Mixture of Multilayer Perceptron Regressions

Ryohei Nakano[1] and Seiya Satoh[2]
[1]*Chubu University, 1200 Matsumoto-cho, Kasugai, 487-8501 Japan*
[2]*Tokyo Denki University, Ishizaka, Hatoyama-machi, Hiki-gun, Saitama 350-0394 Japan*
nakano_ryo@isc.chubu.ac.jp, seiya.satoh@mail.dendai.ac.jp

Keywords: Mixture Models, Regression, Multilayer Perceptrons, EM Algorithm, Model Selection

Abstract: This paper investigates mixture of multilayer perceptron (MLP) regressions. Although mixture of MLP regressions (MoMR) can be a strong fitting model for noisy data, the research on it has been very rare. We employ soft mixture approach and use the EM algorithm as a basic learning framework. Our learning method goes in a double-looped manner; the outer loop is controlled by the EM algorithm and the inner loop by MLP learning method. Given data, we will have many MoMR models; thus, we need a criterion to select the best MoMR model. Bayesian Information Criterion (BIC) is used here because it works nicely for MLP model selection. Our experiments showed that the proposed MoMR found, for artificial data, the expected MoMR model as the best, and selected, for real noisy data, the MoMR model having smaller BIC and smaller residual error than those of any single regression model or any mixture of linear regressions.

1 INTRODUCTION

Mixture models have been widely used in econometrics, marketing, biology, chemistry, and many other fields. The book by McLachlan and Peel (McLachlan and Peel, 2000) contains a comprehensive review of finite mixture models.

When data arise from heterogeneous contexts, it is reasonable to introduce *mixture of regressions (MoR)* as a useful class of mixture models. In mixture of regressions, since the introduction by Goldfeld and Quandt (Goldfeld and Quandt, 1973), mixture of linear or generalized linear regressions has been focused (Bishop, 2006; Qian and Wu, 2011) and implemented as library programs (Leisch, 2004; NCSS, 2013). Around that time, Bayesian approaches to mixture of regressions were vigorously investigated using Markov chain Monte Carlo (MCMC) methods (Hurn et al., 2003).

Since this world is full of nonlinear relationships, mixture of nonlinear regressions may have the great potential. The research on the topic, however, has been relatively few. Huang, Li, and Wang (Huang et al., 2013) investigated mixture of nonlinear regressions by employing kernel regression, but they assumed that explanatory variable is univariate and the extension to multivariate will suffer from curse of dimensionality; this can be a serious limitation.

As another approach, modal regression (Chen et al., 2016) estimates the local modes of the distribution of a dependent variable given a certain value of an explanatory variable, instead of the mean, and may reveal important structure missed by usual regressions. Modal regression, however, will not give us an explicit equation as in usual regressions and the extendability to multivariate data seems not clear.

Since multilayer perceptron (MLP) is a popular powerful nonlinear model, *mixture of MLP regressions (MoMR)* will be quite a reasonable model of mixture of nonlinear regressions; however, MoMR has hardly been addressed so far.

This paper investigates MoMR. There can be two types of the mixture: hard mixture and soft mixture. In hard mixture each data point is exclusively classified into one class, while in the latter each data point belongs probabilistically to every class. Since soft mixture is more natural as a model and more appropriate for computation in terms of convergence, we employ soft mixture approach, and use the EM algorithm (Dempster et al., 1977) because soft MoMR is incomplete data problem.

This paper is organized as follows. Section 2 reviews the background knowledge of our research, and Section 3 explains model formalization, EM solver of MoMR, and model selection of MoMR. Then Section 4 describes our experiments performed to examine how our MoMR works using a two-class artificial dataset and a noisy real dataset.

2 BACKGROUND

2.1 EM Algorithm

The Expectation-Maximization (EM) algorithm is a general-purpose iterative algorithm for maximum likelihood (ML) estimation in incomplete data problems (Dempster et al., 1977). Since incomplete data problems cover a wide variety of statistical situations, the EM and its variants have been applied in many applications (McLachlan and Peel, 2000).

Suppose that a data point $(\mathbf{x},\mathbf{z})$ is generated with the density $p(\mathbf{x},\mathbf{z}|\theta)$, where only $\mathbf{x}$ is observable and $\mathbf{z}$ is hidden. Here θ denotes a parameter vector, and let $p(\mathbf{x}|\theta)$ be the density generating $\mathbf{x}$. In the EM context, $\{(\mathbf{x}^{\mu},\mathbf{z}^{\mu})\}$ is called *complete data*, and $\{\mathbf{x}^{\mu}\}$ is called *incomplete data*, where $\mu = 1,\cdots,N$.

The purpose of ML estimation is to maximize the following log-likelihood from incomplete data.

$$L(\theta) = \sum_{\mu} \log p(\mathbf{x}^{\mu}|\theta). \tag{1}$$

The log-likelihood from complete data is defined as follows.

$$L_{cmp}(\theta) = \sum_{\mu} \log p(\mathbf{x}^{\mu},\mathbf{z}^{\mu}|\theta). \tag{2}$$

The EM algorithm performs ML estimation by iteratively maximizing the following *Q-function*, where $\theta^{(t)}$ is the estimate obtained after the t-th iteration.

$$Q(\theta|\theta^{(t)}) = \sum_{\mu}\sum_{\mathbf{z}^{\mu}} P(\mathbf{z}^{\mu}|\mathbf{x}^{\mu},\theta^{(t)}) \log p(\mathbf{x}^{\mu},\mathbf{z}^{\mu}|\theta), \tag{3}$$

where

$$P(\mathbf{z}^{\mu}|\mathbf{x}^{\mu},\theta^{(t)}) = \frac{p(\mathbf{x}^{\mu},\mathbf{z}^{\mu}|\theta^{(t)})}{\sum_{\mathbf{z}^{\mu}} p(\mathbf{x}^{\mu},\mathbf{z}^{\mu}|\theta^{(t)})}. \tag{4}$$

The EM algorithm goes as below.
[EM Algorithm]

1. Initialize $\theta^{(0)}$ and $t \leftarrow 0$.
2. Iterate the following EM-step until convergence.

   E-step: Compute $Q(\theta|\theta^{(t)})$ by computing the posterior $P(\mathbf{z}^{\mu}|\mathbf{x}^{\mu},\theta^{(t)})$.

