MIXTURE OF PARTIAL LEAST SQUARES REGRESSION EXPERTS APPLIED TO A POLYMERIZATION BATCH PROCESS

Francisco Alexandre A. Souza*, Rui A. M. Araújo*, Antônio Barbosa S. Júnior†, Ricardo Silva Thé Pontes†

*DEEC - Department of Electrical and Computer Engineering;
ISR-Institute of Systems and Robotics;
University of Coimbra, Pólo II;
Coimbra, Portugal

†DEE - Department of Electrical Engineering;
GPAR - Research Group for Automation and Robotics;
University Federal of Ceará;
Ceará, Brazil

Emails: fasouza@isr.uc.pt, rui@isr.uc.pt, barbosa@dee.ufc.br, ricthe@dee.ufc.br

Abstract— This paper proposes the mixture of partial least square regression models (Mix-PLS) for prediction polymerization batch process. The Mix-PLS is the solution of the mixture of experts framework using the partial least squares (PLS) algorithm together with the expectation-maximization (EM) algorithm. The proposed method, is less prone to overfitting with respect to the number of mixture models, when compared to the standard least squares solution of mixture of experts (Mix-LS). The Mix-PLS was successfully applied to predict the viscosity and acidity (two quality measures) in a polymerization batch process. The results were compared with four other regression algorithms (Mix-LS, PLS, MLP, and LS-SVR). The proposed method exhibits the best prediction performance.

Keywords— soft sensors, mixture of experts, multiple modes, expectation-maximization, partial least squares

1 Introduction

Today, soft sensors have many applications in industry (e.g. fault detection, process monitoring, prediction of critical variables, and control) (Fortuna et al., 2006; Kadlec et al., 2009). The major number of soft sensors applications consists on the prediction of critical or hard-to-measure[1] variables, where easy-to-measure variables (i.e. physical sensors) are used in a model to predict the hard-to-measure variable. The most popular data-driven models used in soft sensors applications are the multiple linear regression, with least squares (LS) or partial least squares (PLS) estimation methods, neural networks based models (NN), and support vector regression (SVR).

In almost all soft sensor applications, a single model is tuned using all available training samples, without distinguishing the operating modes of the process during the training phase. However, the existence of multiple operating modes in a process is an inherent characteristic of most industrial applications. Sometimes multiple operating modes result from external disturbances, as for example a change in feedstock or product grade or even changes such as the diurnal load variation of a power plant or the summer-winter operation of a refinery (Matzopoulos, 2010; Wang et al., 2012). In these situations, it would be beneficial for the prediction accuracy and reasonably, to consistently train a model for each operating mode of the process (Yu, 2012), or train a model for each set of correlated operating modes (Facco et al., 2009); And during online operation, when a new sample is made available, the model which is the most adequate for this new sample is identified and then used to make the prediction. The identification of which model will be used is a key issue in the development (Facco et al., 2009; Camacho and Picó, 2006; Lu and Gao, 2005), which can be done using expert knowledge (Facco et al., 2009) or using automatic tools, as finite mixture of Gaussian models (FMGM) (Yu, 2012).

In this work, it is proposed the use of a mixture of partial least squares (PLS) experts (Mix-PLS) for dealing with online prediction of critical variables in processes with multiple operating modes. The Mix-PLS will be derived from the framework of mixture of experts (ME) (Jacobs et al., 1991). The ME models input-output observations by assuming that they have been produced by a set of different random sources (the random sources can be thought as operating modes). Each random source in the ME framework is modeled by an expert, and during the online operation the decision about which experts should be used is modeled by a gating function. Figure 1 illustrates this approach.

The learning of parameters in ME can be done using the maximum likelihood method and the expectation and maximization (EM) algorithm (Dempster et al., 1977). By modeling the experts by a Gaussian distribution and the gates functions as a softmax function, the ME is then reduced to a mixture of linear regression experts (MLRE) (Jordan, 1994). The standard solution of MLRE requires the use of the least squares (LS) estimator to find the model and gate parameters, this solution will be refereed as Mix-LS. However, the

[1]The term hard-to-measure variable, employed here, refers to a variable which can not be measured by physical sensors, due the unavailability of sensor. Usually, this kind of variable is measured by laboratory analysis.

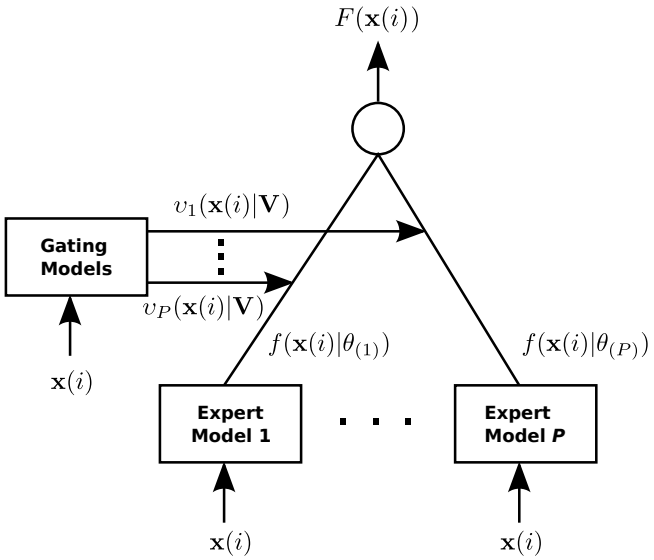


Figure 1: Mixture of linear regression models with P experts, where $\mathbf{x}(i)$ is an input sample, $v_p(\mathbf{x}(i), \mathbf{V})$ is the output of gating function for model p and $f(\mathbf{x}(i), \boldsymbol{\theta}_p)$ is the output of the linear model of expert p.

LS estimator cannot handle input collinearity, and the LS estimator makes the MLRE more prone to overfitting with respect to the number of experts used (Yuksel et al., 2012).

In this work the parameters of each expert and for each gating function are determined using the PLS algorithm. The solution of the parameters using the PLS algorithm overcomes the problem of collinearity of input data and also makes the Mix-PLS less prone to overfitting with respect to the number of mixture models.

