Inferential Estimation of Polymer Melt Index Using Sequentially Trained Bootstrap Aggregated Neural Networks

Jie Zhang1*, Qibing Jin2, and Yongmao Xu3

1Centre for Process Analytics and Control Technology, School of Chemical Engineering & Advanced Materials, University of Newcastle, Newcastle upon Tyne NE1 7RU, U. K.
2School of Information Science and Technology, Beijing University of Chemical Technology, Beijing 100029, P. R. China
3Department of Automation, Tsinghua University, Beijing 100084, P. R. China

Abstract

Inferential estimation of polymer melt index in an industrial polymerisation process using aggregated neural networks is presented in this paper. The difficult-to-measure polymer melt index is estimated from the easy-to-measure process variables and their relationship is estimated using aggregated neural networks. The individual networks are trained on bootstrap re-samples of the original training data by a sequential training algorithm. In this training method, individual networks within a bootstrap aggregated neural network model are trained sequentially. The first network is trained to minimise its prediction error on the training data. In the training of subsequent networks, the training objective is not only to minimise the individual networks’ prediction errors but also to minimise the correlation among the individual networks. Training is terminated when the aggregated network prediction performance on the training and testing data cannot be further improved. Application to real industrial data demonstrates that polymer melt index can be successfully estimated using an aggregated neural network.

Keywords: Estimation, neural networks, polymerisation, modelling, nonlinear models, process control, product quality, robust estimation.

1. Introduction

Polymer production facilities face increasing pressures to improve product quality and reduce production cost due to the increasing global competition. The advanced monitoring and control of polymerisation processes, in particular the properties of polymer products, is of major strategic importance to the polymer manufacturing industries. In the propylene polymerisation industry, various grades of polypropylene are produced in continuous stirred tank reactors (CSTR). The switch from one grade to another is carried out by changing the reactor operating conditions or by using a different type of catalyst. Polypropylene grade specifications are generally quoted in terms of polymer melt

* Corresponding author, E-mail: jie.zhang@newcastle.ac.uk, Tel.: +44-191-2227240, Fax:+44-191-2225292
index (MI) and density.

A common problem in the polymer industry is the lack of on-line measurements for polymer quality variables. The most critical quality variable of propylene, the melt index, is not easily measured. The lack of fast enough measurement will limit the achievable control performance for polymer quality control. These difficult-to-measure polymer quality variables are related to certain easy-to-measure process variables such as temperatures and hydrogen concentration in the reactor. With this relationship, inferential estimation of polymer quality variables can be achieved from the measurements of the easy-to-measure variables. The key in inferential estimation is therefore to find the relationship between the difficult-to-measure variables and the easy-to-measure variables.

A number of researchers have attempted to find this relationship by studying the polymerisation mechanism and using some state estimation techniques, such as the extended Kalman filter, to obtain inferential estimates of polymer quality variables [1-4]. These approaches, however, require a deep understanding of the polymerisation process and the development of a mechanistic model is generally effort demanding. It is not uncommon for a comprehensive mechanistic polymerisation model to involve several dozens of differential and algebraic equations with many model parameters to be determined.

To overcome this difficulty, inferential estimation of polymer quality based on empirical models has been investigated [5,6]. The empirical models can be developed from process operation data. Since polymerisation processes are usually highly nonlinear processes, nonlinear empirical models should be developed. Neural networks have been shown to be able to approximate any continuous nonlinear functions [7] and have been applied to nonlinear system modelling and control [8]. However, if not properly trained and validated, a neural network model may lead to undesired solutions to the engineering problems. A major problem in neural network based modelling is over-fitting and the resulting lack of generalisation capability. To overcome this problem, a number of techniques have been developed including regularisation [9], network pruning [10], early stopping [11], constructive approaches to network building [12], and multiple networks [13-20]. Of these techniques, multiple networks techniques have emerged as a powerful solution to many engineering problems [18,21].

The key of the successful applications of multiple neural networks lies in the ways that the individual networks are developed and combined. Studies of Zhang et al. [18,21] show that significant improvement in model generalisation can be achieved when the individual networks to be combined are independent (or less correlated). It is shown that combining $M$ independent neural network models can reduce the average model error by $M$ fold [22]. To obtain less correlated neural network models, Zhang et al. [18] proposed using bootstrap re-sampling to generate training data for individual networks so that they are trained on different data sets and combining the individual networks using
principal component regression (PCR) so that the independent information captured by the individual networks is focused. The studies of Merz and Pazzani [23] further demonstrate the advantage of combining individual models using PCR.