   M-step: $\theta^{(t+1)} = arg\max_{\theta} Q(\theta|\theta^{(t)})$ and $t \leftarrow t+1$.

It can be shown that the EM iteration makes the likelihood $L(\theta)$ increase monotonically; that is, $L(\theta^{(t+1)}) \geq L(\theta^{(t)})$, which means $\{\theta^{(t)}\}$ converges to a local maximum.

2.2 MLP Learning Methods

In this paper we employ three MLP learning methods: BP, BPQ, and SSF.

The BP algorithm (Rumelhart et al., 1986) is well-known method of MLP learning. BP uses only the gradient and goes in an online mode, thus, called stochastic descent. BP is beautifully simple and easily adaptable to many layers, being used in deep learning (Goodfellow et al., 2016).

Although BP is widely used, its learning is usually very slow in convergence and its capability to find excellent solutions is quite limited; thus, to accelerate the convergence and improve the limited capability, several learning methods such as quasi-Newton and conjugate gradient were proposed (Luenberger, 1984). Here we employ quasi-Newton method called BPQ (BP based on quasi-Newton) (Saito and Nakano, 1997). BPQ uses the BFGS update to get a search direction, and uses 2nd-order approximation to get a suitable step length. Although getting a suitable step length usually requires a lot of time, 2nd-order approximation can be carried out very quickly.

Recently SSF (singularity stairs following) has been proposed as a very powerful learning method of single MLPs (Satoh and Nakano, 2013). SSF successively learns MLPs to stably and systematically find excellent solutions, making good use of singular regions generated by using the optimal solution of one-step smaller model MLP($J-1$), and guaranteeing monotonic decrease of training errors.

2.3 Model Selection

In the context of mixture of regressions, we consider many candidates of mixture models; thus, we need a criterion to evaluate the desirability of each candidate and to select the best model. For this purpose we make use of *information criterion*, which indicates a trade-off between learning error and model complexity. Although many information criteria have been proposed so far, we employ the Bayesian information criterion BIC (Schwarz, 1978), because BIC stably showed good performance on MLP model selection (Satoh and Nakano, 2017). BIC was proposed for regular models, but it also works rather well for singular models such as MLPs.

Let $p(\mathbf{x}|\theta)$ be a learning model with parameter vector θ. Given data $\{\mathbf{x}^{\mu},\ \mu = 1,\cdots,N\}$, the log-likelihood is defined as follows:

$$L(\theta) = \sum_{\mu=1}^{N} \log p(\mathbf{x}^{\mu}|\theta). \tag{5}$$

Let $\widehat{\theta}$ be a maximum likelihood estimate. BIC is obtained as an estimator of free energy $F(D)$ shown below. Here $p(D)$ is called evidence and $p(\theta)$ is a prior distribution of θ.

$$F(D) = -\log p(D), \tag{6}$$

$$p(D) = \int p(\theta) \prod_{\mu=1}^{N} p(\mathbf{x}^{\mu}|\theta)\, d\theta \tag{7}$$

BIC is derived using the asymptotic normality and Laplace approximation.

$$\begin{aligned} \text{BIC} &= -2L(\widehat{\theta}) + M \log N \\ &= -2\sum_{\mu} \log p(\mathbf{x}^{\mu}|\widehat{\theta}) + M \log N \end{aligned} \tag{8}$$

BIC can be calculated using only one point ML estimate $\widehat{\theta}$. Here M is the number of parameters.

We consider another important measure for regression: *goodness of fit*. Total sum of squares (TSS) indicates how much variation the data have, residual sum of squares (RSS) indicates the discrepancy between the data and the estimates, and explained sum of squares (ESS) indicates how well a regression model represents the data. Given data $\{(\mathbf{x}^{\mu}, y^{\mu}), \mu = 1, \cdots, N\}$, TSS, RSS, ESS are given as below, where $\mathbf{x}$ are explanatory variables, y is a dependent variable, $f^{\mu} = f(\mathbf{x}^{\mu})$ is an estimate obtained by a regression function, and $\bar{y}$ is a mean of y.

$$\text{TSS} = \sum_{\mu}(y^{\mu} - \bar{y})^2, \quad \text{RSS} = \sum_{\mu}(f^{\mu} - y^{\mu})^2 \tag{9}$$

$$\text{ESS} = TSS - RSS \tag{10}$$

Thus, ESS/TSS (= 1− RSS/TSS) is an important measure indicating goodness of fit of a regression model. It is called *coefficient of determination* in the linear regression context.

3 MIXTURE OF MLP REGRESSIONS

3.1 Model of MoMR

This subsection formalizes the model of MoMR.

Let $\mathbf{x} = (x_1, \cdots, x_K)^T$ be K explanatory variables, and y denotes a dependent variable. In this paper $\mathbf{a}^T$ denotes the transpose of $\mathbf{a}$.

Given data $\{(\mathbf{x}^{\mu}, y^{\mu}), \mu = 1, \cdots, N\}$, we consider a mixture of C regression functions. Let $f(\mathbf{x}|\mathbf{w}_c)$ be a regression function of class c, where $\mathbf{w}_c$ is the weight vector. Since each regression function is supposed to have a constant term, we extend a vector of explanatory variables to get $\widetilde{\mathbf{x}} = (1, x_1, \cdots, x_K)^T$.