In the experimental part, the Mix-PLS is then applied in one batch process and compared with the state of the art algorithms for regression that are used in state of the art soft sensors methods: a single PLS model, a single layer neural network (MLP) trained using the gradient descent training algorithm, and least squares support vector regression (LS-SVR) with Gaussian kernel (Suykens et al., 2002). The experimental results indicate that the recursive Mix-PLS outperforms the other methods. Moreover, the Mix-PLS has the advantage of being more interpretable than the non linear models with respect to the parameters.

The paper is organized as follows. Section 3 reviews the PLS algorithm and its parameters selection. The proposed Mix-PLS method is presented in Section 4. Section 5 presents experimental results. Finally, Section 6 gives concluding remarks.

2 Notation

The notation used here is defined as follows, $\mathbf{x}(i) = [x_1(i), \ldots, x_D(i)]^T$ and $y(i)$ are the vector of input variables and the output target at instant i, $\mathbf{X}$, with elements $X_{ij} = x_j(i)$, and $\mathbf{y}$, with elements $y_{i,1} = y(i)$ are the input matrix and output vector containing all the k examples. Moreover, $\mathcal{X} = \mathcal{X}_1 \times \ldots \times \mathcal{X}_D$, and $\mathcal{Y}$, denote the space of input variables values and the space of output values, respectively, where $\mathcal{X} \subset \mathbb{R}^D$ and $\mathcal{Y} \subset \mathbb{R}$. A subscript k will be used to denote the value of the corresponding variable after k samples.

3 Partial Least Squares

PLS regression is a method for finding the parameters $\boldsymbol{\theta} = [\theta_1, \ldots, \theta_D]^T$ of a linear model of the form $f(\mathbf{x}, \boldsymbol{\theta}) = \theta_0 + \sum_{j=1}^{D} \theta_j x_j$ from a given a set of input-output samples $\Phi = \{(\mathbf{x}(i), y(i)); i = 1, \ldots, k\}$. This model is composed by a linear combination of the inputs for regression. The objective of the design of the linear combination is to maximize the covariance between the input and output spaces. The PLS estimation method is attractive because it works well on high dimensional data, noisy data, and data with collinearity, which are common characteristics in most industrial applications.

More specifically, PLS projects the information of the data into a low dimensional space defined by a small number of orthogonal latent vectors $\mathbf{t}_m$ and $\mathbf{u}_m$, with $\mathbf{T} = (\mathbf{t}_1, \ldots, \mathbf{t}_M) \in \mathbb{R}^{k \times M}$ (with $M \leq D$ as the number of latent variables) and $\mathbf{U} = (\mathbf{u}_1, \ldots, \mathbf{u}_M) \in \mathbb{R}^{k \times M}$:

$$\mathbf{X} = \mathbf{T}\mathbf{P}^T + \mathbf{E} = \sum_{m=1}^{M} \mathbf{t}_m \mathbf{p}_m^T + \mathbf{E}, \tag{1}$$

$$\mathbf{y} = \mathbf{T}\mathbf{B}\mathbf{Q}^T + \mathbf{F} = \sum_{m=1}^{M} \mathbf{u}_m \mathbf{q}_m^T + \mathbf{F}, \tag{2}$$

where $\mathbf{P} = (\mathbf{p}_1, \ldots, \mathbf{p}_M) \in \mathbb{R}^{D \times M}$ and $\mathbf{Q} = (\mathbf{q}_1, \ldots, \mathbf{q}_M) \in \mathbb{R}^{1 \times M}$ are the loading matrices, $\mathbf{E}$ and $\mathbf{F}$ are the input and output data residuals, $\mathbf{B} = \text{diag}(b_1, \ldots, b_M)$ is a diagonal matrix with the regression weights b_m. Then, the estimated output $\hat{y}$, given an input sample $\mathbf{x}$, is given by:

$$\hat{y} = \mathbf{x}^T \boldsymbol{\theta}, \tag{3}$$

where $\boldsymbol{\theta} = \mathbf{P}^{\dagger}\mathbf{B}\mathbf{Q}^T$, and $\mathbf{P}^{\dagger} = (\mathbf{P}\mathbf{P}^T)^{-1}\mathbf{P}$ is the pseudo-inverse of $\mathbf{P}$. The values of b_m $(m = 1, \ldots, M)$, $\mathbf{T}$, $\mathbf{P}$, $\mathbf{U}$, $\mathbf{Q}$ from the above problem can be computed by using the classical Nonlinear Interative Partial Least Squares (NIPALS) method (Wold, 1975).

3.1 Selecting the Number of Latent Variables

Let $\mathcal{M}$ be such that $M \in \mathcal{M}$, for any possible/eligible number of latent variables, M. The major concern regarding the PLS algorithm is to select the number of latent variables M. Usually it is determined by a K-fold cross-validation procedure applied on the training set. In K-fold cross validation the training set is split randomly into K subsets or folds, then the PLS is trained using the samples from the $(K-1)$ folds and evaluated in the remaining fold using any performance metric, usually the residual sum of squares (RSS); e.g. lower values of RSS indicate better models. It is repeated for all folds K, and with different values for the number of latent factors. The selected number of latent factors M is the one that produced the lowest average cross-validation performance metric among these K realizations. However, the K-fold cross-validation procedure is very efficient as long

as k (the number of samples) is not too large, since it needs to run the PLS algorithm $K|\mathcal{M}|$ times. A fast way of selecting the number of latent variables is using information criterion methods, like the Akaike Information Criterion (AIC) (Akaike, 1974) or the Bayesian Information Criterion (BIC) (Schwarz, 1978), which measure the quality of a model in terms of its accuracy-complexity trade-off (ACT). Using information criterion methods, the PLS algorithm runs just $|\mathcal{M}|$ times.

4 Mixture of Partial Least Squares Regression Experts

In this section, the formulas for the learning of the Mix-PLS will be briefly discussed, due the limited number of pages. For the learning, the parameters of the Mix-PLS are tuned using a set of observations Φ.