Another issue in developing multiple neural network models is how many individual networks should be combined. The studies of Zhang et al. [18,21] show that the generalisation performance of a stacked neural network usually stabilises after stacking around 10 to 20 individual neural networks. The development of a large number of individual networks is usually for the sake of getting a diverse set of individual models. Much computational effort in developing those individual networks could be saved if individual networks are trained co-operatively so that a small number of neural networks that are less correlated are developed. Zhou et al. [19] show that it would be better to ensemble many instead of all the networks at hand and propose a genetic algorithm based technique to select the networks to be combined.

In order to enhance the model performance of aggregated neural networks, a sequential training method for developing aggregated neural network models is presented in this paper. In this method, individual networks are trained sequentially (and co-operatively) on bootstrap re-samples of the original training data. For developing each individual network, a replication of the original training data is generated using bootstrap re-sampling with replacement and serves as the training data for that network. The first network is trained to minimise its prediction error whereas the rest of the networks are trained not only to minimise their prediction errors but also minimise the correlation among the trained networks. In such a way, training of individual networks is co-ordinated in that the information captured by earlier trained networks is considered in the training of later networks. Training of further individual networks can be terminated when the aggregated network performance on the original training and testing data cannot be further improved.

The paper is organised as follows. Section 2 presents bootstrap aggregated neural networks and a sequential training algorithm. Inferential estimation of polymer melt index in an industrial plant is given in Section 3. The last section concludes this paper.

2. Bootstrap Aggregated Neural Networks and the Sequential Training Algorithm

2.1 Bootstrap Aggregated Neural Networks

A diagram of a bootstrap aggregated neural network is shown in Figure 1, where several neural network models are developed to model the same relationship between the input and output variables and are combined together. Instead of selecting a single neural network model, a bootstrap aggregated model combines several neural network models to improve model accuracy and robustness. The
The overall output of a bootstrap aggregated neural network is a weighted combination of the individual neural network outputs. This can be represented by the following equation:

$$F(x) = \sum_{i=1}^{M} \theta_i F_i(x) \quad (1)$$

where $F_i$ represents the $i$th individual network, $x$ is a vector of network inputs, $M$ is the number of individual networks, and $\theta_i$ is the aggregating weight for combining the $i$th individual network, which can be obtained from PCR or simple average.

### 2.2 Sequential Training Algorithm

In this paper, for the simplicity in illustration, simple average is used in combining neural networks and $\theta_1 = \ldots = \theta_M = 1/M$. For the training of the first network, the training objective is simply to minimise the network prediction errors as follows:

$$J_1 = \frac{1}{2N} \sum_{j=1}^{N} (F_1(x_j) - d(x_j))^2 \quad (2)$$

where $N$ is the number of training data points (samples), $d$ is the desired model output, and $x_j$ is the $j$th training data point. This network can be trained using one of a number of training algorithms, such as back propagation and Levenberg-Marquardt optimisation.

After training of the first network, subsequent networks are trained and added to the ensemble of networks. Those networks are trained on bootstrap re-samples of the original training data to encourage diversity among the networks [18,21]. When training these networks, the training objective should not only minimise the individual networks’ prediction errors but also minimise the correlation among the individual networks.

In this study, the training algorithm for the subsequent networks is developed as follows. At the $i$th training stage (training the $i$th individual network and $i > 1$),

$$F(x) = \frac{1}{i-1} \sum_{k=1}^{i-1} F_k(x) \quad (3)$$

and

$$J_i = \frac{1}{2N} \sum_{j=1}^{N} (F_i(x_j) - d(x_j))^2 - \frac{\lambda}{2N} \sum_{j=1}^{N} (F_i(x_j) - F(x_j))^2 \quad (4)$$

In the above training objective function, the first term minimises the prediction error of the $i$th neural network whereas the second term minimises the correlation between the $i$th neural network and the previously trained networks, and $\lambda$ is a weighting parameter for the second term.
The derivative of \( J_i \) with respect to the weights of the \( i \)th network, \( W_i \), can be calculated as:

\[
\frac{\partial J_i}{\partial W_i} = \frac{1}{N} \sum_{j=1}^{N} (F_i(x_j) - d(x_j)) \frac{\partial F_i(x_j)}{\partial W_i} - \frac{\lambda}{N} \sum_{j=1}^{N} (F_i(x_j) - F(x_j)) \frac{\partial F_i(x_j)}{\partial W_i}
\]

\[
= \frac{1}{N} \sum_{j=1}^{N} \{(F_i(x_j) - d(x_j)) - \lambda(F_i(x_j) - F(x_j))\} \frac{\partial F_i(x_j)}{\partial W_i}
\]

\[
= \frac{1}{N} \sum_{j=1}^{N} \{(1 - \lambda)F_i(x_j) - d(x_j) + F(x_j))\} \frac{\partial F_i(x_j)}{\partial W_i}
\]