Since we consider regression, each MLP has only one output unit. Let MLP of class c be MLP$_c$, which has J_c hidden units, weights $\mathbf{w}_j^{(c)}$ between all input units and hidden unit j(=1,$\cdots$,J), and a weight $v_j^{(c)}$ between hidden unit j (=0,1,$\cdots$,J) and an output unit. Then MLP$_c$ regression function is defined as follows. Here $\mathbf{w}_c = (v_0^{(c)}, v_1^{(c)}, \cdots, v_{J_c}^{(c)}, (\mathbf{w}_1^{(c)})^T, \cdots, (\mathbf{w}_{J_c}^{(c)})^T)^T$ for $c = 1, \cdots, C$, and $\sigma(h)$ denotes the sigmoid activation function.

$$f(\mathbf{x}|\mathbf{w}_c) = v_0^{(c)} + \sum_{j=1}^{J_c} v_j^{(c)}\, \sigma\left((\mathbf{w}_j^{(c)})^T \widetilde{\mathbf{x}}\right) \tag{11}$$

When $J_c = 1$, we consider a linear regression function, which is defined as follows.

$$f(\mathbf{x}|\mathbf{w}_c) = \mathbf{w}_c^T\, \widetilde{\mathbf{x}} \tag{12}$$

We assume the value of y is generated by adding a Gaussian noise ε_c to a value of $f(\mathbf{x}|\mathbf{w}_c)$; here, ε_c is supposed to follow the Gaussian with mean 0 and variance σ_c^2.

$$\varepsilon_c \sim \mathcal{N}(0, \sigma_c^2) \tag{13}$$

The dependent variable y follows the following distribution.

$$y \sim \mathcal{N}(f(\mathbf{x}|\mathbf{w}_c),\ \sigma_c^2) \tag{14}$$

Let π_c be the mixing coefficient of class c. Then, the density of complete data is described as follows.

$$p(y, c|\theta_c) = \pi_c\, g_c(y|f(\mathbf{x}|\mathbf{w}_c), \sigma_c^2) \tag{15}$$

Here $g(u|m, s^2)$ denotes a density function where u follows one-dimensional Gaussian with mean m and variance s^2.

$$g(u|m, s^2) = \frac{1}{\sqrt{2\pi}\, s} \exp\left(-\frac{(u-m)^2}{2\, s^2}\right) \tag{16}$$

The density of incomplete data is written as follows.

$$p(y|\theta) = \sum_{c=1}^{C} p(y, c|\theta_c) = \sum_{c} \pi_c\, g_c(y|f(\mathbf{x}|\mathbf{w}_c), \sigma_c^2) \tag{17}$$

Here θ is a vector comprised of all parameters, where θ_c is a vector of class c parameters.

$$\theta = (\theta_1^T, \cdots, \theta_c^T, \cdots, \theta_C^T)^T, \quad \theta_c = (\pi_c,\ \mathbf{w}_c^T,\ \sigma_c^2)^T \tag{18}$$

3.2 EM Solver of MoMR

Bishop describes the framework to solve soft mixture of linear regressions (Bishop, 2006). We extend Bishop's framework to solve soft mixture of nonlinear regressions, including MoMR.

Since class c is a latent variable and cannot be observed, we employ the EM algorithm (Dempster et al., 1977) as a basic learning method to solve the problem.

Posterior probability is written as follows, where $P(c|y,\theta)$ indicates the probability that y belongs to class c under parameter vector θ.

$$P(c|y,\theta) \quad = \quad \frac{p(y,c|\theta)}{\sum_c p(y,c|\theta)} \tag{19}$$

Given data $D = \{(\mathbf{x}^\mu, y^\mu), \mu = 1,\cdots,N\}$, the incomplete-data log-likelihood is defined as below.

$$L(\theta) \quad = \quad \sum_{\mu=1}^{N} \log\, p(y^\mu|\theta) \tag{20}$$

The Q-function to maximize is shown as below. Here $\theta^{(t)}$ denotes the estimate obtained at the t step of the EM algorithm, and let $f_c^\mu \equiv f(\mathbf{x}^\mu|\mathbf{w}_c)$.

$$\begin{aligned} Q(\theta|\theta^{(t)}) &= \sum_{\mu=1}^{N}\sum_{c=1}^{C} P(c|y^\mu,\theta^{(t)})\ \log\ p(y^\mu,c|\theta) \\ &= \sum_\mu \sum_c P_c^{\mu(t)}\ \log(\pi_c\, g_c(y^\mu|f_c^\mu,\sigma_c^2)) \\ &= \sum_\mu \sum_c P_c^{\mu(t)} \left(\log\pi_c - \frac{1}{2}\log(2\pi) \right. \\ &\qquad \left. - \log\sigma_c - \frac{(y^\mu - f_c^\mu)^2}{2\sigma_c^2} \right) \end{aligned} \tag{21}$$

In the above, we use the following for brevity.

$$P_c^{\mu(t)} \equiv P(c|y^\mu,\theta^{(t)}) = \frac{\pi_c^{(t)}\ g_c^{\mu(t)}}{\sum_c \pi_c^{(t)}\ g_c^{\mu(t)}} \tag{22}$$

$$\text{where } g_c^{\mu(t)} \equiv g_c(y^\mu|f_c^{\mu(t)},\sigma_c^{2(t)}) \tag{23}$$

When we maximize the Q-function, we use the Lagrange method since there is an equality constraint $\sum_c \pi_c = 1$. The Lagrangian function can be written as follows with λ as a Lagrange multiplier.

$$J = Q(\theta|\theta^{(t)}) - \lambda\left(\sum_c \pi_c - 1\right) \tag{24}$$

The necessary condition for a local maximizer is shown below for $c = 1,\cdots,C$.