4.1 Mixture of Experts

The ME approximates the true pdf $p(y(i)|\mathbf{x}(i))$ with the following superposition of individual pdfs:

$$p(y(i)|\mathbf{x}(i),\boldsymbol{\vartheta}) = \sum_{p=1}^{P} \upsilon_p(\mathbf{x}(i),\mathbf{V})\, p(y(i)|f_p(\mathbf{x}(i),\boldsymbol{\theta}_p),\boldsymbol{\Omega}), \quad (4)$$

where P is the number of experts, $\boldsymbol{\vartheta} = \{\mathbf{V}, \boldsymbol{\mathcal{E}}\}$, $\mathbf{V}$ and $\boldsymbol{\mathcal{E}} = \{\boldsymbol{\Theta}, \boldsymbol{\Omega}\}$ are defined as the sets of parameters of the gates and model experts, respectively, $\boldsymbol{\Theta} = \{\boldsymbol{\theta}_p | \, p = 1,\ldots,P\}$, $\upsilon_p(\mathbf{x}(i),\mathbf{V})$ is the gating function of expert p, and $p(y(i)|f_p(\mathbf{x}(i),\boldsymbol{\theta}_p),\boldsymbol{\Omega})$ is the pdf of expert model p, with mean $f_p(\mathbf{x}(i),\boldsymbol{\theta}_p)$ and additional pdf parameters $\boldsymbol{\Omega}$. From (4), prediction equation of ME is obtained as the following conditional mean of y:

$$F(\mathbf{x}(i)) = \int y\, p(y|\mathbf{x}(i),\boldsymbol{\vartheta}) \mathrm{d}y = \sum_{p=1}^{P} \upsilon_p(\mathbf{x}(i),\mathbf{V})\, f_p(\mathbf{x}(i),\boldsymbol{\theta}_p). \quad (5)$$

In the ME the log likelihood of (4), given a set of observations Φ is given by (Jacobs et al., 1991):

$$\begin{aligned} \ln p(\mathbf{y}|\mathbf{X},\boldsymbol{\vartheta}) &= \ln\left(\sum_{\mathbf{Z}} p(\mathbf{Z}|\mathbf{X},\mathbf{V})\, p(\mathbf{y}|\mathbf{X},\mathbf{Z},\boldsymbol{\mathcal{E}})\right) \\ &= \ln\left(\prod_{i=1}^{k} p(y(i)|\mathbf{x}(i),\boldsymbol{\vartheta})\right), \\ &= \ln\left(\prod_{i=1}^{k}\sum_{\mathbf{z}(i)} p(\mathbf{z}(i)|\mathbf{x}(i),\mathbf{V})\, p(y(i)|\mathbf{x}(i),\mathbf{z}(i),\boldsymbol{\mathcal{E}})\right), \end{aligned} \quad (6)$$

where $\mathbf{Z}$ denotes a set of hidden variables $\mathbf{Z} = \{z_p(i) | \, p = 1,\ldots,P, i = 1,\ldots,k\}$, and $\mathbf{z}(i) = [z_1(i),\ldots,z_P(i)]^T$ is the vector of hidden variables for a sample i, where $z_p(i) \in \{0,1\}$, and for each sample i, all variables $z_p(i)$ are zero, except for a single value of $z_p(i) = 1$, for some p. The hidden variable $z_p(i)$ indicates which expert p was responsible for generating the data point i.

The maximization of (6) is not straightforward (Bishop, 2006; Jacobs et al., 1991). In order to maximize (6) the Expectation-Maximization (EM) algorithm will be employed. The EM algorithm is a general method for finding the maximum-likelihood estimate of the parameters of an underlying distribution from a given data set when the data has hidden variables (Dempster et al., 1977; Bishop, 2006). The learning of the mixture of experts by the EM algorithm is summarized in Table 1. During the Expectation step (E step) of the EM, the current parameter values $\boldsymbol{\vartheta}^{(\mathrm{old})}$ are used to estimate the posterior distribution of hidden variables:

$$p(\mathbf{Z}|\mathbf{y},\mathbf{X},\boldsymbol{\vartheta}) = \prod_{i=1}^{k}\prod_{p=1}^{P}\left(\frac{p\left(y(i)|z_p(i),\mathbf{x}(i),\boldsymbol{\mathcal{E}}\right)\, p\left(z_p(i)|\mathbf{x}(i),\mathbf{V}\right)}{\sum_{p=1}^{P}\left[p\left(z_p(i)|\mathbf{x}(i),\mathbf{V}\right)\, p\left(y(i)|z_p(i),\mathbf{x}(i),\boldsymbol{\mathcal{E}}\right)\right]}\right)^{z_p(i)}. \quad (7)$$

Then, in the Maximization step (M step), this posterior distribution is used to find the new parameters values $\boldsymbol{\vartheta}^{(\mathrm{new})}$, which maximize the expectation of the complete-data (output and hidden variables) log likelihood:

$$\begin{aligned} Q(\boldsymbol{\vartheta},\boldsymbol{\vartheta}^{(\mathrm{old})}) &= \mathbb{E}_{\mathbf{Z}}[\ln p(\mathbf{y},\mathbf{Z}|\mathbf{X},\boldsymbol{\vartheta})] \\ &= \sum_{\mathbf{Z}} \ln p(\mathbf{y},\mathbf{Z}|\mathbf{X},\boldsymbol{\vartheta})\, p(\mathbf{Z}|\mathbf{y},\mathbf{X},\boldsymbol{\vartheta}^{(\mathrm{old})}). \end{aligned} \quad (8)$$

where the value of $p(\mathbf{y},\mathbf{Z}|\mathbf{X},\boldsymbol{\vartheta})$, necessary to compute $Q(\boldsymbol{\vartheta},\boldsymbol{\vartheta}^{(\mathrm{old})})$ is:

$$\begin{aligned} p(\mathbf{y},\mathbf{Z}|\mathbf{X},\boldsymbol{\vartheta}) &= p(\mathbf{y}|\mathbf{X},\mathbf{Z},\boldsymbol{\vartheta})\, p(\mathbf{Z}|\mathbf{X},\boldsymbol{\vartheta}), \\ &= \prod_{i=1}^{k}\prod_{p=1}^{P}\left[p\left(z_p(i)|\mathbf{x}(i),\mathbf{V}\right)\, p\left(y(i)|z_p(i),\mathbf{x}(i),\boldsymbol{\mathcal{E}}\right)\right]^{z_p(i)}. \end{aligned} \quad (9)$$