(5)

If a back propagation type network training algorithm is used, then the network weights are adjusted as:

\[
\Delta W_i(k+1) = \alpha \Delta W_i(k) - \eta \frac{\partial J_i}{\partial W_i(k)}
\]

(6)

\[
W_i(k+1) = W_i(k) + \Delta W_i(k+1)
\]

(7)

If the Levenberg-Marquardt optimisation algorithm is used, then the weight adjustment is calculated as:

\[
\Delta W_i(k+1) = -\eta \left[ \frac{1}{N} \sum_{j=1}^{N} \frac{\partial F_i(x_j)}{\partial W_i(k)} \left( \frac{\partial F_i(x_j)}{\partial W_i(k)} \right)^T + \delta I \right]^{-1} \frac{\partial J_i}{\partial W_i(k)}
\]

(8)

In Eq(6) to Eq(8), \( W_i(k) \) and \( \Delta W_i(k) \) are the vectors of weights and weight adaptations for training the \( i \)th network at training step \( k \) respectively, \( \alpha \) is the momentum coefficient, \( \eta \) is the learning rate, and \( \delta \) is a parameter to control the searching step size. A large value of \( \delta \) gives a small step in the gradient direction and a small value of \( \delta \) gives a searching step close to the Gauss-Newton step.

Training of the \( i \)th individual network can be terminated when the error gradient is less than a pre-specified value, e.g. \( 10^{-6} \). Training can also be terminated by a cross validation based “early stopping” criterion. When using a cross validation based stopping criterion, data for building a neural network model is divided into a training data set and a testing data set. During network training, the network prediction error on the testing data is continuously monitored. Training of an individual network is terminated when the testing error stops decreasing. Training of the aggregated network can be terminated when the aggregated network performance on the original training and testing data cannot be further improved. Through this sequential training algorithm, a small number of less correlated networks are developed and combined leading to improved model performance and reduced
computation cost.

3. Neural Network based Inferential Estimation of Melt Index in an Industrial Plant

3.1 Process Description

The process considered here is a propylene polymerisation process located in a plant in China. A highly simplified schematic diagram of this process is illustrated in Figure 2. The process consists of a train of reactors in series, two CSTRs and two fluidised-bed reactors (FBR). The feed to the reactor comprises propylene, hydrogen, and catalyst. A comonomer (ethylene or a higher alpha-olefin) will be fed into the forth reactor when copolymer is produced. These liquids and gases supply reactants for the growing polymer particles and provide the heat transfer media. The melt index and density of the polypropylene in the reactor depend on catalyst properties, reactant composition, and reactor temperature etc. Hydrogen is a regulator of molecular weight of polypropylene. Changing the hydrogen feeding rate can regulate the MI of polypropylene to a desired value.

3.2 Inferential estimation of MI

A set of process operational data covering 31 days of plant operation was provided by the plant personnel. During this period several grades of products were produced. Measurements of melt indices in reactors 1 and 4 were logged every 2 hours through laboratory analysis. Measurements of 30 process variables, such as \( \text{H}_2 \) feed rate and \( \text{H}_2 \) concentration, were logged every half hour. Figure 3 shows the melt index data. Figure 4 shows all the process variables. For the reason of industrial confidentiality, the unit of the variables are disguised in these plots. Not all the 30 process variables are useful in estimating MI and the most relevant process variables are identified using statistical correlation analysis.

In order to enhance process monitoring and control performance, it would be desirable that MI measurements are available at a much shorter interval than 2 hours. Soft sensors for estimating MI in reactors 1 and 4 from the measurements of process variables are to be developed. Not all the 30 process variables are required in estimating MI. The most relevant process variables can be identified using statistical correlation analysis of the data. From statistical correlation analysis of the data, it is found that MI in reactor 1 is closely correlated with the hydrogen concentration in reactor 1 and the hydrogen feed rate to reactor 1, whereas the MI in reactor 4 is closely correlated with the hydrogen concentrations in reactors 1 and 2. Hydrogen feed rates to reactor 1 are plotted in Figure 5 whereas Figure 6 shows the hydrogen concentrations in reactors 1 and 2.

