Mixture of Elastic Net Experts and its Application to a Polymerization Batch Process

Francisco Souza∗, and Rui Araújo∗,

∗Institute of Systems and Robotics (ISR-UC), and Department of Electrical and Computer Engineering (DEEC-UC), University of Coimbra, Pólo II, PT-3030-290 Coimbra, Portugal Email: {fasouza,rui}@isr.uc.pt

Abstract—Soft sensors are widely applied in many industrial systems for predicting quality variables in real-time; examples of soft sensors applications can be found in pulp and paper mills, cement kilns, polymerization processes and so on. However, there are several peculiarities that make soft sensors development difficult in some processes. One of these characteristics is the multiple operating mode scenarios, which is characterized by different operating regions of process, e.g., summer-winter operation of a refinery, batch processes, etc. In this paper, a regularized version of mixture of expert models, based on elastic net regularization, is employed for predicting the acidity value, a quality indicator variable of process, in a batch polymerization process. The proposed method allowed the application of 27 experts to compose the final model and attaining better prediction performance when compared to other state-of-the-art algorithms. The proposed method is compared to the predictive modeling state-of-the-art methods, such as support vector machines (SVM), decision trees (DT), and the elastic net model (EN).

1. INTRODUCTION

Soft sensors are inferential models that make use of physical monitored variables (easy-to-measure variables), to predict, in real-time, variables which are related to process quality control (hard-to-measure variables) [1]–[3]. This is motivated by the lack of physical sensors or the expensive value to monitor these variables. Soft sensors are an interesting alternative to acquire the hard-to-measure variables in real-time, then allowing the real-time monitoring of these variables, or even, in a more advanced way, the application of process control strategies on these variables [4]. During soft sensor design, and depending on the industrial plant, many different challenges and needs can appear, e.g., the application of feature selection strategies [5], [6], semi-supervised learning scenarios [7]–[9], multiple operating scenarios or multiphase processes [10]–[12], learning in batch production, etc. Multiple operating modes are characterized by different operating zones, where in each zone there is a different relationship between easy-to-measure and hard-to-measure variables, e.g., summer-winter operation of a refinery. Similarly, in batch production, there is a fixed recipe which is followed to achieve a product quality with determined characteristics; This can also be a characteristic of a multiple operating mode processes, where each batch can be seen as an operating mode, since processes settings change, due to different raw materials properties, gradual changes in process equipment, and changes in environmental variable [13]–[15]. This paper will focus on learning with multiple operating scenarios into batch learning. This setting is similar to the problems handled by mixture of linear expert (MoLE) models, where the output, $y$, to be predicted comes from an heterogeneous population, i.e., $y$ can be decomposed into subpopulations defined on possible overlapping regions of input space $x$ [16].

MoLE models are commonly used for non-linear regression, where, usually, the error of each expert is modeled by a Gaussian distribution, and the gates are modeled by softmax functions. Learning MoLE models in settings where $d \gg n$, where $d$ is the number of features and $n$ is the number of samples, or in the presence of a large number of experts, provides unreliable results. In the case where $d \gg n$, these problems arise from the presence of collinearity in data. In the case of using a large number of experts, the effective number of samples, used to learn each expert and gates model $p$, defined here as $e_p$ and $g_p$, respectively, must be taken into consideration, since the values of $e_p$ and $g_p$ can become small, thus getting into scenarios where $d \gg g_p$ and $d \gg e_p$. Under such scenarios, the application of regularization is an approach which allows the usage of parsimonious models, in both experts and gate parameters, allowing then the use of MoLE models in high dimensional settings and/or under collinearity, or with a large number of experts; in regularization, a penalty factor on model parameters (e.g., norm) is added to the loss function, forcing small values to model parameters associated to unimportant features. MoLE models are applicable when the output $y$ comes from an heterogeneous population, i.e., $y$ can decomposed into sub-populations defined on a number of, possibly overlapping, regions of the input space [16]. In industrial informatics, it is common the presence of heterogeneous sources, for example, the multiple operating modes of an industry (e.g. the summer-winter operation of a refinery), data coming from different batches (e.g. polymerization process), where in all previous examples, the output $y$ to be predicted is related with quality variable in a form of a soft sensor.