The expectation of the complete-data log likelihod (8) can be computed using (7) and (9), which gives:

$$\begin{aligned} Q(\boldsymbol{\vartheta},\boldsymbol{\vartheta}^{(\mathrm{old})}) &= \sum_{\mathbf{Z}} \ln p(\mathbf{y},\mathbf{Z}|\mathbf{X},\boldsymbol{\vartheta})\, p(\mathbf{Z}|\mathbf{y},\mathbf{X},\boldsymbol{\vartheta}), \\ &= Q_g(\mathbf{V},\boldsymbol{\vartheta}^{(\mathrm{old})}) + Q_e(\boldsymbol{\mathcal{E}},\boldsymbol{\vartheta}^{(\mathrm{old})}), \end{aligned} \quad (10)$$

where $Q_g(\mathbf{V},\boldsymbol{\vartheta}^{(\mathrm{old})})$ is the contribution of gate for the expectation of complete-data log likelihood, (Q_g is a simplified notation) :

$$Q_g = \sum_{i=1}^{k}\sum_{p=1}^{P} \gamma_p^{(\mathrm{old})}(i) \ln p\left(z_p(i) = 1|\mathbf{x}(i),\mathbf{V}\right), \quad (11)$$

and $Q_e(\boldsymbol{\mathcal{E}},\boldsymbol{\vartheta}^{(\mathrm{old})})$ is the contribution of expert for the expectation of complete-data log likelihood, (Q_e is a simplified notation) :

$$Q_e = \sum_{i=1}^{k}\sum_{p=1}^{P} \gamma_p^{(\mathrm{old})}(i) \ln p\left(y(i)|z_p(i) = 1,\mathbf{x}(i),\boldsymbol{\mathcal{E}}\right), \quad (12)$$

where $\gamma_p^{(\mathrm{old})}(i)$, defined as the responsibility of model p, is the expectation of $z_p(i)$ with respect to its distribution (7), and it accounts for the probability of model p generating the data sample i:

$$\gamma_p^{(\mathrm{old})}(i) = \frac{p\left(z_p(i) = 1|\mathbf{x}(i),\mathbf{V}^{(\mathrm{old})}\right) p\left(y(i)|z_p(i) = 1,\mathbf{x}(i),\boldsymbol{\mathcal{E}}^{(\mathrm{old})}\right)}{\sum_{l=1}^{P}\left[p\left(z_l(i) = 1|\mathbf{x}(i),\mathbf{V}^{(\mathrm{old})}\right) p\left(y(i)|z_l(i) = 1,\mathbf{x}(i),\boldsymbol{\mathcal{E}}^{(\mathrm{old})}\right)\right]}. \quad (13)$$

Table 1: EM Algorithm

1. Initialize $\boldsymbol{\vartheta}$ to be equal to some initial $\boldsymbol{\vartheta}^{(\text{old})}$;
2. Repeat 3) to 5) until the EM algorithm converges*;
3. **E** step:
   a) Estimate the distribution $p(\mathbf{Z}|\mathbf{y},\mathbf{X},\boldsymbol{\vartheta}^{(\text{old})})$ using (7);
4. **M** step:
   a) Find the new parameters values $\boldsymbol{\vartheta}^{(\text{new})}$, which maximize the expectation of the complete-data log likelihood $Q(\boldsymbol{\vartheta},\boldsymbol{\vartheta}^{(\text{old})})$.
      i. $\boldsymbol{\vartheta}^{(\text{new})} = \arg\max_{\boldsymbol{\vartheta}} Q(\boldsymbol{\vartheta},\boldsymbol{\vartheta}^{(\text{old})}) = \arg\max_{\boldsymbol{\vartheta}} \left(\sum_{\mathbf{Z}} \ln p(\mathbf{y},\mathbf{Z}|\mathbf{X},\boldsymbol{\vartheta}) p(\mathbf{Z}|\mathbf{y},\mathbf{X},\boldsymbol{\vartheta}^{(\text{old})})\right)$ (Equation (14));
5. Set $\boldsymbol{\vartheta}^{(\text{old})} \leftarrow \boldsymbol{\vartheta}^{(\text{new})}$;
6. Return $\boldsymbol{\vartheta}^{(\text{new})}$.

*The convergence of the EM algorithm can be verified by analyzing the convergence of the expectation $Q(\boldsymbol{\vartheta},\boldsymbol{\vartheta}^{\text{old}})$. It is also possible to set pre-specified maximum number of iterations.

In (10), Q_g and Q_e are the contributions of gate and expert parameters for the expectation of complete-data log likelihood. Then, the M step of the EM algorithm can be performed, by separately maximizing the gate and expert contributions, as follows:

$$\begin{aligned}\boldsymbol{\vartheta}^{(\text{new})} &= \arg\max_{\boldsymbol{\vartheta}} Q(\boldsymbol{\vartheta},\boldsymbol{\vartheta}^{(\text{old})}),\\ &= \left\{\arg\max_{\mathbf{V}} Q_g(\mathbf{V},\boldsymbol{\vartheta}^{(\text{old})}), \arg\max_{\boldsymbol{\mathcal{E}}} Q_e(\boldsymbol{\mathcal{E}},\boldsymbol{\vartheta}^{(\text{old})})\right\}. \quad (14)\end{aligned}$$

Thus, the determination of the parameters for the gates $\mathbf{V}$ and the experts $\boldsymbol{\mathcal{E}}$ is independently performed by the maximizations in (14). In the Mix-PLS, such maximizations are done using the PLS algorithm, as derived in Subsections 4.2 and 4.3 below.