The time delays between the model input and MI can be identified from cross correlation analysis. Figure 7 (a) shows the cross-correlations between the hydrogen concentration in reactor 1 and MI in
reactor 1 and Figure 7 (b) shows the cross-correlations between the hydrogen feed rate to reactor 1 and MI in reactor 1. Figure 8 (a) shows the cross-correlations between the hydrogen concentration in reactor 1 and MI in reactor 4 and Figure 8 (b) shows the cross-correlations between the hydrogen concentration in reactor 2 and MI in reactor 4. Figure 7 (a) indicates that there is no time delay between the hydrogen concentration in reactor 1 and MI in reactor 1 since the maximum cross-correlation occurs at 0 time lag. Figure 7 (b) indicates that the time delay between the hydrogen feed rate to reactor 1 and MI in reactor 1 is about 2 hours since the maximum cross-correlation occurs at a time lag of about -2 hours. Figure 8 indicates that time delays exist between the hydrogen concentrations in both reactors 1 and 2 and MI in reactor 4. Since MI and process variables were sampled as different sampling rates, better estimation of the time delays can be obtained by inspecting the cross-correlations between the time-shifted process variables (shifted by several units of the smaller sampling interval of 0.5 hours) and MI shown in Figure 9. The time delays used in Figure 9 are appropriate since the maximum cross-correlations between the time-shifted process variables and MI occur at 0 time lag.

Models of the following forms were then developed for the inferential estimation of MI

\[ MI_1(t) = f_1[H_1(t), H_1(t-1), H_1(t-2), F(t-9), F(t-10), F(t-11)] \]  \hspace{1cm} (9)

\[ MI_2(t) = f_2[H_1(t-7), H_1(t-8), H_1(t-9), H_2(t-6), H_2(t-7), H_2(t-8)] \]  \hspace{1cm} (10)

where \( MI_1 \) and \( MI_2 \) are, respectively, the MI in reactors 1 and 4, \( H_1 \) and \( H_2 \) are, respectively, the hydrogen concentrations in reactors 1 and 2, \( F \) is the hydrogen feed rate to reactor 1, and \( t \) is the discrete time. The sampling time used in the inferential estimators is 0.5 hours. The variables \( H_1, H_2, \) and \( F \), correspond, respectively, the 2\(^{nd}\) and the 3\(^{rd}\) plots in the last row and the 3\(^{rd}\) plot in the second row of Figure 4.

These industrial data were divided into three sets: a training data set, a testing data set, and an unseen validation data set. Networks were trained on the training data set and tested on the testing data set so that an appropriate network structure can be determined. The final selected neural network model was then evaluated on the unseen validation data. The first 200 data points in Figure 3, with the corresponding model input data shown in Figures 4 and 5, were randomly partitioned into a training data set and a testing data set. The remaining part serves as unseen validation data.

Linear models were first attempted followed by single neural networks and then aggregated networks. When building these models, the data are first pre-processed. As can be seen from Figure 4, some of the process measurements contain outliers. The first step in data pre-processing is to remove outliers in the data. Then the data are scaled to zero mean and unit variance. Table 1 gives the sum of squared errors (SSE) of different models on the whole data. Note that the SSE values in Table 1 are for scaled
data. It can be seen from Table 1 that the linear models possess quite large errors. The estimation errors are reduced in the single neural network models and further reduced in the aggregated network models. The neural networks used here are single hidden layer feed forward networks and trained with Levenberg-Marquardt optimisation algorithm with cross validation based “early stopping”. Network weights were initialised as random numbers uniformly distributed in the range (-0.1, 0.1). Hidden layer neurons use the sigmoidal activation function while output layer neurons use the linear activation function. The numbers of hidden neurons were found through cross validation analysis. The best number of hidden neurons for the single network for estimating $MI_1$ is 5 while that for $MI_2$ is 4. The aggregated neural networks were trained using the sequential training algorithm.

Figures 10 to 12 show, respectively, predictions of $MI_1$ and prediction errors form the linear model, the single neural model, and the aggregated neural network model. It can be seen that the linear model possesses significant errors. For many samples, it gives negative estimates that are physically unrealistic. The single neural network model improves the estimation with no negative estimates. It can be seen from Figure 12 that the aggregated neural network gives even more accurate estimates than the single network.