$$\frac{\partial J}{\partial \pi_c} \quad = \quad \sum_\mu P_c^{\mu(t)}/\pi_c - \lambda = 0 \tag{25}$$

$$\frac{\partial J}{\partial \mathbf{w}_c} \quad = \quad \sum_\mu P_c^{\mu(t)} \frac{(y^\mu - f_c^\mu)}{\sigma_c^2}\frac{\partial f_c^\mu}{\partial \mathbf{w}_c} = \mathbf{0} \tag{26}$$

$$\frac{\partial J}{\partial \sigma_c} \quad = \quad \sum_\mu P_c^{\mu(t)} \left(-\frac{1}{\sigma_c} + \frac{(y^\mu - f_c^\mu)^2}{\sigma_c^3}\right) = 0 \tag{27}$$

Since we have $\lambda = N$ from eq.(25) and the equality constraint, a new estimate of π_c is given below.

$$\pi_c^{(t+1)} = \frac{1}{N}\sum_\mu P_c^{\mu(t)} \tag{28}$$

From eq.(27) a new estimate of σ_c^2 is given below.

$$(\sigma_c^2)^{(t+1)} = \sum_\mu P_c^{\mu(t)}(y^\mu - f_c^\mu)^2 \Big/ \sum_\mu P_c^{\mu(t)} \tag{29}$$

From eq.(26) we obtain a new estimate of $\mathbf{w}_c$ by solving the following.

$$\sum_\mu P_c^{\mu(t)}(y^\mu - f_c^\mu)\frac{\partial f_c^\mu}{\partial \mathbf{w}_c} = \mathbf{0} \tag{30}$$

Note that the condition eq.(30) is equal to the following optimal condition of $E_c(\mathbf{w}_c)$.

$$\frac{\partial E_c(\mathbf{w}_c)}{\partial \mathbf{w}_c} \quad = \quad \mathbf{0}. \tag{31}$$

Here the following is sum-of-squares error of class c.

$$E_c(\mathbf{w}_c) \quad = \quad \frac{1}{2}\sum_\mu P_c^{\mu(t)}(f_c^\mu - y^\mu)^2 \tag{32}$$

Residual sum of squares (RSS) in MoMR is given as below.

$$\mathrm{RSS} = 2\sum_c E_c(\mathbf{w}_c) = \sum_\mu \sum_c P_c^{\mu(t)}(f_c^\mu - y^\mu)^2 \tag{33}$$

In $E_c(\mathbf{w}_c)$, squared error $(f_c^\mu - y^\mu)^2$ for data point μ is weighted by posterior probability $P_c^{\mu(t)}$. Thus, in MLP$_c$ learning, the gradient for data point μ should be weighted by posterior $P_c^{\mu(t)}$. This modification should be embodied in both BP and BPQ.

The learning of MoMR is carried out in a double-looped manner: the outer loop is controlled by the EM algorithm and the inner loop is controlled by MLP learning BP or BPQ.

3.3 Model Selection of MoMR

This subsection describes how BIC is calculated in MoMR.

The density of incomplete data is given by eq.(17). Then, incomplete-data log-likelihood at the optimal point is given as follows.

$$\begin{aligned} L(\theta) &= \sum_{\mu=1}^{N} \log\ p(y^\mu|\theta) \\ &= \sum_\mu \log\left[\sum_c \pi_c\ g_c(y^\mu|f(\mathbf{x}^\mu|\theta_c),\sigma_c^2)\right] \end{aligned} \tag{34}$$

Hence, BIC in MoMR is obtained as below.

$$\begin{aligned} \mathrm{BIC} = &-2\sum_\mu \log\left[\sum_c \pi_c\ g(y^\mu|f(\mathbf{x}^\mu|\widehat{\theta}_c),\sigma_c^2)\right] \\ &+ M\ \log N \end{aligned} \tag{35}$$

Here M, the number of parameters, is calculated as follows. We should not forget to count two parameters π_c and σ_c in calculating M_c.

$$M = \sum_c M_c, \quad M_c = \begin{cases} K+3 & \text{if } J_c = 1 \\ J_c(K+2)+3 & \text{if } J_c \geq 2 \end{cases} \tag{36}$$

TSS, RSS and ESS in MoMR are considered here. Since, in MoMR, data point μ is weighted by posterior probability $P_c^{\mu(t)}$, TSS, RSS and ESS are calculated as follows.

$$\begin{aligned} \text{TSS} &= \sum_\mu \sum_c P_c^{\mu(t)} (y_c^\mu - \bar{y})^2 \quad (37) \\ \text{RSS} &= \sum_\mu \sum_c P_c^{\mu(t)} (f_c^\mu - y^\mu)^2 \quad (38) \\ \text{ESS} &= TSS - RSS \quad (39) \end{aligned}$$

ESS/TSS in MoMR is calculated using the above.

4 EXPERIMENTS

4.1 Design of Experiments

The following 26 models are considered for each dataset. MLP(J=1) is reduced to linear regression here. Models are given model numbers, which are used in the figures and explanations shown later.

(a) Models 1 to 10: 10 single MLP(J) regressions: $J = 1, \cdots, 10$,

(b) Models 11 to 16: 6 mixtures of MLP(J_1) and MLP(J_2) regressions: (J_1,J_2) = (1,1), (1,2), (1,3), (2,2), (2,3), (3,3),

(c) Models 17 to 26: 10 mixtures of MLP(J_1), MLP(J_2) and MLP(J_3) regressions: (J_1,J_2,J_3) = (1,1,1), (1,1,2), (1,1,3),(1,2,2), (1,2,3), (1,3,3), (2,2,2), (2,2,3), (2,3,3), (3,3,3).

The learning method of MoMR is double-looped: the outer loop is controlled by the EM algorithm, and the inner loop by MLP learning method BP or BPQ. As for the learning of mixture of linear regressions (MoLR), refer to (Nakano and Satoh, 2018). A single MLP(J) regression is learned by SSF or BP if $J \geq 2$.

Parameters of BP and BPQ are selected through our preliminary experiments, as shown in Table 1. Very weak regularization of weight decay is employed to prevent weight values from getting huge. Note that the maximum of sweeps per EM loop needs not be large since posterior probabilities may change during EM learning. For SSF, maximum of search tokens is set to be 20. We used a PC having Xeon(R)E5 3.7GHz with 8GB memory for computation.