4.2 Modeling the Experts With the PLS Algorithm

In this paper, it is assumed that each pdf $p\left(y(i)|z_p(i)=1,\mathbf{x}(i),\boldsymbol{\mathcal{E}}\right)$ in $Q_e(\boldsymbol{\mathcal{E}},\boldsymbol{\vartheta}^{(\text{old})})$ (10) is described by a Gaussian distribution $\mathcal{N}(y(i)|f_p(\mathbf{x}(i),\boldsymbol{\theta}_p),\omega_p)$, where $f_p(\mathbf{x}(i),\boldsymbol{\theta}_p)$, and ω_p are the mean and variance of the model of expert p, respectively. The mean is modeled by a linear model $f_p(\mathbf{x}(i),\boldsymbol{\theta}_p) = \mathbf{x}(i)^T\boldsymbol{\theta}_p$. Specifically, the experts parameters $\boldsymbol{\mathcal{E}} = \{\boldsymbol{\Theta},\boldsymbol{\Omega}\}$, include the parameters of $\boldsymbol{\Theta} = \{\boldsymbol{\theta}_p|\, p=1,\ldots,P\}$, and $\boldsymbol{\Omega} = \{\omega_p|\, p=1,\ldots,P\}$. Thus, the contribution $Q_e(\boldsymbol{\mathcal{E}},\boldsymbol{\vartheta}^{(\text{old})})$ of all experts to the expectation of complete data log likelihood (10) can be rewritten as:

$$Q_e = \sum_{p=1}^{P}\sum_{i=1}^{k} \gamma_p^{(\text{old})}(i) \ln \mathcal{N}(y(i)|f_p(\mathbf{x}(i),\boldsymbol{\theta}_p),\omega_p), \quad (15)$$

and from (13) the responsibility $\gamma_p^{(\text{old})}(i)$ is equal to:

$$\gamma_p^{(\text{old})}(i) = \frac{\upsilon_p^{(\text{old})}(i)\mathcal{N}(y(i)|f_p(\mathbf{x}(i),\boldsymbol{\theta}_p^{(\text{old})}),\omega_p^{(\text{old})})}{\sum_{l=1}^{P}\upsilon_l^{(\text{old})}(i)\mathcal{N}(y(i)|f_l(\mathbf{x}(i),\boldsymbol{\theta}_l^{(\text{old})}),\omega_l^{(\text{old})})}, \quad (16)$$

where $\upsilon_p^{(\text{old})}(i) = p\left(z_p(i)=1|\mathbf{x}(i),\mathbf{V}^{(\text{old})}\right)$ is the probability of model p generating sample i, which will be determined in Section 4.3.

Then, $Q_e(\boldsymbol{\mathcal{E}},\boldsymbol{\vartheta}^{(\text{old})})$ is maximized with respect to $\boldsymbol{\mathcal{E}}$ by solving equations $\frac{\partial Q_e(\boldsymbol{\mathcal{E}},\boldsymbol{\vartheta}^{(\text{old})})}{\partial\boldsymbol{\theta}_p} = 0$, and $\frac{\partial Q_e(\boldsymbol{\mathcal{E}},\boldsymbol{\vartheta}^{(\text{old})})}{\partial\omega_p} = 0$, which gives the following solution:

$$\boldsymbol{\theta}_p^{(\text{new})} = \left(\mathbf{X}^T\boldsymbol{\Gamma}_p\mathbf{X}\right)^{-1}\mathbf{X}^T\boldsymbol{\Gamma}_p\mathbf{y}, \quad (17)$$

$$\omega_p^{(\text{new})} = \frac{\left|\left|\mathbf{y}_{(\boldsymbol{\Gamma},p)} - \mathbf{X}_{(\boldsymbol{\Gamma},p)}\boldsymbol{\theta}_p^{(\text{new})}\right|\right|^2}{\text{Tr}(\boldsymbol{\Gamma}_p)}, \quad (18)$$

where $\boldsymbol{\Gamma}_p = \text{diag}(\gamma_p^{(\text{old})}(1),\gamma_p^{(\text{old})}(2),\ldots,\gamma_p^{(\text{old})}(k))$ is a diagonal matrix, and $\mathbf{y}_{(\boldsymbol{\Gamma},p)}$ and $\mathbf{X}_{(\boldsymbol{\Gamma},p)}$ are defined in (19)-(20). As can be noticed, the maximization of Q_e (15) is equivalent to a weighted least squares problem, where the responsibility $\gamma_p^{(\text{old})}(i)$ is the importance of each sample.

In this work, the parameters of each model $\boldsymbol{\theta}_p^{(\text{new})}$ (17) will be solved using the PLS algorithm. In the PLS algorithm, from (1)-(2), the inputs $\mathbf{X}$ and output $\mathbf{y}$ are traditionally represented through their approximation with M latent and loading variables representation, i.e. $\mathbf{X} \approx \mathbf{TP}^T$ and $\mathbf{y} \approx \mathbf{TBQ}^T$. However, solving (17) after replacing these approximations is not straightforward. A simpler approach is to multiply both $\mathbf{X}$ and $\mathbf{y}$ by $\sqrt{\boldsymbol{\Gamma}_p}$, so that the weighted representation of $\mathbf{X}$ and $\mathbf{y}$ becomes equal to:

$$\mathbf{X}_{(\boldsymbol{\Gamma},p)} = \sqrt{\boldsymbol{\Gamma}_p}\mathbf{X} \approx \mathbf{T}_{(\boldsymbol{\Gamma},p)}\mathbf{P}_{(\boldsymbol{\Gamma},p)}^T, \quad (19)$$

$$\mathbf{y}_{(\boldsymbol{\Gamma},p)} = \sqrt{\boldsymbol{\Gamma}_p}\mathbf{y} \approx \mathbf{T}_{(\boldsymbol{\Gamma},p)}\mathbf{B}_{(\boldsymbol{\Gamma},p)}\mathbf{Q}_{(\boldsymbol{\Gamma},p)}^T, \quad (20)$$

where $\mathbf{X}_{(\boldsymbol{\Gamma},p)}$ and $\mathbf{y}_{(\boldsymbol{\Gamma},p)}$ are the weighted inputs and output matrices of model p with weight matrix $\boldsymbol{\Gamma}_p$. $\mathbf{T}_{(\boldsymbol{\Gamma},p)}$ and $\mathbf{P}_{(\boldsymbol{\Gamma},p)}$ are the PLS latent and loading matrices of the weighted input $\mathbf{X}_{(\boldsymbol{\Gamma},p)}$, and $\mathbf{B}_{(\boldsymbol{\Gamma},p)}$ and $\mathbf{Q}_{(\boldsymbol{\Gamma},p)}^T$ are the PLS latent and loading matrices of the weighted output $\mathbf{y}_{(\boldsymbol{\Gamma},p)}$. It is assumed that the weighted input and output decomposition for expert p through the PLS algorithm is made with Me_p latent variables.