Figure 13 shows the SSE of aggregated networks with different numbers of individual networks in estimating $MI_1$. The parameter $\lambda$ was selected as 0.1. Figure 13 indicates that, after combining 11 neural networks, the SSE on the training and testing data stopped decreasing. Hence, network training can be stopped after combining the first 11 networks. Figure 13 also shows that the SSE on all data after combining the first 11 networks is among the lowest level. This indicates the reliability of bootstrap aggregated neural networks in that their performance is quite consistent on training data and unseen data.

Model estimation performance for $MI_2$ from the linear model, the single network model, and the aggregated neural network model is given in Table 1. The linear model estimates possess significantly large errors. The single neural network estimates are significantly better than those from the linear model. The aggregated network further improves the estimation accuracy.

4. Conclusions

Inferential estimation of polymer melt index in an industrial plant using bootstrap aggregated neural networks is presented in this paper. A sequential training algorithm for building bootstrap aggregated neural network models is developed so that the individual networks are trained co-operatively. The training algorithm not only minimises the individual network prediction errors but also minimise the
correlation among the individual networks. Through this sequential training algorithm, effective combination of individual networks is achieved in that a relatively small number of neural networks need to be developed. For the purpose of comparison, linear models, single network models, and bootstrap aggregated network models were developed and compared. Due to the nonlinearities in the polymerisation process, the linear models possess significantly large estimation errors. The estimation errors can be reduced by using single neural network models and even further reduced by using bootstrap aggregated models. Application of the proposed training method to the development of soft sensors in an industrial polymerisation plant demonstrates its effectiveness.

Acknowledgements

The authors gratefully acknowledge the support from the UK EPSRC (GR/R10875 and GR/S90461).

References


List of Figure Captions:

Figure 1. A bootstrap aggregated neural network

Figure 2. A simplified schematic diagram of the propylene polymerisation process

Figure 3. Melt index data

Figure 4. Plots of all process variables

Figure 5. H\textsubscript{2} feed rate to reactor D201

Figure 6. H\textsubscript{2} concentrations in reactors 1 and 2

Figure 7. Cross-correlations: (a). between the hydrogen concentration in reactor 1 and MI in reactor 1; (b). between the hydrogen feed rate to reactor 1 and MI in reactor 1

Figure 8. Cross-correlations: (a). between the hydrogen concentration in reactor 1 and MI in reactor 4; (b). between the hydrogen concentration in reactor 2 and MI in reactor 4

Figure 9. Cross-correlations between the time-shifted process variables and MI: (a). $F(t-9)$ and $MI_1(t)$; (b). $H_1(t-7)$ and $MI_2(t)$; (c). $H_2(t-6)$ and $MI_3(t)$;

Figure 10. Predictions of MI in D201 from the linear model

Figure 11. Predictions of MI in D201 from the single neural network model

Figure 12. Predictions of MI in D201 from the aggregated neural network model

Figure 13. SSE of aggregated network with different numbers of individual networks

List of Table Caption:

Table 1. SSE of different models on the whole data
Table 1. SSE of different models on the whole data

<table>
<thead>
<tr>
<th>Models</th>
<th>MI₁</th>
<th>MI₂</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear</td>
<td>27.7870</td>
<td>30.5355</td>
</tr>
<tr>
<td>Single network</td>
<td>20.8751</td>
<td>21.6210</td>
</tr>
<tr>
<td>Aggregated network</td>
<td>16.7023</td>
<td>18.7750</td>
</tr>
</tbody>
</table>
**Figure 1.** A bootstrap aggregated neural network

**Figure 2.** A simplified schematic diagram of the propylene polymerisation process
Figure 3. Melt index data
Figure 4. Plots of all process variables
Figure 5. H$_2$ feed rate to reactor D201

Figure 6. H$_2$ concentrations in reactors 1 and 2
Figure 7. Cross-correlations: (a). between the hydrogen concentration in reactor 1 and MI in reactor 1; (b). between the hydrogen feed rate to reactor 1 and MI in reactor 1

Figure 8. Cross-correlations: (a). between the hydrogen concentration in reactor 1 and MI in reactor 4; (b). between the hydrogen concentration in reactor 2 and MI in reactor 4
Figure 9. Cross-correlations between the time-shifted process variables and MI: (a) $F(t-9)$ and $MI_1(t)$; (b) $H_1(t-7)$ and $MI_2(t)$; (c) $H_2(t-6)$ and $MI_2(t)$;

Figure 10. Predictions of MI in D201 from the linear model
Figure 11. Predictions of MI in D201 from the single neural network model

Figure 12. Predictions of MI in D201 from the aggregated neural network model
Figure 13. SSE of aggregated network with different numbers of individual networks