Table 1: MLP parameters for the experiments

Parameter	BP	BPQ
max sweeps/EM loop (MoMR)	500	500
learning rate (MoMR)	0.05	adaptive
weight decay coeff (MoMR)	10^{-7}	10^{-6}
max sweeps (Single)	5000	5000
learning rate (Single)	0.05	adaptive
weight decay coeff (Single)	10^{-7}	10^{-6}

4.2 Experiments Using Artificial Data

We generated one-dimensional two-class artificial data to see how MoMR works. The following two parabolas were used to generate 51 data points for each class by adding small Gaussian noise $\mathcal{N}(0, 0.035^2)$. The range of x_1 is $[0.1, 1.0]$.

$$\begin{aligned} y_1 &= -4(x_1 - 0.6)^2 + 2.0 \quad (40) \\ y_2 &= -2(x_1 - 0.6)^2 + 1.5 \quad (41) \end{aligned}$$

Figure 1 shows two parabolas and 102 data points. Since MLP($J \geq 2$) can fit a parabola well, two MLPs(J=2) are expected to fit the artificial data well as the minimal model.

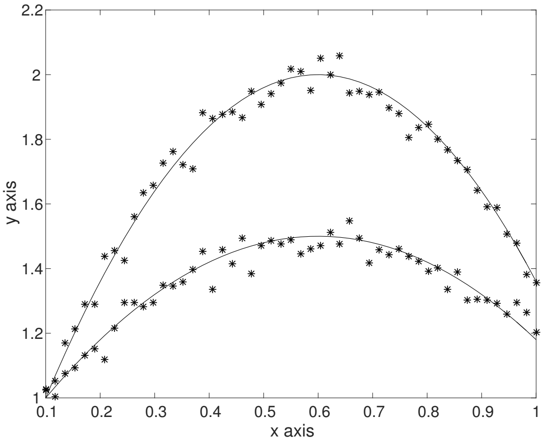

Figure 1: Artificial data with two generating parabolas

Figure 2 compares BIC of each model for artificial data. Horizontal axis indicates model number. BIC obtained by EM+BPQ was always smaller (better) than the corresponding BIC by EM+BP except linear Models 1, 11 and 17. This was caused by BP's weak capability to find excellent solutions. BIC obtained by EM+BPQ selected Model 14, two MLPs(J=2), as the best, which we expected. On the other hand, BIC obtained by EM+BP selected unexpected Model 20, one linear and two MLPs(J=2), which is larger than the optimal Model 14. As the best mixture of linear

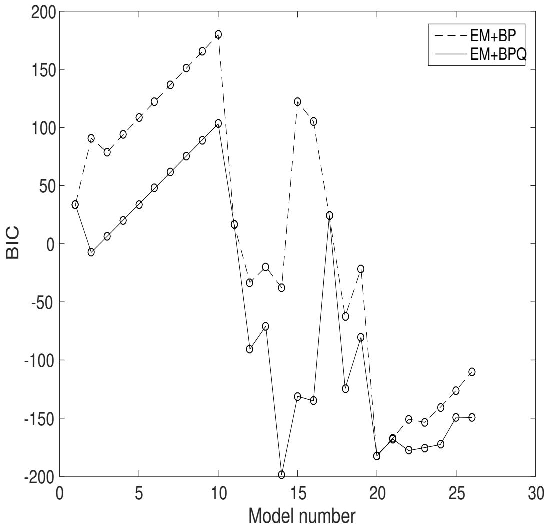

Figure 2: BIC comparison for artificial data

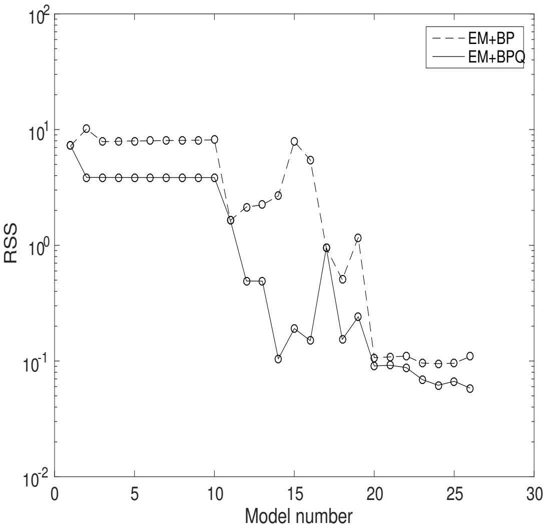

Figure 3: RSS comparison for artificial data

regressions, BIC selected Model 11, which is composed of two lines. Among single regression models, BIC obtained by SSF selected Model 2, MLP(J=2), while BIC by BP selected wrong Model 1, linear regression.

Figure 3 compares residual sum of squares (RSS) of each model for artificial data. It can be seen that the solid line (EM+BPQ) always obtained smaller RSS than the dotted line (EM+BP) except linear Models 1, 11 and 17. RSS of Model 14, the best model obtained by EM+BPQ, was 0.1046 and thus its goodness of fit $1-\text{RSS}/\text{TSS}$ was very high 0.9867 since TSS = 7.8436 for artificial data. The solid line indicates that mixture models had much smaller RSS than single models. Among mixture models, the solid line also indicates that MoMR models had much smaller RSS than mixture of linear regressions, Models 11 and 17. Hence we can say MoMR effectively improved goodness of fit compared with single regression models or mixtures of linear regressions.

