^{(t)}\right)\phi\left(y_i;\mathbf{x}_i^T\boldsymbol{\beta}_j^{(t)},\sigma_j^{2(t)}\right)}{\sum_{l=1}^{J} g\left(\mathbf{x}_i;\boldsymbol{\alpha}_l^{(t)}\right)\phi\left(y_i;\mathbf{x}_i^T\boldsymbol{\beta}_l^{(t)},\sigma_l^{2(t)}\right)},\end{aligned} \tag{2.3}$$

where the superscript (t) represents the iteration number and $\Theta^{(t)} = (\boldsymbol{\alpha}^{(t)}, \boldsymbol{\beta}_1^{(t)},\ldots,\boldsymbol{\beta}_J^{(t)}, \sigma_1^{2(t)},\ldots,\sigma_J^{2(t)})$ is the set of all parameters given at t-th iteration.

The M-step is then to find the maximizer of

$$Q(\Theta|\Theta^{(t)}) = \sum_{i=1}^{n}\sum_{j=1}^{J}\hat{z}_{ij}\left[\log g(\mathbf{x}_i;\boldsymbol{\alpha}_j) + \log\phi(y_i;\mathbf{x}_i^T\boldsymbol{\beta}_j,\sigma_j^2)\right].$$

If one uses logistic or multinomial logit function for the gate function $g(\mathbf{x})$, additional iterative procedure is required to update $\boldsymbol{\alpha}_j$ because there is no closed form. In this case, we generally use the iterative reweighted least square technique to update the gate parameter $\boldsymbol{\alpha}_1,\ldots,\boldsymbol{\alpha}_J$. In addition, if we use linear experts with normal errors, the expert parameters can be explicitly computed as

$$\boldsymbol{\beta}_j^{(t+1)} = (\boldsymbol{X}^T\hat{\mathbf{Z}}_{(j)}\boldsymbol{X})^{-1}\boldsymbol{X}\hat{\mathbf{Z}}_{(j)}\mathbf{Y}, \tag{2.4}$$

$$\sigma_j^{2(t+1)} = \frac{(\mathbf{Y}-\boldsymbol{X}\boldsymbol{\beta}_j^{(t+1)})^T\hat{\mathbf{Z}}_{(j)}(\mathbf{Y}-\boldsymbol{X}\boldsymbol{\beta}_j^{(t+1)})}{tr(\hat{\mathbf{Z}}_{(j)})}, \tag{2.5}$$

where $\mathbf{X} = \left(\boldsymbol{X}_1^T,\ldots,\boldsymbol{X}_n^T\right)$, $\mathbf{Y} = (Y_1,\ldots,Y_n)^T$, and $\hat{\mathbf{Z}}_{(j)} = diag(\hat{z}_{1j},\ldots,\hat{z}_{nj})$. We practice the EM algorithm until the observed likelihood converges.

3. Semi-parametric mixture of experts

3.1. Semi-parametric mixture of experts

The ME uses a parametric model to estimate the mixing proportion. However, this can cause misspecification problems and result in wrong statistical inference especially when

our focus is mainly on the prediction of a response variable. For this, Huang and Yao (2012) proposed a nonparametric method to estimate the mixing proportion using a kernel regression technique and a local likelihood method for the mixture of regression models. Since their method can also be used in our mixture of experts problem, we adopt their method and call semi-parametric mixture of experts (SME).

Because we do not assume any functional form for mixing proportions, we can expect that SME is robust to the model misspecification and have good prediction strength without any prior information for the mixing proportion. In this case, we write (2.1) as

$$f(y_i|\mathbf{x}_i, \Theta) = \sum_{j=1}^{J} \pi_j(\mathbf{x}_i)\phi(y_i; \mathbf{x}_i^T\boldsymbol{\beta}_j, \sigma_j^2), \tag{3.1}$$

where $\pi_j(\mathbf{x})$'s are an unknown positive functions that represents the mixing proportion of the j-th component at given $\mathbf{x}$, and $\sum_{j=1}^{J} \pi_j(\mathbf{x}) = 1$ for all $\mathbf{x}$.