Mixture of experts (MoE) was first introduced in the seminal paper [17], where the basis model considered was a neural network and the gates were softmax functions. The MoE model...
was then applied to multi-speaker vowel discrimination. Since then, and to deal with collinearity and large number of features in MoLE settings (MoLE is the linear version of MoE), some authors proposed different regularization schemes. Based on ridge-regression, lasso and scad regularization, the work in [16] developed a regularized MoLE model by adding a ridge-regression penalty to the gate, and a lasso and scad penalties to the gates and experts parameters, to perform feature selection. A similar approach was proposed in [18], in the classification context, where a lasso penalty was added for learning the expert and gates parameters, and the results have shown that the sparsity on the gates and experts solution provides better solutions, when compared with classical MoLE, and other state-of-the-art classifiers. Other related and recent approaches, have been proposed for finite mixture of experts, most of them are based on the lasso penalty. See [19], for further analysis on this subject.

Learning in such scenarios is not straightforward, thus the use of regularization schemes is an effective way to deal with the above discussed problems. In such scenarios, ridge-regression cannot provide parsimonious models, since all coefficients will have a contribution to prediction. On the other hand, the lasso approach provides sparse solutions, but it does not consider the effect group (i.e. for correlated features, it will tend to select one while shrinking the coefficient of others to zero). An approach in such kinds of scenario is to apply elastic-net regularization, which combines the ridge-regression and lasso penalties, while taking the benefits of both, since it provides sparse solutions (lasso part) while considering the effect group problem (ridge-regression part). Thus, in this paper, the learning of MoLE models is considered by using elastic-net penalty, on both, gates and experts parameters. The expectation maximization (EM) algorithm [20], together with coordinate descent algorithm, are used to find the parameters of the MoLE model. The regularization parameter of experts and gate parameters, are going to be defined based on the leave-one-out error (LOOVC). The proposed method will be evaluated in a real polymerization batch process to predict the acidity number, a quality-related variable.

The paper is divided as follows. Section II gives the background, including the notation and the background on mixture of experts. Section III introduces the MoLE-EN algorithm. Section IV presents experimental results. Finally, Section V gives concluding remarks.

II. BACKGROUND

In this section, the preliminaries for the remaining of the paper are given. These preliminaries includes the notation used in the paper, and an overview on mixture of experts learning based on the expectation and maximization (EM) algorithm.

A. Notation

The notation used here is defined as follows, assume that finite random variables are represented by capital letters and their values by the corresponding lowercase letters, e.g. random variable $A$, and corresponding value $a$. Matrices and vectors are represented by boldface capital letters, e.g. $A$ and boldface lowercase letters, e.g. $a$. The input and output/target variables are defined as $X = \{X_1, \ldots, X_d\}$ and $Y$, respectively. The variables $X_1, \ldots, X_d$ can take $n$ different values as $\{x_{i1}, \ldots, x_{id}\}$ and $y_i$, where $X = [x_{ij} \in R^{n \times d}]$, with elements $x_{ij} = x_{ij}$, and $y$, with elements $y_{i1} = y_i$ are the input matrix and output vector containing all the $n$ examples. The matrix $X_A$, where $A \subseteq \{j = 1, \ldots, d\}$ is the subset index of matrix $X$, is the matrix with the variables defined in the set of index $A$, defined as $[X_A]_{ij} = [x_{ij} | j \in A \land i = 1, \ldots, n]$. The set of examples is defined as $\Phi_n = (x_i, y_i)_{i=1}^n$. The probability that a random variable take the value $a$, i.e. $A = a$, is defined as $p(a)$, the conditional pdf $p(y|x)$ is defined for $Y|X$.

B. Mixture of Experts

The ME method is based on the divide and conquer approach. It partitions the data into soft regions and learns a local regression model for each region. When a data point falls into a specific region or an overlap of regions, the final output is given by a weighted combination of the local models of each region. In ME, the input data is partitioned into soft regions

$$j_{ME}(x_i) = \sum_{p=1}^P \pi_p(x_i) \hat{y}_p(x_i),$$

where $P$ is the total number of experts, $\hat{y}_p(x_i)$ is predicted output of region defined by expert $p$, and $\pi_p(x_i)$ is the gate function, which assigns mixture proportions to each predicted output. The gate function has the following constraints $\sum_{p=1}^P \pi_p(x_i) = 1$ and $0 \leq \pi_p(x_i) \leq 1$. The softmax function (2) is usually used as the gate function,

$$\pi_p(x_i) = \frac{1}{1 + \sum_{\ell=1}^P \exp(\langle v_\ell, x_i \rangle)}$$

where $v_p$ is the gate parameter of gate function $\pi_p(x_i)$. The softmax function, defined above, follows the constraint $\sum_{p=1}^P \pi_p(x_i) = 1$ and $0 \leq \pi_p(x_i) \leq 1$. The regression model of expert $p$, is usually chosen as a multivariate linear regression (MLR) model in the form $\hat{y}_p(x_i) = x_i^T \theta_p$. Fig. 1 shows an example of a MoLE model structure with $P$ experts.