Figure 4 depicts the best model (Model 14) among all the models obtained by EM+BPQ (or SSF, MoLR); Model 14 is composed of two MLPs(J=2). We can see these two curves are very close to the original parabolas. The parameters of Model 14 obtained by EM+BPQ are shown below.

$$\widehat{\pi}_1 = 0.504, \quad \widehat{\pi}_2 = 0.496 \tag{42}$$

$$\begin{aligned}\widehat{f}_1 &= 0.642 - 0.327\,\sigma(-10.707 + 12.563x_1) \\ &\quad +0.881\,\sigma(-1.107 + 7.602x_1)\end{aligned} \tag{43}$$

$$\begin{aligned}\widehat{f}_2 &= -11.804 + 16.542\,\sigma(1.272 + 2.040x_1) \\ &\quad -22.829\,\sigma(-3.704 + 1.738x_1)\end{aligned} \tag{44}$$

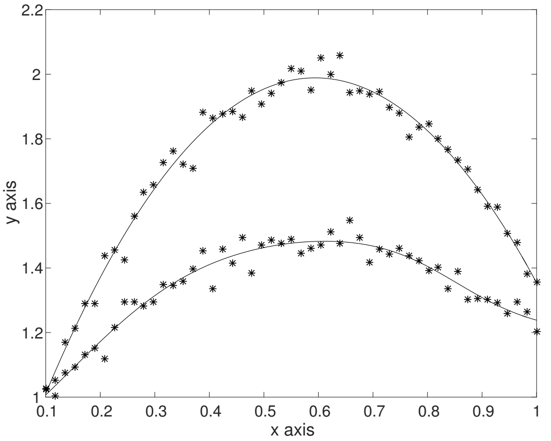

Figure 4: Best MoMR model obtained by EM+BPQ for artificial data

Figure 5 indicates how RSS decreased through EM learning in the best Model 14, two MLPs(J = 2) obtained by EM+BPQ. The error decreased very smoothly and monotonically.

Figure 6 shows the best model (Model 20) among all the models obtained by EM+BP (or BP), which is composed of one linear and two MLPs(J=2). The parameters of Model 20 obtained by EM+BP are shown

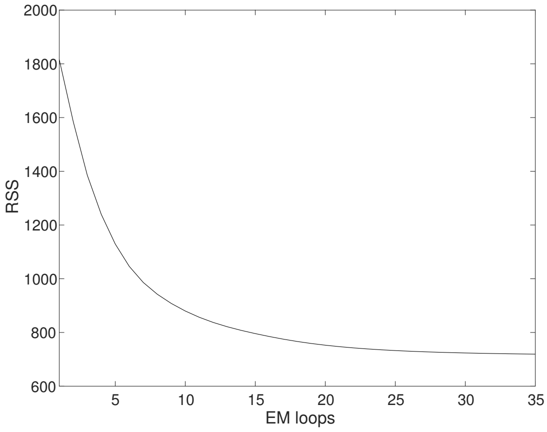

Figure 5: EM learning of best Model 14 for artificial data

below.

$$\begin{aligned}\widehat{\pi}_1 &= 0.052,\ \widehat{\pi}_2 = 0.455,\ \widehat{\pi}_3 = 0.492 && (45)\\ \widehat{f}_1 &= 1.062 + 0.678x_1 && (46)\\ \widehat{f}_2 &= 0.556 - 2.672\ \sigma(-1.757 + 1.804x_1) \\ &\quad +2.088\ \sigma(-0.720 + 4.633x_1) && (47)\\ \widehat{f}_3 &= -0.458 + 2.974\ \sigma(-0.412 + 4.993x_1) \\ &\quad -5.107\ \sigma(-4.963 + 3.679x_1) && (48)\end{aligned}$$

Since the mixing coefficient of linear regression was very small and two MLP regressions fitted well, we will have the optimal two MLPs(J=2) if we neglect the linear regression and relearn.

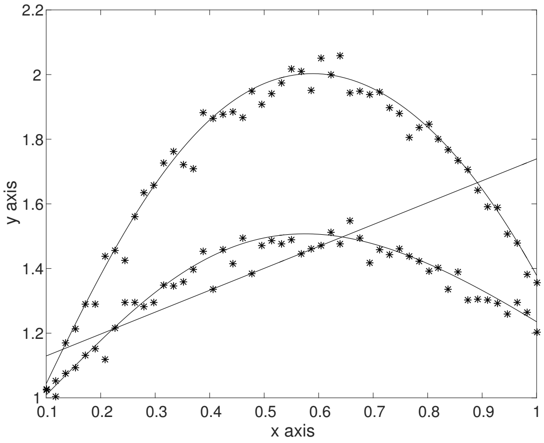

Figure 6: Best MoMR model obtained by EM+BP for artificial data

Figure 7 depicts the best model (Model 11) among mixtures of linear regressions, which is composed of two lines. Its BIC was larger (worse) than that of the best single MLP model, which means the best single MLP model is more preferable than the best mixture of linear regressions. The obtained parameters of Model 11 are shown below.

$$\begin{aligned}\widehat{\pi}_1 &= 0.680, \quad \widehat{\pi}_2 = 0.320 && (49)\\ \widehat{f}_1 &= 1.221 + 0.262x_1 && (50)\\ \widehat{f}_2 &= 1.753 + 0.209x_1 && (51)\end{aligned}$$

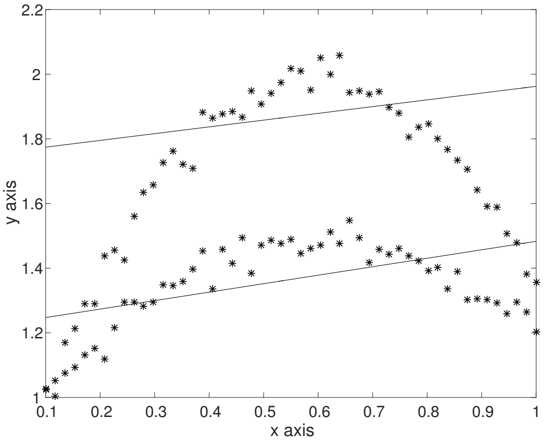

Figure 7: Best mixture of linear regressions for artificial data

Figure 8 shows the best model (Model 2) among single regression models, which runs in a middle empty space between two parabolas. The obtained parameters of Model 2 are shown below.