3.2. EM-like algorithm

For given sample $\{(\mathbf{x}_i, Y_i) : i = 1, \ldots, n\}$, the log-likelihood based on (3.1) can be expressed as

$$l(\boldsymbol{\pi}(\mathbf{x}), \boldsymbol{\beta}, \boldsymbol{\sigma}^2) = \sum_{i=1}^{n} \log\Big\{ \sum_{j=1}^{J} \pi_j(\mathbf{x}_i)\phi(y_i; \mathbf{x}_i^T\boldsymbol{\beta}_j, \sigma_j^2)\Big\}, \tag{3.2}$$

where $\boldsymbol{\beta} = (\boldsymbol{\beta}_1^T, \ldots, \boldsymbol{\beta}_J^T)^T$, $\boldsymbol{\sigma}^2 = (\sigma_1^2, \ldots, \sigma_J^2)^T$, and $\boldsymbol{\pi}(\mathbf{x}) = (\pi_1(\mathbf{x}), \ldots, \pi_J(\mathbf{x}))^T$. Owing to the existence of the unknown function π_j, the EM algorithm described in Section 2.2 is not directly applicable. For this, Huang and Yao (2012) proposed a one-step backfitting procedure which estimates π_j using the following local likelihood function

$$l_1(\boldsymbol{\pi}, \boldsymbol{\beta}, \boldsymbol{\sigma}^2) = \sum_{i=1}^{n} \log\Big\{ \sum_{j=1}^{J} \pi_j\phi(y_i; \mathbf{x}_i^T\boldsymbol{\beta}_j, \sigma_j^2)\Big\} K_{\mathbf{H}}(\mathbf{x}_i - \mathbf{x}), \tag{3.3}$$

where K is a kernel density function, $\mathbf{H} = diag\{h_1, \ldots, h_p\}$ is a bandwidth matrix and $K_{\mathbf{H}}(\mathbf{x}_i - \mathbf{x}) = \mathbf{H}^{-1}K(\mathbf{x}_i - \mathbf{x})$.

Then, the regression parameters can be obtained using the EM algorithm similar to Section 2.2, and these procedure will be repeated until a certain stopping criterion is satisfied. But this generally requires too much computing efforts. For this reason, Huang and Yao (2012) also provided a faster version of one-step backfitting algorithm. We here briefly describe their algorithm as follows.

In the E-step, we calculate the posterior probability that indicates the membership of each observation as

$$r_{ij}^{(t+1)} = \frac{\pi_j^{(t)}(\mathbf{x}_i)\phi(y_i; \mathbf{x}_i^T\boldsymbol{\beta}_j^{(t)}, \sigma_j^{2(t)})}{\sum_{l=1}^{J} \pi_l^{(t)}(\mathbf{x}_i)\phi(y_i; \mathbf{x}_i^T\boldsymbol{\beta}_l^{(t)}, \sigma_l^{2(t)})}, \tag{3.4}$$

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

In the M-step, we update $\boldsymbol{\beta}, \boldsymbol{\sigma}^2$, and $\boldsymbol{\pi}(\mathbf{x})$ given as

$$\begin{aligned}
\boldsymbol{\beta}_j^{(t+1)} &= (\mathbf{X}^T\mathbf{R}_j^{(t+1)}\mathbf{X})^{-1}\mathbf{X}^T\mathbf{R}_j^{(t+1)}\mathbf{Y}, \\
\sigma_j^{2(t+1)} &= \frac{\sum_{i=1}^n r_{ij}^{(t+1)}(Y_i - \mathbf{x}_i^T\boldsymbol{\beta}_j^{(t)})^2}{\sum_{i=1}^n r_{ij}^{(t+1)}}, \\
\pi_j^{(t+1)}(\mathbf{x}) &= \frac{\sum_{i=1}^n r_{ij}^{(t+1)} K_{\mathbf{H}}(\mathbf{x}_i - \mathbf{x})}{\sum_{i=1}^n K_{\mathbf{H}}(\mathbf{x}_i - \mathbf{x})},
\end{aligned}$$

where $\mathbf{R}_j^{(t+1)} = diag\{r_{1j}^{(t+1)}, \ldots, r_{nj}^{(t+1)}\}$.

3.3. Choice of bandwidth

Since our focus is the prediction strength of the SME, for the choice of bandwidth, we propose to use K-fold cross validation (K-fold CV) method based on prediction errors. For this, first we randomly split the given dataset into K subdatasets. One of the K subsets is retained as a test dataset, and the remaining K-1 sets are used as a training dataset. The CV process repeats K times with each of the K groups used exactly once for testing.