Then, by replacing (19) and (20) into (17), the parameters of model p can be written as:

$$\boldsymbol{\theta}_p^{(\text{new})} = \left(\mathbf{P}_{(\boldsymbol{\Gamma},p)}\mathbf{P}_{(\boldsymbol{\Gamma},p)}^T\right)^{-1}\mathbf{P}_{(\boldsymbol{\Gamma},p)}\mathbf{B}_{(\boldsymbol{\Gamma},p)}\mathbf{Q}_{(\boldsymbol{\Gamma},p)}^T. \quad (21)$$

4.3 Modeling the Gates with the PLS Algorithm

Let the gate parameters be $\mathbf{V} = \{\mathbf{v}_p | p = 2, \ldots, P\}$, where $\mathbf{v}_p$ is the regression coefficient of gate p. In this work, the gate of each expert in (4) is modeled using the softmax function as follows:

$$\upsilon_p(i) = p\left(z_p(i) = 1 | \mathbf{x}(i), \mathbf{V}\right) = \tag{22}$$

$$= \begin{cases} \frac{1}{1+\sum_{l=2}^{P} \exp\left(\mathbf{x}^T(i)\mathbf{v}_l\right)}, & p = 1, \\ \frac{\exp\left(\mathbf{x}^T(i)\mathbf{v}_p\right)}{1+\sum_{l=2}^{P} \exp\left(\mathbf{x}^T(i)\mathbf{v}_l\right)}, & p = 2, \ldots, P, \end{cases} \tag{23}$$

where $\upsilon_p(i)$ is used as a simplified notation for $\upsilon_p(\mathbf{x}(i), \mathbf{V})$.

It can be seen that (23) keeps valid the constraint $\sum_{p=1}^{P} p\left(z_p(i) = 1 | \mathbf{x}(i), \mathbf{V}\right) = 1$. Then, the gate contribution $Q_g(\mathbf{V}, \boldsymbol{\vartheta}^{(\text{old})})$ to $Q(\boldsymbol{\vartheta}, \boldsymbol{\vartheta}^{(\text{old})})$ (see (10), (14)) can be rewritten as:

$$Q_g(\mathbf{V}, \boldsymbol{\vartheta}^{(\text{old})}) = \sum_{i=1}^{k} \left[\sum_{p=2}^{P} \gamma_p^{(\text{old})}(i)\, \mathbf{x}^T(i)\mathbf{v}_p - \sum_{p=1}^{P} \gamma_p^{(\text{old})}(i) \ln\left(1 + \sum_{l=2}^{P} \exp(\mathbf{x}^T(i)\mathbf{v}_l)\right) \right]. \tag{24}$$

In order to find the parameters $\mathbf{V}$ to update the gating parameters in the M step, it is necessary to maximize equation (24). The maximization of $Q_g(\mathbf{V}, \boldsymbol{\vartheta}^{(\text{old})})$ with respect to each gate parameter $\mathbf{v}_p$ will be obtained by the iterative reweighted least squares (IRLS) method (Jordan, 1994; Nabney, 1999) as follows:

$$\mathbf{v}_p^{(\text{new})} = \mathbf{v}_p^{(\text{old})} + \left[-\frac{\partial^2 Q_g(\mathbf{V}, \boldsymbol{\vartheta}^{(\text{old})})}{\partial \mathbf{v}_p \mathbf{v}_p^T} \right]^{-1} \left[\frac{\partial Q_g(\mathbf{V}, \boldsymbol{\vartheta}^{(\text{old})})}{\partial \mathbf{v}_p} \right]. \tag{25}$$

From (24), the derivatives in (25) can be obtained:

$$\left[-\frac{\partial^2 Q_g(\mathbf{V}, \boldsymbol{\vartheta}^{(\text{old})})}{\partial \mathbf{v}_p \mathbf{v}_p^T} \right]^{-1} = (\mathbf{X}^T \mathbf{R}_p \mathbf{X})^{-1}, \tag{26}$$

$$\left[\frac{\partial Q_g(\mathbf{V}, \boldsymbol{\vartheta}^{(\text{old})})}{\partial \mathbf{v}_p} \right] = \mathbf{X}^T \mathbf{u}_p, \tag{27}$$

where $\mathbf{R}_p = \text{diag}(\upsilon_p(1)(1 - \upsilon_p(1)), \upsilon_p(2)(1 - \upsilon_p(2)), \ldots, \upsilon_p(k)(1 - \upsilon_p(k)))$ is a diagonal matrix and $\mathbf{u}_p = [\gamma_p^{(\text{old})}(1) - \upsilon_p(1), \gamma_p^{(\text{old})}(2) - \upsilon_p(2), \ldots, \gamma_p^{(\text{old})}(k) - \upsilon_p(k)]^T$. After some manipulations, equation (25) can be transformed to:

$$\mathbf{v}_p^{(\text{new})} = \left(\mathbf{X}^T \mathbf{R}_p \mathbf{X}\right)^{-1} \mathbf{X}^T \mathbf{R}_p \mathbf{z}_p, \tag{28}$$

where $\mathbf{z}_p = \mathbf{X}\mathbf{v}_p^{(\text{old})} - \mathbf{R}_p^{-1}\mathbf{u}_p$. Now the parameters $\mathbf{v}_p$ for $p > 1$ can be solved using the PLS algorithm, similarly to the method that was used to determine the expert parameters (Section 4.2). Using (1)-(2), the weighted input and output values are written in terms of their latent and loading variables as follows:

$$\mathbf{X}_{(\mathbf{R},p)} = \sqrt{\mathbf{R}_p}\,\mathbf{X} \approx \mathbf{T}_{(\mathbf{R},p)} \mathbf{P}_{(\mathbf{R},p)}^T, \tag{29}$$