$$\begin{aligned}\widehat{f} &= -3.059 + 5.314\ \sigma(0.874 + 3.304x_1) \\ &\quad -1.893\ \sigma(-4.754 + 4.619x_1) && (52)\end{aligned}$$

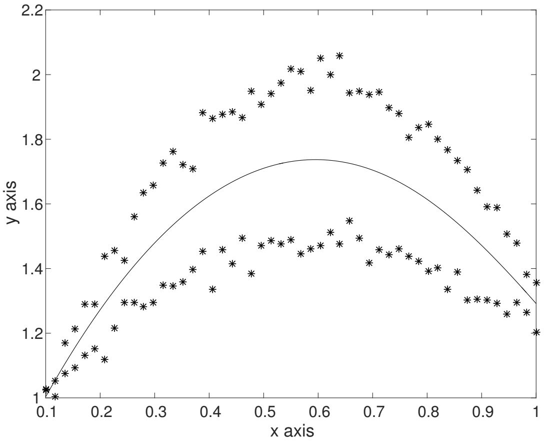

Figure 8: Best single regression for artificial data

The CPU time required to get the results for artificial data is compared. As average CPU time required to learn 16 MoMR models per initialization, EM+BPQ required 1m 12s, while EM+BP required 7m 16s. Although BPQ computes more information than BP, its average CPU time was smaller because it converged faster for this dataset.

4.3 Experiments Using Real Data

As real data we used Abalone dataset from UCI Machine Learning Repository. We selected this dataset because any single powerful regression model cannot fit well. The dataset has seven numerical explanatory variables and the number of data points $N = 4177$.

Figure 9 compares BIC of each model for Abalone data. It can be seen that BIC obtained by EM+BPQ was always much smaller (better) than the corresponding BIC by EM+BP except three linear models. BIC(EM+BPQ) selected Model 20, one linear and two MLPs(J=2), as the best, while BIC(EM+BP) selected inadequate Model 17, mixture of two linear regressions, as the best. The parameters of the best Model 20 obtained by EM+BPQ are shown below.

$$\widehat{\pi}_1 = 0.332,\ \widehat{\pi}_2 = 0.210,\ \widehat{\pi}_3 = 0.458 \qquad (53)$$

$$\begin{aligned} \widehat{f}_1 &= -2.340 + 0.948x_1 + 1.426x_2 \\ &\quad +3.311x_3 + 1.489x_4 - 2.552x_5 \\ &\quad +0.117x_6 + 0.713x_7 \qquad (54) \\ \widehat{f}_2 &= -138.885 + 142.380\ \sigma(2.997 + 3.082x_1 \\ &\quad -2.201x_2 + 0.568x_3 + 7.902x_4 \\ &\quad -5.468x_5 + 0.084x_6 + 2.232x_7) \qquad (55) \\ &\quad -3.138\ \sigma(-4.091 + 17.621x_1 - 7.847x_2 \\ &\quad -8.219x_3 - 42.627x_4 + 36.819x_5 \\ &\quad +10.864x_6 - 0.422x_7) \qquad (56) \\ \widehat{f}_3 &= -109.338 + 111.663\ \sigma(3.366 + 1.007x_1 \\ &\quad -0.306x_2 - 0.432x_3 - 0.283x_4 \\ &\quad +0.051x_5 - 0.782x_6 + 3.334x_7) \\ &\quad -1.086\ \sigma(2.879 + 39.533x_1 - 32.959x_2 \\ &\quad -83.840x_3 - 179.304x_4 + 209.318x_5 \\ &\quad +17.659x_6 + 18.004x_7) \qquad (57) \end{aligned}$$

Note that Model 20 had smaller BIC than any single model or any mixture of linear regressions. Among single models MLP(J=7) is the best single model.

Figure 10 compares RSS of each model for Abalone data. We can see that EM+BPQ always obtained much smaller RSS than EM+BP except three linear models. RSS of the best single model MLP(J=7) was 1543.36, then the goodness of fit, coefficient of determination, was 1−RSS/TSS = 0.6304, which is not so high. Note that TSS = 4176 for normalized Abalone data. RSS of Model 20, the best model among all the models obtained by EM+BPQ, was 727.32, then the goodness of fit was 1−RSS/TSS = 0.8258, showing nice fitting. RSS of Model 17, the best mixture of linear regressions, was 865.32, and its goodness of fit was 0.7928, a bit worse than the best model. Model 24 had the smallest RSS 656.00 among all the models, and its goodness of fit was 0.8429. Goodness of fit for Abalone data can be improved to this level by using MoMR.

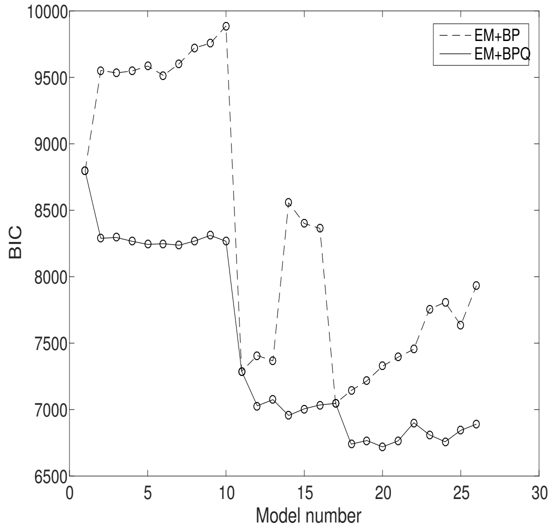

Figure 9: BIC comparison for Abalone data

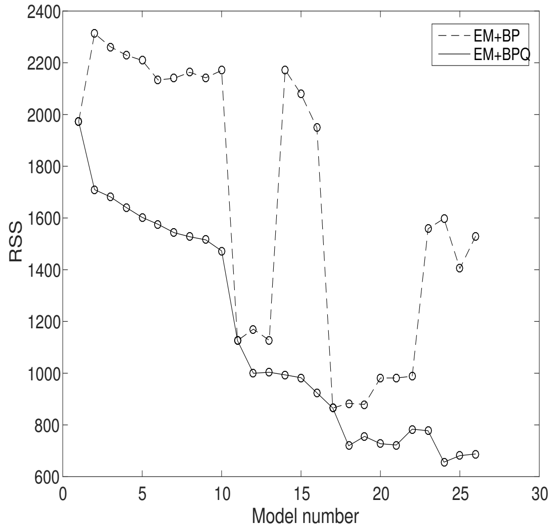

Figure 10: RSS comparison for Abalone data

Figure 11 indicates how RSS decreased through

EM learning in the best Model 20, one linear and two MLPs(J=2) obtained by EM+BPQ. The error decreased again very smoothly and monotonically.