To explain this further, let D denote the full data set. First, we need to randomly split the index set $\mathcal{I} = \{1, \ldots, n\}$ into K mutually exclusive subindex sets $\mathcal{I}_1, \ldots, \mathcal{I}_K$ satisfying $\cup_{k=1}^K \mathcal{I}_k = \mathcal{I}$. The estimators $\hat{\boldsymbol{\pi}}^{(-k)}(\cdot), \hat{\boldsymbol{\sigma}}^{2(-k)}, \hat{\boldsymbol{\beta}}^{(-k)}$ are then computed based on data $\cup_{m \neq k} D_m$, where D_m is the subset of D corresponding to the index set $\mathcal{I}_m$. Now, the predicted response for the k-th subset is computed as

$$\hat{y}_i = \sum_{j=1}^J \hat{\pi}_j^{(-k)}(\mathbf{x}_i)\mathbf{x}_i^T\hat{\boldsymbol{\beta}}_j^{(-k)}, \quad \text{for} \quad i \in \mathcal{I}_k.$$

By repeating the above procedure K times, we can compute the CV score as

$$CV = \frac{1}{K}\sum_{k=1}^K \frac{1}{n_k}\sum_{i \in \mathcal{I}_k}(y_i - \hat{y}_i)^2,$$

where n_k is the cardinality of $\mathcal{I}_k$. This CV value will be computed over different bandwidth on a predetermined grid, and the optimal bandwidth would be the one with the smallest CV value.

4. Numerical studies

4.1. Simulation studies

In this section, we carry out simulation experiments to investigate the performance of ME, SME and neural networks (NN) focusing especially on their prediction power. For the simulation, we consider two linear expert models as

$$f_1(x_1, x_2) = 5 - 3x_1 - 2x_2 \quad \text{and} \quad f_2(x_1, x_2) = 1 + 2x_1 - x_2,$$

where f_1 and f_2 are the regression functions for the first and second components. In the first simulation, we consider the following logistic pattern for the mixing proportion.

$$\pi_1(x_1, x_2) = \frac{\exp(1 + 2x_1 + 0.5x_2)}{1 + \exp(1 + 2x_1 + 0.5x_2)},$$
$$\pi_2(x_1, x_2) = 1 - \pi_1(x_1, x_2).$$

With above models, we generate the covariate vector $\mathbf{x} = (x_1, x_2)$ of size $n = 500$ and 1000 from the bivariate normal distribution with mean vector $\boldsymbol{\mu}$ and covariance matrix Σ, where

$$\boldsymbol{\mu} = \begin{pmatrix} 0 \\ -3 \end{pmatrix} \quad \text{and} \quad \Sigma = \begin{pmatrix} 1 & 0 \\ 0 & 3 \end{pmatrix}.$$

The response variable Y is then generated from $N(f_1(x_1, x_2), 1)$ with probability $\pi_1(x_1, x_2)$ and from $N(f_2(x_1, x_2), 2)$ with probability $\pi_2(x_1, x_2)$.

Although, we need to select a suitable bandwidth for SME, we show, in this section, the result based on several choices of h for $\mathbf{H}_h = h\mathbf{I}$ rather than using the optimal value. Instead, in the next section, we conduct K-fold CV to choose the optimal bandwidth with real data. For fair comparison with NN, several numbers of neurons are also used for NN.

To fit the model, we use *Mixtools* (Benaglia *et al.*, 2009; Gunther and Fritsch, 2010) *Neuralnet* (Gunther and Fritsch, 2010) packages in the statistical software R for ME and NN, respectively. For SME, we used our own R program. To see the prediction ability of each model, we randomly split the data into two equal sized subsets called training and test sets. We then fit the model only using the training set and compute the residual sum of squares (RSS) and predicted residual error sum of squares (PRESS) based on 100 replications, where RSS and PRESS are defined as

$$\text{RSS} = \sum_{i \in \text{training set}} (y_i - \hat{y}_i)^2 \quad \text{and} \quad \text{PRESS} = \sum_{j \in \text{test set}} (y_j - \hat{y}_j)^2.$$

For comparison, we also report the results from the multiple linear regression model. Table 4.1 displays the results with four methods. In this table, we can see that ME shows the best prediction performance, and this is natural because the gate network model is correctly specified in ME. SME with bandwidth $h = 0.4$ has a better prediction power than the NN models and only slightly inferior to ME. Especially, we can see that RSS is decreasing and PRESS is increasing as the number of neurons is increasing. We can also see that the NN with 8 neurons shows even worse performance than the multiple linear regression in sample size $n = 500$, which shows an overfitting problem in NN.