$$\mathbf{z}_{(\mathbf{R},p)} = \sqrt{\mathbf{R}_p}\,\mathbf{z}_p \approx \mathbf{T}_{(\mathbf{R},p)} \mathbf{B}_{(\mathbf{R},p)} \mathbf{Q}_{(\mathbf{R},p)}^T, \tag{30}$$

where $\mathbf{X}_{(\mathbf{R},p)}$ and $\mathbf{y}_{(\mathbf{R},p)}$ are the weighted input matrix and weighted output vector of model p with weight matrix $\mathbf{R}_p$, and $\mathbf{T}_{(\mathbf{R},p)}$ and $\mathbf{P}_{(\mathbf{R},p)}$ are the latent and loading matrices of weighted input $\mathbf{X}_{(\mathbf{R},p)}$ and similarly, $\mathbf{B}_{(\mathbf{R},p)}$ and $\mathbf{Q}_{(\mathbf{R},p)}^T$ are the latent and loading matrices of weighted output $\mathbf{z}_{(\mathbf{R},p)} = [z_{(\mathbf{R},p)}(1), \ldots, z_{(\mathbf{R},p)}(k)]^T$. It is assumed that the weighted input and output decomposition through the PLS algorithm is made with Mg_p latent variables.

Then, from (28)-(30) the parameters vector of each gate p is updated using the PLS algorithm as follows:

$$\mathbf{v}_p^{(\text{new})} = \left(\mathbf{P}_{(\mathbf{R},p)} \mathbf{P}_{(\mathbf{R},p)}^T\right)^{-1} \mathbf{P}_{(\mathbf{R},p)} \mathbf{B}_{(\mathbf{R},p)} \mathbf{Q}_{(\mathbf{R},p)}^T. \tag{31}$$

5 Experimental Results

This section presents experimental results of the Mix-PLS applied in one batch polymerization process, presented in (Facco et al., 2009). As the objective of this work is to evaluate the proposed method, and not discuss the process itself. Only a short description of polymerization process is given as follows. The objective in the polymerization data set is the estimation of the quality of a resin produced in an industrial batch polymerization process (Facco et al., 2009). The resin quality is determined by the values of two chemical properties: the resin acidity number (N_A) and the resin viscosity (μ). The dataset is composed of 24 input variables and 521 samples for train and 133 for test.

In the experiments, the Mix-PLS, Mix-LS and PLS models will be tuned by using as input of the model the original variables (i.e. without any transformation on them) and the squared values of these variables; the objective while using the squared values of input variables is to introduce some nonlinearity into the linear models (Mix-PLS, Mix-LS and PLS). In the experiments the proposed Mix-PLS method will be compared with the Mix-LS, a single PLS model, a MLP trained using the gradient descent (MLP) training algorithm, and a LS-SVR with Gaussian kernel (Suykens et al., 2002, Chapter 3). From the results, it can be seen that Mix-PLS attains better results when compared with Mix-LS, PLS and to the MLP and LS-SVR non-linear models. Moreover, the Mix-PLS has the advantage of having more interpretability with respect to its parameters when compared with non linear models MLP and LS-SVR.

Table 2: NRMSE results on the test set.

Data set name	Mix-PLS	Mix-LS	PLS	MLP	LS-SVR
Poly.: (Viscosity) (Facco et al., 2009)	**8.07**	23.43	24.23	9.95	12.38
Poly.: (Acidity) (Facco et al., 2009)	**3.62**	5.54	4.25	3.93	5.94

Table 3: Parameters selected for each model in each experiment.

Data set	Mix-PLS	Mix-LS	PLS	MLP	LS-SVR
Viscosity	$P = 2$	$P = 2$	$M = 10$	$N = 3$	$\gamma_{\text{LS-SVR}} = 50$ $\sigma_{\text{LS-SVR}} = 10$
Acidity	$P = 2$	$P = 2$	$M = 17$	$N = 3$	$\gamma_{\text{LS-SVR}} = 50$ $\sigma_{\text{LS-SVR}} = 25$

In all data sets the normalized root mean square error (NRMSE) was used as a performance measure to compare the results of the methods:

$$\text{NRMSE} = \frac{\sqrt{\sum_{i=1}^{k} \left(y(i) - \hat{y}(i)\right)^2}}{\max\left(y\right) - \min\left(y\right)}, \tag{32}$$

where $y(i)$, and $\hat{y}(i)$ are the observed and predicted targets, respectively, and $\max(y)$, and $\min(y)$ are the maximum and minimum values of the observed target. NRMSE is often expressed in percentage. The closer the NRMSE is to 0 the better is the quality of prediction.

5.0.1 Polymerization Data-Set

This data set was studied in (Facco et al., 2009), and the objective is to estimate the viscosity and acidity of a resin produced in an industrial batch polymerization process. The number of hidden nodes N of the MLP and the regularization parameter $\gamma_{\text{LS-SVR}}$ and the Gaussian kernel parameter $\sigma_{\text{LS-SVR}}$ of the LS-SVR were determined using a 10-fold cross validation. For the PLS model the number of latent variables M, was determined using the BIC criterion. For the Mix-LS, and Mix-PLS the numbers of experts P were obtained from the method proposed in (Ng et al., 2006). Additionally, for the Mix-PLS the set that contains the number of latent variables for each expert and gates was determined using the BIC criterion. Table 3 shows the parameters obtained for each model and for each data set in the experiments.

According with Table 2, for predicting the viscosity, the Mix-PLS reached the best results among all the models in terms of NRMSE. Inspecting the results from the gates activation on the train and test set which are presented in Figure 2, it is possible to note that the prediction of the first expert is predominant at the beginning of each batch, and the prediction of the two models are combined, usually at the end of each batch. The Mix-PLS suggests, that for viscosity prediction, just two models are necessary and that their prediction should be combined at the end of each batch.

For predicting the acidity, the Mix-PLS also reached the best results in terms of NRMSE, as indicated in Table 2. The Mix-PLS and Mix-LS used 2 experts to predict the acidity. The plots of gates and prediction on the train and test sets were removed due the limited number of pages.

As can be seen the Mix-PLS was successfully applied on the Polymerization data set, delivering satisfactory predition results. Moreover, Mix-PLS has shown better results when compared with the nonlinear models.