The ME approximates the true pdf $p(y_i|x_i)$ by a superposition of weighted pdfs, as follows:

$$p(y_i|x_i) = \sum_{p=1}^P \pi_p(x_i, \Theta_p) p(y_i|x_i, \Theta_p),$$

where $p(y_i|x_i, \Theta_p)$ is the conditional pdf of $Y|X$ of expert $p$, governed by parameters $\Theta_p = \{\theta_p, \sigma_p^2\}$, and $\pi_p(x_i, \Theta_p)$ are the gate outputs governed by parameters $\Theta = \{v_1, \ldots, v_P\}$. The collection of all parameters in (3) is given by $\Omega = \{\theta_1, \ldots, \theta_P, \sigma_1^2, \ldots, \sigma_P^2, v_1, \ldots, v_P\}$. Assuming that $y_i = x_i^T \theta_p + \epsilon_p$, where $\epsilon_p \sim N(0, \sigma_p^2)$, then the pdf $p(y|x, \Theta_p)$
can be represented by $N(y_t | x_t^T \theta_p, \sigma_p^2)$. In such case, (3) is re-written as

$$p(y_t | x_t, \Omega) = \sum_{p=1}^{P} \pi_p(x_t, V)N(y_t | x_t^T \theta_p, \sigma_p^2),$$

where the output prediction is given by (1). This form is usually called a mixture of linear experts (MoLE) model. The solution for gates $\nu_p$ and experts $\Theta_p$ parameters by direct maximization of log-likelihood of Eq. (4) is unfeasible. Usually, the expectation maximization (EM) and gradient descent (GD) algorithms are used for that purpose, but unlike the GD solution, the EM algorithm allows the improvement of the likelihood (3) by a series of decoupled maximum likelihood estimates for each gate and expert parameters, performed at each iteration of EM algorithm.

Through EM algorithm, the conditional pdf of $Y | X$ is represented by

$$p(y_t | x_t, \Omega) = \sum_{p=1}^{P} p(z_{pi} | x_t, V)p(y_t | x_t, z_{pi}, \Theta_p),$$

where $z_{pi}$ are defined as hidden variables, where $z_{pi} \in \{0, 1\}$ and for each sample $i$, all variables $z_{pi}$ are zero, except for a single one. The hidden variable $z_{pi}$ indicates which expert $p$ was responsible of generating data point $i$. Define $z_i = [z_{i1}, \ldots, z_{ip_i}]^T$ and $Z = [z_{pi}] \in \mathbb{R}^{n \times P}$ as the vector and the matrix of hidden variables, respectively. Assuming that the gate is given by the softmax function and the expert is given by a Gaussian pdf, then $p(z_i | x_t) = \prod_{p=1}^{P} \pi_p(x_t)^{z_{pi}}$ and $p(y_t | x_t, z_i, \Theta_p) = \prod_{p=1}^{P} \left[ N(y_t | x_t^T \theta_p, \sigma_p^2) \right]^{z_{pi}}$, where the following relation holds $\pi_p(x_t, V) = p(z_{pi} = 1 | x_t, V)$ and $N(y_t | x_t^T \theta_p, \sigma_p^2) = p(y_t | x_t, z_{pi} = 1, \Theta_p)$.

In EM, instead of the direct log-likelihood maximization of (5) with respect to $\Omega$, the algorithm considers the iterative maximization of the log-likelihood through the iterative improvement of the expectation of log-likelihood (ELL) of the complete data $\{y, Z\}$ with respect the hidden variables $Z$, through two phases, the expectation step (E-Step), and the maximization step (M-Step). The ELL is defined as $Q(\Omega, \Omega^t) = E_q [\log p(y, Z|X, \Omega)]$, where the superscript $t$ is the EM iteration number. By the EM formulation, the contribution of each expert and gates in ELL can be seen separately, as follows

$$Q(\Omega, \Omega^t) = Q_e(\Theta, \Omega^t) + Q_g(V, \Omega^t),$$

where $Q_e(\cdot)$ and $Q_g(\cdot)$ accounts by the experts and gates contribution to ELL, respectively, as follows