As a second simulation experiment, we generate data in which the mixing proportion does not follow logistic pattern. The sample sizes, the number of replications, and the true expert models are the same as those of the first simulation, but the simulated models for the mixing proportion are different. We generate $\mathbf{x}$ from the bivariate normal distribution with mean $\boldsymbol{\mu}$ and covariance matrix Σ, and set the nonlogistic pattern of varying mixing proportions as

$$\pi_1(x_1, x_2) = \frac{\exp(1 + 2x_1 + 3x_1^2 + 0.5x_2 - x_2^2)}{1 + \exp(1 + 2x_1 + 3x_1^2 + 0.5x_2 - x_2^2)}, \quad \pi_2(x_1, x_2) = 1 - \pi_1(x_1, x_2),$$

$$\boldsymbol{\mu} = \begin{pmatrix} 0 \\ 3 \end{pmatrix} \quad \text{and} \quad \Sigma = \begin{pmatrix} 3 & 0 \\ 0 & 3 \end{pmatrix}.$$

Table 4.1 Average of RSS and PRESS for logistic mixing proportion

n	Model	RSS	PRESS
500	ME	2203.978	2389.651
	SME(h=0.1)	1007.906	3400.277
	SME(h=0.4)	2135.564	2515.422
	SME(h=0.9)	2588.503	2778.312
	NN(n=2)	2324.645	2747.354
	NN(n=4)	1985.630	2826.424
	NN(n=8)	1667.952	7175.497
	regression	3222.815	4184.452
1000	ME	4540.525	4655.746
	SME(h=0.1)	2591.372	5931.625
	SME(h=0.4)	4321.241	4865.523
	SME(h=0.9)	5302.616	5468.277
	NN(n=2)	4990.103	5318.647
	NN(n=4)	4315.172	5059.171
	NN(n=8)	3933.967	6150.468
	regression	6502.454	8311.495

Table 4.2 Average of RSS and PRESS for nonlogistic mixing proportion

n	Model	RSS	PRESS
500	ME	31139.399	33203.494
	SME(h=0.1)	716.622	16950.694
	SME(h=0.5)	2099.921	4656.201
	SME(h=0.9)	3921.392	5495.459
	NN(n=2)	10444.993	12113.834
	NN(n=4)	8594.578	10512.621
	NN(n=8)	3694.580	6336.545
	regression	19681.238	44142.895
1000	ME	61359.813	65969.644
	SME(h=0.1)	1395.994	29325.081
	SME(h=0.5)	4757.048	8599.068
	SME(h=0.9)	8308.995	10549.488
	NN(n=2)	19969.647	10549.488
	NN(n=4)	15003.089	17489.213
	NN(n=8)	7325.469	10578.294
	regression	39611.676	90084.673

Table 4.2 summarizes the results from the second simulation. The prediction performance of ME is very poor compared to NN and SME as we expected. Among three methods, SME with bandwidth $h = 0.5$ shows the best prediction results.

4.2. Real data analysis

As a real data example, we obtain the cardiac arrhythmia data from UCI Machine Learning Repository (`https://archive.ics.uci.edu/ml/`). This data contains 452 observations with 279 variables. Among those, we choose the heart rate as a response variable, and two independent variables: QT interval and P interval. In cardiology, the QT interval is the time between the start of the Q wave and the end of the T wave in the heart's electrical cycle. The QT interval represents electrical depolarization and repolarization of the ventricles. A

lengthened QT interval is a marker for the potential of ventricular tachyarrhythmias and a risk factor for sudden death. The P interval is average duration of P wave in millisecond.

For this dataset, we consider ME, SME and NN similar to the previous section. To choose a suitable bandwidth for SME, we compute 5-fold CV scores for 2,3, and 4 components models as shown in Table 4.3. From this table, we chose h=1.0, 1.1, and 1.3 for 2,3, and 4 components models, respectively.

Table 4.3 Cross-validation scores for 2, 3 and 4 components SME models

Number of components	h	CV score
2	0.7	89.436
	0.8	89.410
	0.9	89.407
	1.0	89.401
	1.1	89.429
	1.2	89.486
	1.3	89.730
3	0.7	90.021
	0.8	90.248
	0.9	90.019
	1.0	89.387
	1.1	89.194
	1.2	89.263
	1.3	89.273
4	0.7	88.283
	0.8	88.164
	0.9	88.360
	1.0	73.008
	1.1	71.362
	1.2	71.486
	1.3	70.094

Next, the full data set is randomly divided into training and test set to compare the performance of models by using the PRESS. Table 4.4 shows RSS and PRESS for ME, SME, and NN. In the table, the numbers in the parenthesis followed by NN represent the number of neurons in the first and second hidden layers. For example, NN(2) means NN with 2 neurons and NN(8,2) stands for NN with 8 neurons in the first hidden layer and 2 neurons in the second hidden layer.