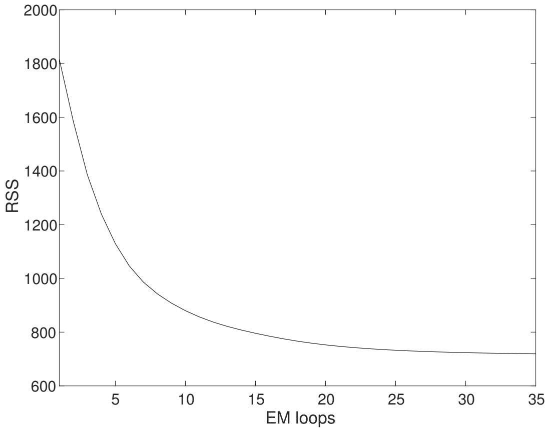

Figure 11: EM learning of best model for Abalone data

The CPU time required to get the results for Abalone data is compared. As average CPU time required to learn 16 MoMR models per initialization, EM+BPQ required 6h 7m 40s, while EM+BP required 4h 46m 37s.

4.4 Considerations

The results of our experiments may suggest the following.

(a) MoMR worked well, selecting the expected model MLPs($J = 2$) as the best for artificial data, and selecting the model composed of one linear and two MLPs as the best for Abalone data. These best models show smaller BIC and RSS values than those of any mixture of linear regressions or any single MLP regression.

(b) The learning of MoMR goes in a double loop; the EM algorithm controls the outer loop and MLP learning method controls the inner loop. As for MLP learning, a quasi-Newton method called BPQ worked well for MoMR, while BP worked rather poorly, frequently finding rather poor solutions, having larger (worse) RSS than BPQ, selecting inadequate models different from those by BPQ. This tendency was caused by BP's weak capability to find excellent solutions.

(c) MoMR using EM+BPQ is expected to improve goodness of fit for data having poor fit by any single regression model or mixture of linear regressions.

5 CONCLUSIONS

This paper proposes model and learning of mixture of MLP regressions (MoMR). The learning of MoMR goes in a double loop; the outer loop is controlled by the EM algorithm and the inner by MLP learning. As for MLP learning in MoMR, a quasi-Newton method called BPQ worked satisfactorily, while BP did not work. Our experiments showed MoMR worked well for artificial and real datasets. In the future we plan to apply MoMR using EM+BPQ to more data to show MoMR can be a useful regression model for noisy data.

ACKNOWLEDGMENT

This work was supported by Grants-in-Aid for Scientific Research (C) 16K00342.

REFERENCES

Bishop, C. M. (2006). *Pattern recognition and machine learning*. Springer.

Chen, Y.-C., Genovese, C., Tibshirani, R., and Wasserman, L. (2016). Nonparametric modal regression. *The Annals of Statistics*, 44(2):489–514.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum-likelihood from incomplete data via the EM algorithm. *J. Royal Statist. Soc. Ser. B*, 39:1–38.

Goldfeld, S. and Quandt, R. (1973). A Markov model for switching regressions. *Journal of Econometrics*, 1(1):3–15.

Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep learning*. MIT Press.

Huang, M., Runze, L., and Shaoli, W. (2013). Nonparametric mixture of regression models. *Journal of the American Association*, 108(503):929–941.

Hurn, M., Justel, A., and Robert, C. (2003). Estimating mixtures of regressions. *Journal of Computational and Graphical Statistics*, 12(1):1–25.

Leisch, F. (2004). FlexMix: A general framework for finite mixture models and latent class regression in R. *Journal of Statistical Software*, 11(8):1–18.

Luenberger, D. G. (1984). *Linear and nonlinear programming*. Addison-Wesley.

McLachlan, G. J. and Peel, D. (2000). *Finite mixture models*. John Wiley & Sons.

Nakano, R. and Satoh, S. (2018). Weak dependence on initialization in mixture of linear regressions. In *Proc. of Int. Conf. on Artificial Intelligence and Applications 2018*, pages 1–6.

NCSS (2013). Regression clustering. Technical Report Chapter 449, pp.1–7, NCSS Statistical Software Documentation.

Qian, G. and Wu, Y. (2011). Estimation and selection in regression clustering. *European Journal of Pure and Applied Mathematics*, 4(4):455–466.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Learning internal representations by error propagation. In *Parallel Distributed Processing, Vol.1*, pages 318–362. MIT Press.

Saito, K. and Nakano, R. (1997). Partial BFGS update and efficient step-length calculation for three-layer neural networks. *Neural Comput.*, 9(1):239–257.

Satoh, S. and Nakano, R. (2013). Fast and stable learning utilizing singular regions of multilayer perceptron. *Neural Processing Letters*, 38(2):99–115.

Satoh, S. and Nakano, R. (2017). How new information criteria WAIC and WBIC worked for MLP model selection. In *Proc. of 6th Int. Conf. on Pattern Recognition Applications and Methods (ICPRAM)*, pages 105–111.

Schwarz, G. (1978). Estimating the dimension of a model. *Annals of Statistics*, 6:461–464.