$$Q_e(\Theta, \Omega^t) = \sum_{i=1}^{n} \sum_{p=1}^{P} \gamma_{pi} \log N(y_t | x_t^T \theta_p, \sigma_p^2),$$

$$Q_g(V, \Omega^t) = \sum_{i=1}^{n} \sum_{p=1}^{P} \gamma_{pi} \log p_i, p(y_t | x_t, V),$$

where $\gamma_{pi}$ is called a responsibilities, which accounts for the probability of sample $x_t$ to belong to the region covered by expert model $p$. It is defined as

$$\gamma_{pi} = \gamma_{p}(x_t) = \frac{\pi_p(x_t)N(y_t | x_t^T \theta_p, \sigma_p^2)}{\sum_{k=1}^{P} \pi_k(x_t)N(y_t | x_t^T \theta_k, \sigma_k^2)}.$$
where \( p(z_{pi} = 1|x_i, V) = \pi_{pi} \). In order to find the parameters \( V \), it is necessary to find the solution of (9). The direct maximization of (14) is not straightforward. Instead the iterative reweighted least squares (IRLS) is usually used for such a purpose. The IRLS algorithm works in the following way. First, define the follows auxiliary variable \( \psi^* = \psi^p \), where \( \psi^* \) is the auxiliary parameter in the \( k \)-th iteration of IRLS algorithm, and \( \psi_p \) is the \( l \)-th iteration of EM algorithm; then the auxiliary parameter \( \psi^* \) is updated as follows

\[
\psi^k+1 = \psi^k + \left( \frac{\partial^2 Q_p(V, \Omega)}{\partial \psi^k(V^k)} \right)^{-1} \left( \frac{\partial Q_p(V, \Omega)}{\partial \psi^k} \right)
\]

(15)

where \( R_p \) is the diagonal matrix of responsibilities. Under such scenario, the last iteration of the IRLS algorithm. In practice, few iterations \( K \) are necessary to find the solution of each \( \psi_p \).

\[ I_{R} \]

**III. MIXTURE OF LOCAL EN EXPERTS**

Consider the ME formulation defined in Section II-B. The objective here is to maximize the log-likelihood of experts \( Q_e(\cdot) \) and gate \( Q_o(\cdot) \) as defined by (10)-(11), (14) by using an elastic net penalty on the parameters to develop the MoLE-EN.

\[ A. Local Experts Regularization by EN \]

First, consider the weighted conditional pdf of expert \( p \), as follows

\[
p_p(y_i|x_i, \theta_p) = \left[ \frac{1}{\sqrt{2\pi\sigma}} \exp \left( \frac{-(y_i-x_i^T \theta_p)^2}{2\sigma^2} \right) \right]^{\gamma_{pi}}.
\]

(16)

Considering \( \gamma_{pi} = \gamma_{pi}^{e} \), the log-likelihood of (16) becomes \( Q_e(\cdot) \) in (11), where the solution for each \( \theta_p \) is given by (12). In that sense, in this paper, the EN regularization is going to be applied to the model expert \( p \), which allows the application of MoLE-EN in scenarios where \( d \ll n \), with a straightforward interpretation of coefficients for later feature selection. Then, defining the unnormalized prior density function, for the parameter of expert \( p \), as being \( p(\theta_p) \propto \exp \left[ -\lambda \sum_{i=1}^{d} |\theta_{pi}|^2 + (1-\alpha) \sum_{i=1}^{d} |\theta_{pi}| \right] \), the following relation holds

\[
\log(p(\theta_p|x_i, y_i)) = \log(p_{\gamma}(y_i|x_i, \theta_p))p(\theta_p),
\]

where

\[
\log(p_{\gamma}(y_i|x_i, \theta_p)) = Q_{e,\lambda} = Q_e + \lambda_p \left( \alpha \sum_{i} |\theta_{pi}^e|^2 + (1-\alpha) \sum_{i} |\theta_{pi}| \right)
\]

(17)

is the contribution of expert \( p \) to ELL with a EN penalty on model parameters. The solution for parameters \( \theta_p \) can be found as

\[
\theta_p = \arg \min_{\theta_p} \frac{1}{2} \sum_{i=1}^{n} \gamma_{pi}^e(y_i - x_i^T \theta_p)^2 + \lambda_p \left( \alpha \sum_{i} |\theta_{pi}^e|^2 + (1-\alpha) \sum_{i} |\theta_{pi}| \right).
\]

(18)

The minimization of (18) can be achieved using the coordinate gradient descent (CGD) algorithm, described in detail in [21]. The toolbox glmnet [21] provides an effective solution for EN regularization based on the CGD algorithm.