In Table 4.4, the three components SME with bandwidth h=1.1 shows the best performance among all models considered. As the number of neurons is increasing, the prediction preformance is getting worse in NN. Furthermore, although NN(n=8,2) has the smallest RSS value, the PRESS is more than three times of the PRESS of SME(h=1.1) which again shows the overfitting problem.

In the above illustration, we tested 2,3, and 4 components models for ME and SME without considering a suitable number of components. For this data, we also computed Bayesian information criterion (BIC) values for the chosen bandwidth and these are given in Table 4.5. From this table, we can see that the 2-component model for ME and 3-component model for SME are suitable, and this is also in accord with the result obtained from Table 4.4.

Table 4.4 RSS and PRESS for various models

Model	RSS	PRESS
2 com ME	235.261	221.143
3 com ME	244.053	231.477
4 com ME	244.072	231.580
2 com SME(h=1.0)	207.266	191.625
3 com SME(h=1.1)	227.818	188.506
4 com SME(h=1.3)	239.504	210.942
NN(n=2)	210.171	207.320
NN(n=3)	145.032	298.825
NN(n=4)	140.669	333.799
NN(n=6)	112.084	493.734
NN(n=4,2)	126.393	276.631
NN(n=6,2)	113.326	547.575
NN(n=8,2)	89.659	630.297

Table 4.5 BIC values

Model	BIC
2 com ME	562.675
3 com ME	603.393
4 com ME	635.916
2 com SME(h=1.0)	550.370
3 com SME(h=1.1)	542.055
4 com SME(h=1.3)	543.618

5. Conclusions

In practice, it is generally difficult to find an appropriate functional form for the mixing proportion which depends on some of input variables. In this case, the SME can be a good alternative because the SME can allow to capture any pattern of mixing proportions unlike the parametric ME. From our simulation studies, the performance of the SME is slightly inferior to the parametric ME when the component membership is generated based on logistic models. However, when the mixing proportion is wrongly specified, the parametric ME shows poor performance while the SME still gives reasonable performance. In addition, unlike neural network models, SME is less suffered from overfitting problems. Another advantage for the use of SME is that the SME could give us some practical interpretation unlike the neural network models.

References

Benaglia, T., Chauveau, D., Hunter, D. R. and Young, D. (2009). Mixtools: An R package for analyzing finite mixture models. *Journal of Statistical Software*, **32**, 1-29.

Dempster, A. P., Laird, N. M. and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, **39**, 1-38.

Goldfeld, S. M. and Quandt, R. E. (1973). A Markov model for switching regression. *Journal of Econometrics*, **1**, 3-15.

Gunther, F. and Fritsch, S. (2010). Neuralnet: Training of neural networks. *The R Journal*, **2**, 30-38.

Huang M. and Yao, W. (2012). Mixture of regression models with varying mixing proportions: A semiparametric approach. *Journal of the American Statistical Association*, **107**, 711-724.

Hwang, S., Sohn, S. H. and Oh. C. (2015). Maximum likelihood estimation for a mixure distributions. *Journal of the Korean Data & Information Science Society*, **26**, 313-322.

Jacobs, R. A., Jordan, M. I., Nowlan, S. J. and Hinton, G. E. (1991). Adaptive mixtures of local experts. *Neural Computation*, **3**, 79-87.

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

Lee, K. E. (2004). Curve clustering in microarray. *Journal of the Korean Data & Information Science Society*, **15**, 575-584.

Masoudnia, S. and Ebrahimpour, R. (2014). Mixture of experts: A literature survey. *Artificial Intelligence Review*, **42**, 275-293.

Oh, C. (2014). A maximum likelihood estimation method for a mixture of shifted binomial distributions. *Journal of the Korean Data & Information Science Society*, **25**, 255-261.

Young, D. S. and Hunter, D. R. (2010). Mixtures of regressions with predictor-dependent mixing proportions. *Computational Statistics and Data Analysis*, **54**, 2253-2266.