6 Conclusion

This paper proposed the use of Mix-PLS to be used as a soft sensor in a real batch polymerization process. In the proposed Mix-PLS method, the solution of the mixture of linear regression models is done using the partial least squares regression model. The formulas for learning were derived based on the EM algorithm. Furthermore, in this work the proposed method has been evaluated and compared with the current state of art methods, encompassing the prediction of two variables. The results obtained with Mix-PLS were superior when compared with a Mix-LS, a single PLS, a MLP, and LS-SVR models. Differently of the non linear models, the Mix-PLS gives more interpretability to the prediction.

Future directions of this work are to research on the implementation of the method in an online manner, further increasing the applicability.

Acknowledgments

Francisco Alexandre A. Souza has been supported by Fundação para a Ciência e a Tecnologia (FCT) under grant SFRH/BD/63454/2009.

The authors acknowledge the support of FCT project PEst-C/EEI/UI0048/2013.

References

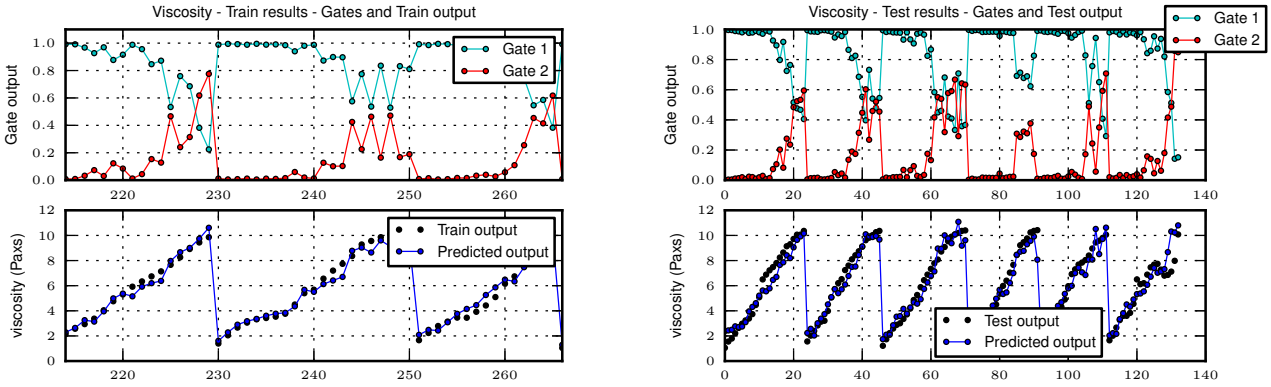
Akaike, H. (1974). A new look at the statistical model identification, *IEEE Transactions on Automatic Control* **19**(6): 716–723.

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*, 1 edn, Springer.

Camacho, J. and Picó, J. (2006). Online monitoring of batch processes using multi-phase principal component analysis, *Journal of Process Control* **16**(10): 1021–1035.

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, Series B* **39**(1): 1–38.

Facco, P., Doplicher, F., Bezzo, F. and Barolo, M. (2009). Moving average pls soft sensor for online



(a) **(b)**

Figure 2: Plots of viscosity prediction on Polymerization data set. (a) Train results, gates and prediction. (b) Test results, gates and prediction.

product quality estimation in an industrial batch polymerization process, *Journal of Process Control* **19**(3): 520–529.

Fortuna, L., Graziani, S., Rizzo, A. and Xibilia, M. G. (2006). *Soft Sensors for Monitoring and Control of Industrial Processes*, Advances in Industrial Control, 1 edn, Springer.

Jacobs, R. A., Jordan, M. I., Nowlan, S. J. and Hinton, G. E. (1991). Adaptive mixtures of local experts, *Neural Computation* **3**(1): 79–87.

Jordan, M. I. (1994). Hierarchical mixtures of experts and the em algorithm, *Neural Computation* **6**(2): 181–214.

Kadlec, P., Gabrys, B. and Strandt, S. (2009). Data-driven soft sensors in the process industry, *Computers & Chemical Engineering* **33**(4): 795–814.

Lu, N. and Gao, F. (2005). Stage-based process analysis and quality prediction for batch processes, *Industrial & Engineering Chemistry Research* **44**(10): 3547–3555.

Matzopoulos, M. (2010). Dynamic process modeling: Combining models and experimental data to solve industrial problems, *in* M. C. Georgiadis, J. R. Banga and E. N. Pistikopoulos (eds), *Process Systems Engineering*, Wiley-VCH Verlag GmbH & Co. KGaA, pp. 1–33.

Nabney, I. T. (1999). Efficient training of rbf networks for classification, *Proc. Ninth International Conference on Artificial Neural Networks, 1999 (ICANN 99)*, Vol. 1, Edinburgh, Scotland, pp. 210–215.

Ng, S.-K., McLachlan, G. J. and Lee, A. H. (2006). An incremental em-based learning approach for on-line prediction of hospital resource utilization, *Artificial Intelligence in Medicine* **36**(3): 257–267.

Schwarz, G. (1978). Estimating the dimension of a model, *Annals of Statistics* **6**(2): 461–464.

Suykens, J. A. K., Gestel, T. V., Brabanter, J. D., Moor, B. D. and Vandewalle, J. (2002). *Least Squares Support Vector Machiness*, World Scientific.

Wang, F., Tan, S., Peng, J. and Chang, Y. (2012). Process monitoring based on mode identification for multi-mode process with transitions, *Chemometrics and Intelligent Laboratory Systems* **110**(1): 144–155.

Wold, H. (1975). Path models with latent variables: The nipals approach, *in* H. M. B. et al. (ed.), *Quantitative Sociology: International Perspectives on Mathematical and Statistical Model Building*, Academic Press, pp. 307–357.

Yu, J. (2012). A nonlinear kernel gaussian mixture model based inferential monitoring approach for fault detection and diagnosis of chemical processes, *Chemical Engineering Science* **68**(1): 506–519.

Yuksel, S. E., Wilson, J. N. and Gader, P. D. (2012). Twenty years of mixture of experts, *IEEE Transactions on Neural Networks and Learning Systems* **23**(8): 1177–1193.