1) Model Selection Criteria for Expert \( p \): The major issue is selecting properly the values of \( \alpha \) and \( \lambda_p \), which control the sparsity of solution. In this paper, the value of \( \alpha \) is going to be kept fixed at \( \alpha = 0.5 \). However, there is still the problem of selecting the appropriate value of \( \lambda_p \), and for such purpose, the leave-one-out error (LOOCV) for each expert is going to be employed. The following derivation is based on the PRESS statistics [22]. First, define the active set for the \( p \)-th expert model as \( A_p = \{i \in \{1, \ldots, n\} | \theta_{pi} \neq 0 \} \), which are the index of features with non-zero coefficients. Under such scenario, and assuming that the active set of expert \( p \) remains the same after removing a sample \( i \), then the hat-matrix of expert, defined as \( H_p \), is given by:

\[
H_p = X_{A_p}(X_{A_p}^T W_p X_{A_p} + \lambda I)^{-1} X_{A_p}^T W_p.
\]

(19)

where \( \lambda = \lambda_p \alpha \) and \( W_p = \text{diag}(\gamma_{p1}, \ldots, \gamma_{pn}) \) is the diagonal matrix of responsibilities. Under such scenario, the approximated predicted output of expert \( p \) at the sample, and without sample \( i \) into the fitting, is given by:

\[
\hat{y}_{pi} = \frac{(\hat{y}_{pi} - h_{pi} y_i)}{1 - h_{pi}}.
\]

(20)

where \( h_{pi} \) is the \( i \)-th diagonal element of matrix \( H_p \). Then, under these circumstances the weighted LOOCV error for expert \( p \), defined here as eLOOCV, and considered in this work is defined as:

\[
eLOOCV_{p} = \sum_{i=1}^{n} \gamma_{pi} \left( y_i - \hat{y}_{pi} \right)^2.
\]

(21)

The value of \( \lambda_p \) is the one that minimizes criterion (21).

\[ B. Local Gates Regularization by EN \]

In experts learning, the objective is the minimization of Eq. (18), where the EN regularization was easily added to the maximization of \( \theta_p \)’s. On the other hand, during the gates learning, the application of the regularization term is not so explicit. Then, assuming the closed form solution given by the IRLS algorithm, in the form of a weighted least squares equation, as described in (15), and mimicking the results derived from the experts to the gates, the value of \( \psi_p \) to be found at each IRLS solution, with the EN regularization added, is then given by

\[
\psi_p = \arg \min_{\psi_p} \frac{1}{2} \sum_{i=1}^{n} \gamma_{pi}^e (\hat{z}_i - x_i^T \psi_p^e)^2 + \lambda_p \left( \alpha \sum_{i} |\hat{\psi}_{pi}^e|^2 + (1 - \alpha) \sum_{i} |\hat{\psi}_{pi}| \right).
\]

(22)
Similarly to the case of the experts, the solution of (22) can be achieved by using the coordinate gradient descent algorithm, described in [21].

1) Model Selection Criteria for Gate $p$: The major issue here is to find the most appropriate model for each gate $p$ and at each iteration of IRLS algorithm, where the value of $\lambda_g^p$ controls the sparsity of the solution. Similarly to the experts parameters, the selection of $\lambda_g^p$ for each gate model is going to be based on the LOOCV criterion. For the gates it is defined as gLOOCV as follows:

$$g\text{LOOCV}_{p,\lambda_g^p} = \sum_{i=1}^{n} u_{pi} \left( \hat{z}_i - \hat{z}_i(-1) \right)^2,$$  \hspace{1cm} (23)

where $\hat{z}_i(-1)$ is the predicted output of model $z$ without the sample $i$, and

$$\hat{z}_i = \frac{(\hat{z}_{pi} - m_{pi} z_i)}{1 - m_{pi}},$$  \hspace{1cm} (24)

where $m_{pi}$ is the $i$th diagonal element of $M_{pi}$, which is the hat-matrix given by each iteration of the IRLS algorithm, and it is given by

$$M_p = X_{B_p}(X_{B_p}^T \Gamma_p X_{B_p} + \lambda I)^{-1}X_{B_p}^T \Gamma_p,$$  \hspace{1cm} (25)

where $\lambda = \lambda_g^p \alpha$ and $B_p = \{i \in \{1, \ldots, n\} | \pi_{pi} \neq 0\}$ is the active set of gate $p$. The value of $\lambda_g^p$ to be selected is the one that minimizes the criterion (23).

IV. EXPERIMENTAL RESULTS

This section discusses the experimental results on the acidity prediction in polymerization process. Process description, experimental settings and results will be discussed in this section.

A. Polymerization Process Description

In this paper, the objective is the prediction of the acidity number, which is a quality variable measured in laboratory and scarcely along the process. The plant is equipped with several sensors for online measurement, connected to a process computer that records the values every 30 [s]. In total, 34 variables are collected and recorded, either by simply obtaining the measurements from sensors (temperatures, pressures, valve openings, etc) or, in controller set-points, adjusted manually by the process operators. Each batch then contains anywhere between 4500 and 7500 recordings of each process variable (each batch takes 40-70 [h]). The acidity is not measured online. Instead, product samples are taken manually, quite infrequently and unevenly (once every 1.5 - 2[h], depending on the evolution of the batch), and are sent for laboratory analysis, which takes approximately 20 [min] to be completed, with the accuracy being $\sim 10\%$ of the reading. Moreover, the quality measurements are only available 8-10 [h] after the batch starts (i.e. after at least 1000 time instants have elapsed). This equates to only 15-20 quality measurements being available for each batch. The assembled dataset (kindly provided by [23]) contains data from 33 batches (16 months of operating effort), split into two subsets: 27 batches constitute the training set, and the remaining 6 represent the test set.

Table I: Performance results of all methods on the Acidity test dataset.

<table>
<thead>
<tr>
<th></th>
<th>EN</th>
<th>MoLE-EN</th>
<th>SVM</th>
<th>DT</th>
</tr>
</thead>
<tbody>
<tr>
<td>$R^2$</td>
<td>0.952</td>
<td>0.965</td>
<td>0.897</td>
<td>0.908</td>
</tr>
</tbody>
</table>

Figure 2: Gate output $\pi_{pi}$ for all experts; first 100 samples.

The objective in this data set is the estimation of the quality of a resin produced in an industrial batch polymerization process [23]. One of the resin quality indicators is the resin acidity number ($N_A$). The data set is composed of 24 input variables and the authors [23] have predefined 521 samples for training and 133 for test. The original and squared features were considered for learning, then $d = 48$ in experiments.

B. Experimental Settings

To access the performance of each regression model, the sum of square error (SSE) and correlation coefficient ($R^2$) were applied. These indexes are defined as follows. $SSE = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2$ and $R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$, where $\hat{y}_i$ is the predicted output and $\bar{y}$ is the mean value of output $y$:

$$\bar{y} = \frac{1}{n} \sum_{i=1}^{n} y_i.$$

The MoLE-EN model was learned by using 27 experts, the same number of batch cycles in training data. For all the other models, a 10-fold cross-validation scheme was applied to set the hyper-parameters. A radial basis function kernel was considered for SVM. The convergence of the MoLE-EN model was measured by using the computed ELL. A total of 10 MoLE-EN models were learned and the one with the highest ELL was considered for evaluation.

C. Experimental Results

Table I shows the prediction results for EN, and MoLE-EN model with 27 experts, the SVM and DT, respectively. The MoLE-EN had the best prediction performance among all models, with some improvement performance over the EN model. However, the MoLE-EN model is more interpretable, in the sense that each batch can be described by the gates, as shown in Fig. 2. It is possible to notice some common behavior along each batch. This information can be used later for process monitoring or for defining the profile of each batch, e.g. from a new process it is possible to known what is the batch most similar into history data. This situation will be explored in future works. The effective number of samples, used in learning each expert, given by (13), is shown in Fig. 3.
It shows that the effective number of samples used for learning the MoLE-EN model in each expert is always less than the total number of features. The MoLE-EN prediction is shown in Figure 4.

V. CONCLUSIONS

In this paper, a MoLE model based on the EN regularization, was proposed and applied for production quality prediction in batch settings; more specifically for prediction of acidity value in a polymerization process. Preliminary results have shown that the MoLE-EN can make use of a large number of experts, without prejudicing the final model performance. This is an ongoing work, and future works will address the problem of model selection in MoLE-EN models, study the sensitivity analysis in MoLE-EN models, and the applicability of MoLE-EN for batch monitoring.

ACKNOWLEDGMENTS

The authors acknowledge the support of FCT project UID/EIA/00048/2013. Francisco A. A. Souza has been supported by Fundação para a Ciência e a Tecnologia (FCT) under grant SFRH/BPD/112774/2015.

REFERENCES